1992

The Euclidean Distance Transform (Thesis)

Ching-Shoei Chiang

Report Number:
92-050

THE EUCLIDEAN DISTANCE TRANSFORM

Ching-Shoei Chiang

CSD-TR 92-050
August 1992

THE EUCLIDEAN DISTANCE TRANSFORM

A Thesis
Submitted to the Faculty

of

Purdue University

by

Ching-Shoei Chiang

In Partial Fulfillment of the
Requirements for the Degree

of

Doctor of Philosophy

August 1992

To my parents,

Goang-Long Chiang and Shiow-Mei Hwang Chiang

ACKNOWLEDGMENTS

I would like to acknowledge the great support and encouragement of my advisor, Professor Christoph M. Hoffmann. His insight, guidance and patience made this thesis possible. I would like to thank the other members of my graduate committee: Professors David C. Anderson, Robert E. Lynch, and Elias N. Houstis, for their interest in my research and for serving on my thesis committee. I am also indebted to Professor George Vanecek Jr. who spent generous amount of time to discuss the implementation aspect of this thesis.

I thank all the friends who have offered assistance and encouragement during my stay at Purdue. Among them are William Bouma, Denny Chen, Xiangping Chen, Jung-Hong Chuang, Neelam Jasuja, Robert Juan, Jyh-Jong Tsay, Pamela Vermeer, Ko-Yang Wong, Jiaxun Yu, Jianhua Zhou,

My wife Mei-Yin Hou deserves special thanks for her love, encouragement, patience, understanding support and her time to take care of my son Hans while she was busy pursuing her M.S. degree.

Finally, to my parents, Goang-Long Chiang and Shiow-Mei Hwang Chiang, I would like to express my deepest appreciation. This thesis is dedicated to them.

I would like to acknowledge the financial support provided by the Office of Naval Research under contract N00014-90J-1599, and the National Science Foundation under grant CCR-8619817 and ECD–88-19817.

TABLE OF CONTENTS

Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Figure Page

Figure Page

Appendix
Figure

ABSTRACT

Chiang, Ching-Shoei. Ph.D, Purdue University, August 1992. The Euclidean Distance Transform. Major Professor: Christoph M. Hoffmann.

The medial-axis transform (MAT), also called skeleton, is a shape abstraction proposed by computer vision, and has a number of important engineering applications such as finite-element mesh generation. The theory for the MAT of 2D solids is investigated. We prove the uniqueness, divisibility and connectedness properties of the MAT of 2D solids. Algorithms are proposed to extract the MAT based on two criteria, the maximal circle criterion and the equal distance criterion. The algorithms which use the maximal circle criterion produce the MAT along with many noisy point, and further refinement using a threshold value is needed. The algorithms which use the equal distance criterion walk along the MAT from a starting MAT point. Because the connectedness property, we can find all of the MAT if a good strategy is used to walk along the MAT. Systems of equations are generated so that Newton iteration can be applied, and new MAT points can be found. The algorithms can be extended to 3D with more complex data structure.

Detecting self-intersections in offsets is a problem that has both a mathematical and a combinatorial character. Some self-intersections can be detected based on local criteria applied to boundary elements, but others require evaluating the spatial relationship between unknown parts of the base curve, and this is difficult for the traditional offset algorithms. We solve the problem by applying the Euclidean distance transform having suitably discretized the geometric shapes. Several strategies are presented and are compared for efficiency and performance.

1. INTRODUCTION

There are three sections in this chapter. The first section introduces the distance function and distance transform in the real plane for a binary image. The second section describes the medial axis and medial axis transform, and the last section describes the organization of this thesis.

1.1 The Euclidean Distance Function and Distance Transform

Many distance functions, such as city block distance, chessboard distance, chamfer distance, octagonal distance, and quasi-Euclidean distance, have been used for digital image processing [7, 8, 46, 60, 14]. Most of them try to approximate Euclidean distance. They have important applications in digital image processing. Examples include offsetting and skeletonizing of binary images. This section describes the idea of distance functions, and also describes the distance transform (DT) for a binary image.

Let us consider the Euclidean distance function first, and then discuss some distance functions for binary images. In Euclidean geometry, the distance from point p to point q can be measured by the size of the vector $q-p$. The size of a vector with n elements is measured by a *norm*. The norm assigns to each n-vector a a real number $\|a\|$, called the norm of a, subject to the following restrictions: [13]

1. For all n-vectors a, $\|a\| \geq 0$. Moreover, $\|a\| = 0$ if and only if a is the zero vector.

2. For all n-vectors a and all numbers α, $\|\alpha a\| = |\alpha|\|a\|$

3. For any two n-vectors a and b, $\|a + b\| \leq \|a\| + \|b\|$

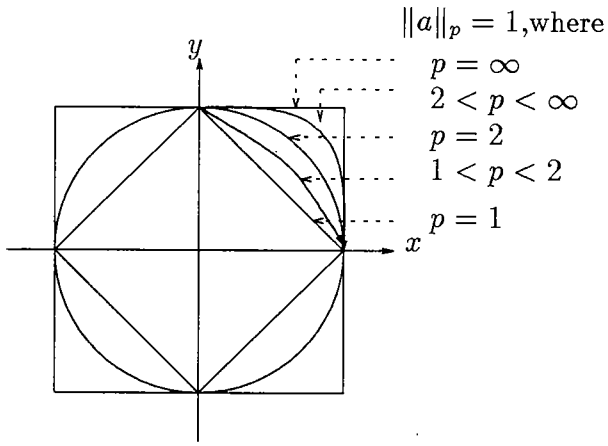
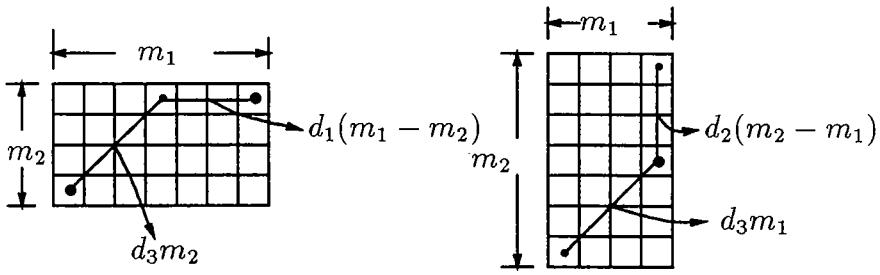


Figure 1.1 The p-norm unit circle for $1 \leq p \leq \infty$

Various norms have been investigated in mathematics. The vector norms in frequent use include the 1-norm, the 2-norm (also called Euclidean length), and the ∞-norm, also called maximum norm. The p-norm of an n-vector is defined as $\|a\| = \|a\|_p = (\sum_{i=1}^{n} |a_i|^p)^{1/p}$, where p is a real number between 1 and ∞. A picture for $\|a\|_p = 1$, where a is 2-vector, is shown in Figure 1.1.

A digital image can be thought of as an array of pixels. If the image is black-and-white, a black pixel is called feature element and a white pixel is called nonfeature element. We thus think of a digital image as a discrete set of feature points that are indexed by integer coordinates. The distance between two image points can be an integer or a real number, depending on numbers d_1, d_2, and d_3 used to measure the distance of a image point to its neighbors. Here, d_1 and d_2 measure the distance of horizontal and vertical neighbors, and d_3 measures the distance to diagonal neighbors, as shown in Figure 1.3. Different values have been assigned to d_1, d_2, and d_3 for different distance functions, as shown in Table 1.1. In this table, Chamfer Euclidean is synonymous with Chamfer$(1.0, \sqrt{2})$ and Chamfer optimal is synonymous with Chamfer(1.0,1.351). Consider the distance between two images points (i_1, j_1) and (i_2, j_2). Let $m_1 = |i_1 - i_2|$ and $m_2 = |j_1 - j_2|$, then the distance between these two image points is $d_3 m_1 + d_2(m_2 - m_1)$ for $m_2 \geq m_1$ and $d_3 m_2 + d_1(m_1 - m_2)$ for



(a) for $m_1 > m_2$ (b) for $m_1 \leq m_2$

Figure 1.2 The distance between two image points

$m_2 < m_1$, as shown in figure 1.2. Instead of using integer or real value for d_1, d_2, and d_3, Euclidean distance uses the 2-vector norm to measure the distance. An example for the distance of two image points for each distance function is shown in Figure 1.4. By using different assignments for d_1, d_2, and d_3, one tries to approximate scaled Euclidean distance to some accuracy. In all cases except Euclidean distance, the scale value d_1 is equal to d_2. In Euclidean distance, the norm for d_1 and d_2 are the same. For comparison the distance with Euclidean distance in Figure 1.4, the computed distance must divided a scale d_1 in Chamfer(3,4) or Chamfer(2,3). Rosenfeld and Pfaltz [60] define the distance function as follows:

Definition 1 Let G be a grid image with coordinates (i, j) where $1 \leq i \leq m$ and $1 \leq j \leq n$. The function $f : G \times G \rightarrow N$, where N is the set of nonnegative integers, is called a distance function if

1. $f(x, y) = 0$ if and only if $x = y$.
2. $f(x, y) = f(y, x)$ for all $x, y \in G$
3. $f(x, z) \leq f(x, y) + f(y, z)$ for all $x, y, z \in G$

[60] proves that the city block distance function, given by $f_1((i_1, j_1), (i_2, j_2)) = |i_1 - i_2| + |j_1 - j_2|$, and the chessboard distance function, given by $f_2((i_1, j_1), (i_2, j_2)) =$

d_3	d_2	d_3
d_1		d_1
d_3	d_2	d_3

Figure 1.3 Neighbors symbol for image point

Table 1.1 Neighbors distance value for image point

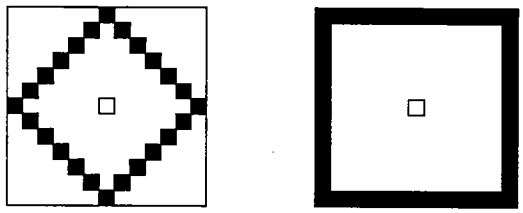
Distance Function	d_1	d_2	d_3
City block	1	1	2
Chessboard	1	1	1
Chamfer Euclidean	1.0	1.0	$\sqrt{2}$
Chamfer optimal	1.0	1.0	1.351
Chamfer(3,4)	3	3	4
Chamfer(2,3)	2	2	3
Euclidean	(1,0)	(0,1)	(1,1)

.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	a	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	b	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	c
.	.	.	.	.	.	.	.	.	.

Distance Function	a to b	b to c	c to a
City block	10	10	10
Chessboard	5	8	7
Chamfer Euclidean	7.071	8.828	8.234
Chamfer optimal	6.755	8.702	8.053
Chamfer(3,4)	20	26	21
Chamfer(2,3)	15	18	17
Euclidean	7.071	8.246	7.616

For comparison with the Euclidean distance, the computed distance in Chamfer(m, n) must be divided by m.

Figure 1.4 Example for distance using different distance function



City Block Chessboard

Figure 1.5 City block distance and chessboard distance

$max(|i_1 - i_2|, |j_1 - j_2|)$, are distance functions in this sense. Notice that the picture of the circle for $f_1((i,j),(0,0)) = C$, as shown in Figure 1.5, is similar to $\|(i,j)\|_1 = C$, where C is a constant. The only difference is that $f_1((i_1,j_1),(i_2,j_2)) = C$ consists of discrete image points, whereas $\|(i_2 - i_1, j_2 - j_1)\|_1 = C$ consists of a continuum of points. In this sense, the Euclidean distance is similar to the 2-norm, and chessboard distance to the ∞-norm. With more complex formulas, we can also find the octagonal distance function [60].

Let G be a binary picture with coordinates (i,j), for $1 \leq i \leq m, 1 \leq j \leq n$. Then $G(i,j)$ has two types of elements, called feature and nonfeature elements. A distance transformation (DT) is an operation that converts G into a integer picture or real picture, depending on which distance function it used, where each element of the picture has a value that approximates the distance to a nearest feature element. Consider a $(m+2) \times (n+2)$ matrix $G'(i,j)$, and assume "max" is a sufficient large number. We initialize G' by:

$$G'(i,j) = \begin{cases} 0 & \text{if } G(i,j) \text{ is a feature element} \\ max & \text{otherwise} \end{cases}$$

The algorithm for finding DT for G by using city block, chessboard, or chamfer(m,n) distance function is:

For $i = 1$ to m step 1

For $j = 1$ to n step 1

$$G'(i,j) = \min \begin{cases} G'(i,j) \\ G'(i-1,j-1) + d_3 \\ G'(i-1,j) + d_2 \\ G'(i-1,j+1) + d_3 \\ G'(i,j-1) + d_1 \end{cases}$$

For $i = m$ to 1 step -1

For $j = n$ to 1 step -1

$$G'(i,j) = \min \begin{cases} G'(i,j) \\ G'(i+1,j+1) + d_3 \\ G'(i+1,j) + d_2 \\ G'(i+1,j-1) + d_3 \\ G'(i,j+1) + d_1 \end{cases}$$

Borgefors [7] uses masks to describe the algorithm graphically. The masks for this algorithm are shown in Figure 1.6. This algorithm uses 2 passes and 10 comparison on each grid point. Figure 1.9 shows the DT for a binary image in which feature elements are represented by the value 0.

Danielson [14] computes Euclidean distance using two algorithms, 4SED and 8SED. The masks of 4SED and 8SED are shown in Figure 1.7 and Figure 1.8, and are called Danielson 4 and Danielson 8 distance in these algorithms [7]. Four passes with 10 comparisons at each grid point, or three passes with 11 comparisons at each grid point, for 8SED are necessary [64]. The key to his algorithm is to store at each grid point the distance amplitudes, i.e., the distance in the x and y directions to the nearest boundary grid point. Integer values can be used for the distance amplitudes because the distance amplitudes are always a multiple of h, where h is the grid spacing. If a given set B is not discretized as a set of grid points, interpolatory schemes for computing the distance transform can be devised [9]. For example, the curve B can be approximated by a polygonal arc, where the segments are induced by the grid lines, and adjacent grid points can be assigned the perpendicular distance to the nearest boundary segment, see Figure 5.5. This improves the accuracy of the

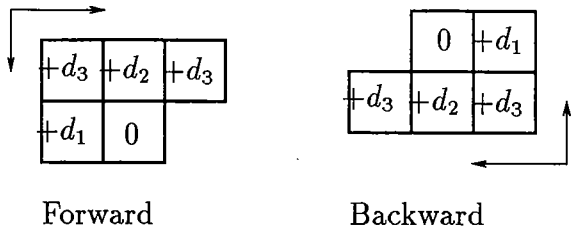


Figure 1.6 The masks for city block, chessboard and chamfer distance

distance computation significantly, but the distance amplitudes are no longer multiples of the grid spacing. Further accuracy is obtained using iteration. This approach will be described in more detail in Chapter 5 of this thesis. With slight modification [64] by determining the signed distance amplitudes at each grid point, we can know not only the distance to the boundary, but also the direction to the boundary. The distance computed by Danielson's 4SED and 8SED is not exactly the Euclidean distance. With a post process computation after 4SED or 8SED, that can be done in parallel [79], the exact Euclidean distance function can be found.

Montanari [46, 47] uses quasi-Euclidean distance functions to approximate the Euclidean distance function. In his approach, method 0 is exactly the same as Rosenfeld's [66, 59] approach. He then defines methods 1, 2, ..., where each method uses a more complicated distance function. When k is large enough, such as the maximal distance of the 2D image, method k finds a skeleton that is exactly the same as the one using Euclidean distance. His algorithm is time consuming when k is large.

Different algorithms for different distance functions have been proposed by many researchers. All algorithms try to minimize the number of passes over the image, and achieve fewer comparisons at each image point, while obtaining an accurate approximation to Euclidean distance. As stated in [7]:

> Not all DTs yield results that are distances in the mathematical sense, as the triangle inequality may be violated.

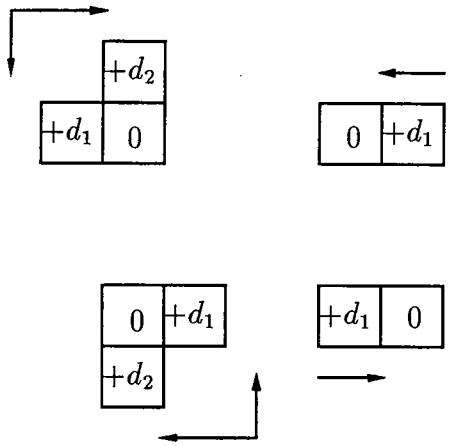


Figure 1.7 The masks for 4SED

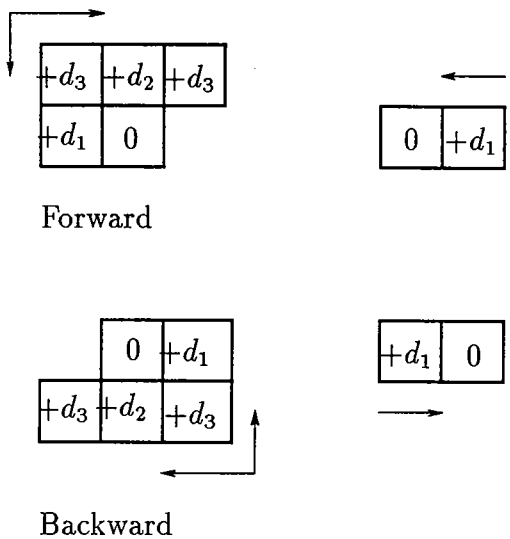


Figure 1.8 The masks for 8SED

Table 1.2 The comparison on a (M+1) × (M+1) squared picture

Distance Transform	Upper bound	θ angle	2/4 passes	Integer/Real distance
City block	-0.586M	45^0	2	Integer
Chessboard	0.414M	45^0	2	Integer
Chamfer Euclidean	-0.090M	24.5^0	2	Real
Chamfer optimal	±0.064M	20.5^0 or 45^0	2	Real
Chamfer(3,4)	0.081M	19.5^0	2	Integer
Chamfer(2,3)	-0.134M	30^0	2	Integer
Octagonal	0.118M for M>5	-	4	Integer
4SED	-0.29	-	4	Real
8SED	-0.076	-	4	Real

Borgefors [7] compares the upper bound of the difference between the Euclidean distance and the computed distances, as shown in Table 1.2. The second column of the table lists the upper bound and the third column lists the direction where the upper bound is attained. The angle θ in this column indicates the direction that is θ degrees from the horizontal or vertical lines. The sign in the second column indicates that the computed distance at the maximal upper bound is overestimated or underestimated, if negative or positive respectively, with respect to the Euclidean distance. The third column lists the number of passes over the image needed for the distance transform. For example, 2 passes are needed when using Chamfer(3,4) and the resulting distance transform has a maximal absolute error bound 0.081M, if the picture is in a (M+1) × (M+1) grid, and the error bound occurs on lines that are about 24.5^0 from horizontal or vertical lines. As stated in [7], the DT using city block distance always overestimates and the DT using chessboard distance always underestimates the Euclidean distance.

Although a discrete plane is used in image processing, we would like to analyze problems in the continuous plane, so that the result of the image processing program

8	7	6	5	4	3	2	3	4	5
7	6	5	4	3	2	1	2	3	4
6	5	4	3	2	1	0	1	2	3
5	4	5	4	3	2	1	2	3	4
4	3	4	5	4	3	2	3	4	5
3	2	3	4	5	4	3	4	5	4
2	1	2	3	4	5	4	5	4	3
1	0	1	2	3	4	5	4	3	2
2	1	2	3	4	5	4	3	2	1
3	2	3	4	5	4	3	2	1	0
4	3	4	5	6	5	4	3	2	1

City block

6	5	4	3	2	2	2	2	2	3
6	5	4	3	2	1	1	1	2	3
5	5	4	3	2	1	0	1	2	3
4	4	4	3	2	1	1	1	2	3
3	3	3	3	2	2	2	2	2	3
2	2	2	2	3	3	3	3	3	3
1	1	1	2	3	4	3	3	3	3
1	0	1	2	3	4	3	2	2	2
1	1	1	2	3	4	3	2	1	1
2	2	2	2	3	4	3	2	1	0
3	3	3	3	3	4	3	2	1	1

Chessboard

Figure 1.9 City block distance and chessboard distance

can be predicted. Because of the numerical error caused by the discretization process, the predicted result is not always the same as the the result from the image processing program. But, we can use this predicted result to evaluate whether the algorithm is good or not, depending on whether its results are close to the predicted results or not.

Now, let us analyze the distance functions in the continuous plane, and produce the "unit circle", such as in Figure 1.1, for each distance function. Without loss of generality, we only consider half of the first quadrant, say case (a) in Figure 1.2. Let us use x, y for real coordinates, as opposed to m_1, m_2 for integer coordinate in Figure 1.2. Then the distance for the point (x, y) to the origin is $d_1(x-y)+d_3y = d_1x+(d_3-d_1)y$ for the case $x > y, x \geq 0$ and $y \geq 0$. The picture for $d_1x + (d_3 - d_1)y = d_1$ is a line with slope $-d_1/(d_3 - d_1)$. Notice that in chessboard distance, the slope of the line is $-\infty$ because $d_3 - d_1 = 0$. Now, consider all four quadrants. The full picture of the points at unit distance to the origin in the continuous plane will be an octagon. We sketch the picture for city block, chess board and chamfer Euclidean distance in the continuous plane, and compare it to a circle, as shown in Figure 1.10. Because of the symmetry of the picture, only a quarter of the circle is drawn. A computer generated picture in the discrete plane can be found in [64].

The upper bound for the error using 4SED or 8SED, as shown in Table 1.2, does not depend on the size of the picture. Furthermore, it has a small error bound compared to the DT using the other distance function. So, we use it to calculate the DT for image. There are three ways to extend 4SED and 8SED to 3D, namely 6SED, 18SED, and 26SED, depending on how many neighbor grid point are considered for finding the DT of a specified grid point. Although their upper error bound is higher than in 2D, for example, the upper error bound for 6SED is 0.42, we believe that it is still better than other distance function proposed in 3D. With a slight modification, the 6SED, 18SED and 26SED determines not only the distance to a feature point, but also the direction to the feature point [80].

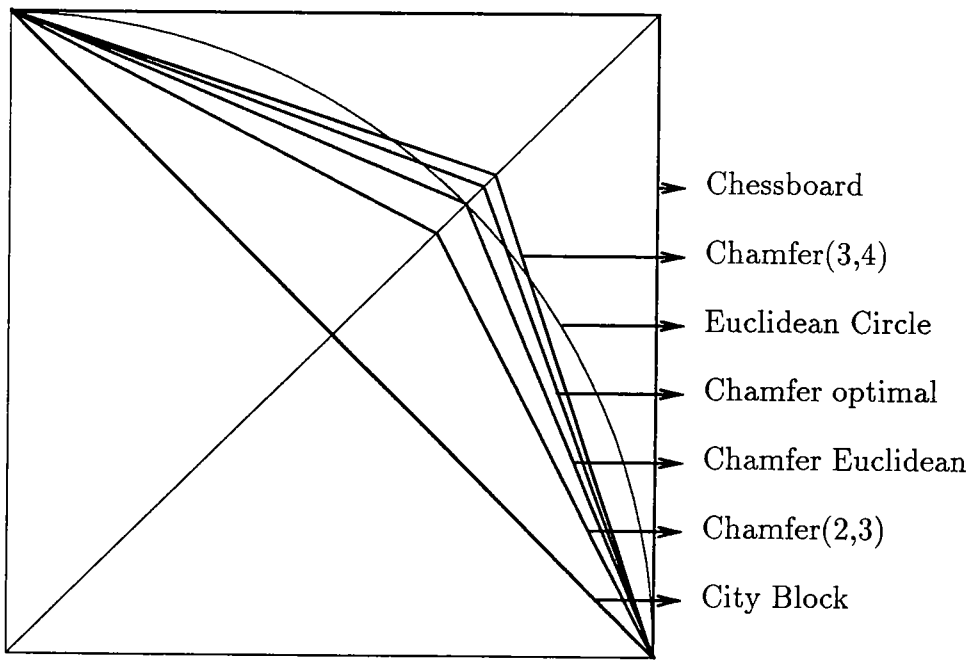


Figure 1.10 The constant distance to a feature element

1.2 The medial axis and medial axis transform

Consider a compact solid T in $\Re^2$ or $\Re^3$ whose boundary is ∂T, for example a CSG object. There are two equivalent definitions for the medial axis (MA) of such solids, namely:

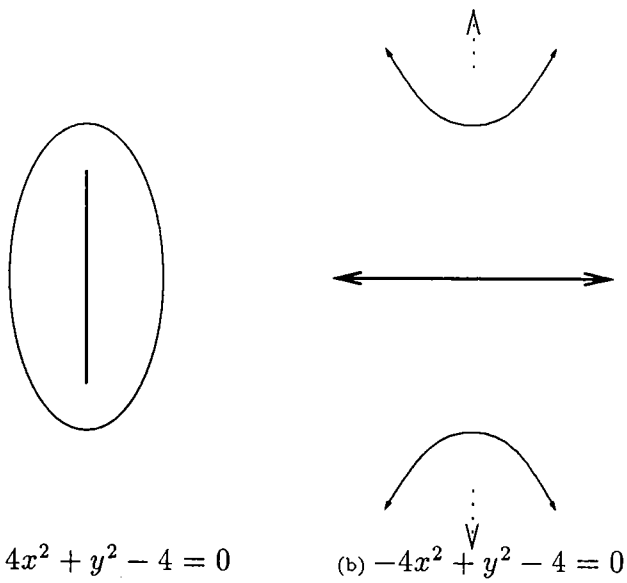
Definition 2 (Blum) The internal medial axis (or skeleton) of a 2D compact area or a 3D compact volume is the closure of the locus of the centers of all maximal disks or spheres which are contained in the shape.

Definition 3 Let a footpoint of p be a point p' on the boundary of a 2D area or of a 3D volume that has minimum distance to p. The interior medial axis of a 2D closed area or a 3D volume is the closure of the locus of the points inside the area or the volume which have more than one footpoint.

Figure 1.13 shows a 2D area and its medial axis. The definition 3 can be extended to certain noncompact 2D areas or 3D volumes. Figure 1.11 and Figure 1.12 show some medial axis examples. In these pictures, the medial axis for the region $f(x,y) < 0$ is drawn as a solid curve and for the region $f(x,y) > 0$ is drawn as a dashed curve. Notice that an end point of the medial axis has only one footpoint.

The function which maps an MA point to the distance between the MA point and its footpoints is called the *radius function.* The medial axis and the associated radius function define the medial axis transform (MAT). The definition extends naturally to the concept of an exterior MAT. Because we are not interested in the exterior MAT, the MAT always means the interior MAT in this thesis.

The MAT is also called prairie fire transformation [16], and an MA point is also called a quench point, and the radius function is also called quench function. Imagine that the interior of the domain is composed of dry grass and the exterior of the domain is composed of unburnable wet grass. Suppose a fire is set simultaneously at all boundary points of the domain, and that the fire will propagate inward with



(a) $4x^2 + y^2 - 4 = 0$ (b) $-4x^2 + y^2 - 4 = 0$

Figure 1.11 The medial axis for ellipse and hyperbola

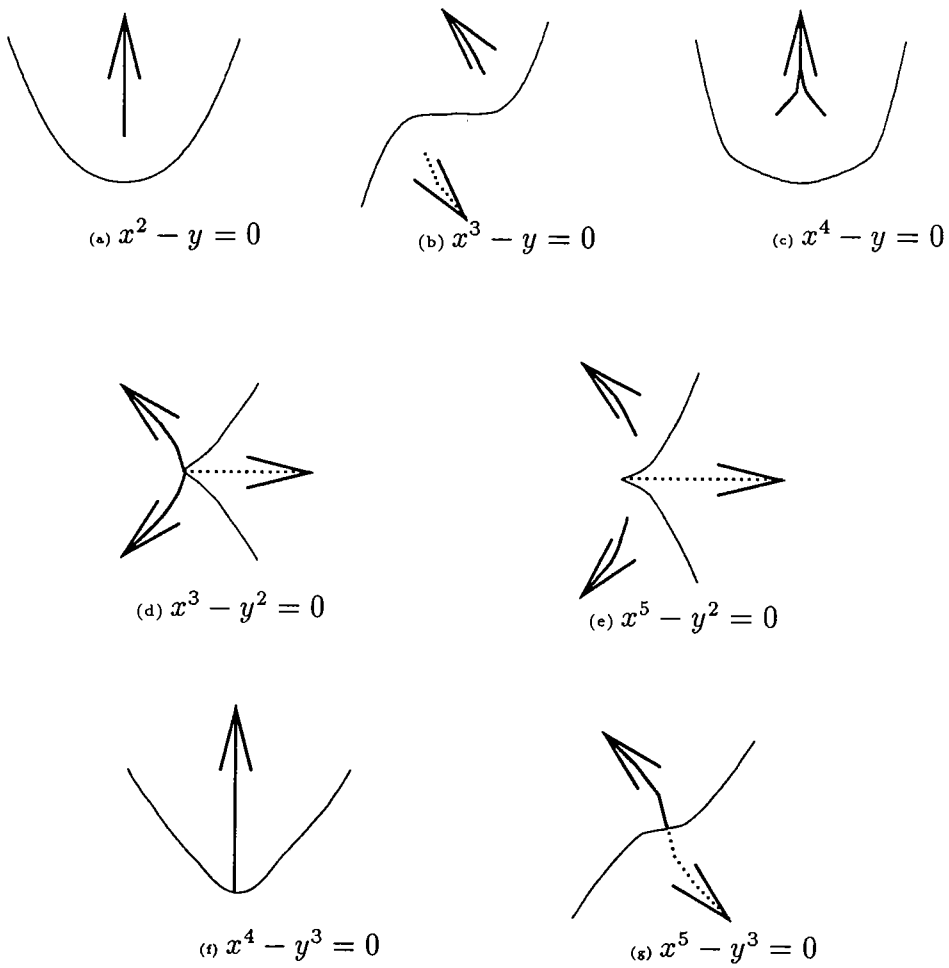


Figure 1.12 The medial axis for the family $x^m - y^n = 0$

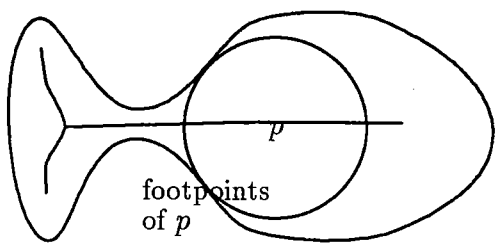


Figure 1.13 skeleton in 2D image

uniform speed. The points at which the fire wave extinguishes itself define the quench points of the fire. The set of quench points defines the MA of the domain. If the exterior grass is also dry, a similar definition of the exterior MAT can be given.

Consider a 2D continuous image D and its discretized image D'; we ask:

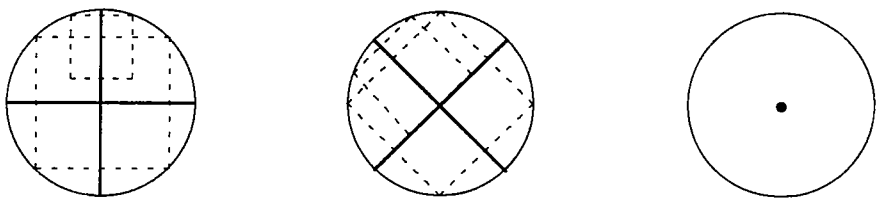
> What does the MAT of D' look like if chessboard or city block distance function are used?

The result MAT depends on the strategy by which we extract them. But, as mentioned in the previous section, we would like have a standard result, so that all strategies can be compared. The closer the MAT we produced, is to the true MAT the better. The true MAT can be obtained by considering the problem in continuous plane on D and using the corresponding distance function. From Figure 1.1 and 1.10, we notice that the unit circle by using the chessboard or the city block distance function in a discrete plane can be considered the 1-norm or ∞-norm unit circle respectively, in the continuous plane. So, the problem becomes:

> What does the MAT of D looks like if 1-norm or ∞-norm are used to measure the distance?

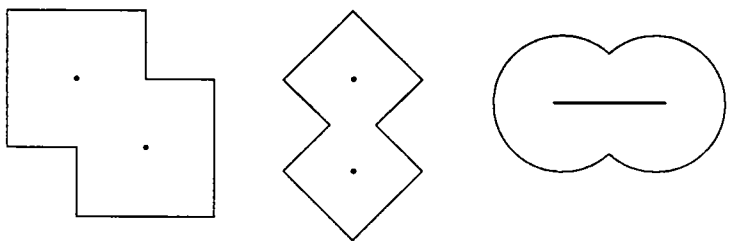
The answer to this problem is simple. By extracting the MAT for D, we want to find the center of the maximal inscribed circle of D. Now, the circle is the 1-norm or ∞-norm circle, as shown in Figure 1.1. So, the MAT for an Euclidean circle, using the 1-norm, ∞-norm and Euclidean norm to measure the distance, is shown in Figure 1.14. The MAT of a continuous image is not necessarily connected if the 1-norm or ∞-norm are used, as shown in Figure 1.15.

There is a close connection between the MA and the Voronoi diagram of a 2D solid [43]. Consider a polygon P that has n elements $E = \{e_1, e_2, \ldots, e_n\}$ where e_i is either an edge or a concave vertex, that is, a vertex whose internal angle is greater than π. The Voronoi polygon associated with e_i is the closure of the set of points closer to e_i than to any other element. The Voronoi diagram of P is the collection of



(a) ∞-norm (b) 1-norm (c) 2-norm

Figure 1.14 The MAT for a Euclidean circle by using different norm



(a) ∞-norm (b) 1-norm (c) 2-norm

Figure 1.15 The connectivity of the MAT by using different norm

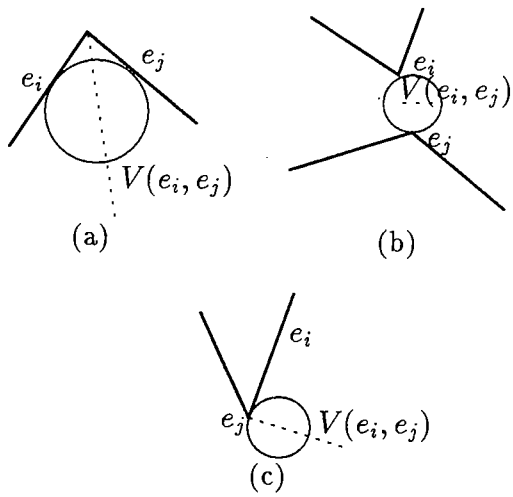


Figure 1.16 Voronoi Edges

Voronoi polygons associated with each of the n elements. Every point on a Voronoi edge is on the boundary of the Voronoi polygon of at least two elements of E and has a footpoint on each of those elements. Let $V(e_i, e_j)$ be the Voronoi edge associated with elements e_i and e_j. When e_i and e_j are both edges, $V(e_i, e_j)$ is an MA because the circles centered on it which are tangent to both edges are maximal inscribed circles, as shown in Figure 1.16(a). Similarly, if both elements are vertices, the circles centered on $V(e_i, e_j)$ touching both e_i and e_j must be maximal, as shown in 1.16(b). However, not all Voronoi edges belong to the MA. When one of e_i and e_j is a concave vertex and the other is the adjacent edge, the resulting $V(e_i, e_j)$ is not a component of the MA. This can be seen in Figure 1.16(c), where $V(e_i, e_j)$ is the Voronoi edge associated with edge e_i and vertex e_j. Inscribed circles with center on $V(e_i, e_j)$ which touch e_j are not necessarily maximal circles, and so $V(e_i, e_j)$ is not part of the MA. From these observations, we can conclude that the MA of P can be obtained from its Voronoi diagram, by deleting some Voronoi edges, or vice versa, by adding certain edges.

1.3 Thesis Organization

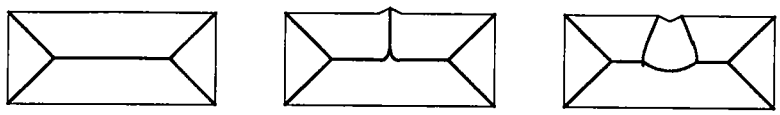
In this chapter, the related work for distance function and medial axis transform has been discussed. Chapter 2 describes the related concepts of distance function from geometry and mechanics. Chapter 3 lists and proves some properties of the 2D MAT of a "simple" domain, and extends them to more general domains. Chapters 4 and 5 discuss the applications of the Euclidean distance function in MAT and offsets of 2D and 3D solids. Different algorithms are proposed and their performances are evaluated. Conclusions and future work are listed in the last chapter.

2. RELATED WORK ON THE MEDIAL AXIS TRANSFORM

There are 5 sections in this chapter. The first four sections introduce how the MAT has been investigated in specific application areas. The first section discusses the MAT development in computer vision. The second section discusses the geometric approaches finding the MAT. The third section discusses the cyclographic map in descriptive geometry and its relation to the MAT. The fourth section discusses the Hamilton-Jacobi equation and its relation to the MAT. The last section discusses other applications of the MAT.

2.1 The MAT in Pattern Recognition

Blum has proposed the medial axis transform to describe biological shapes [4] and many researchers have investigated the use of the medial axis transfer (MAT) in computer vision [66, 59, 28, 61, 5]. There are two major approaches to extract the MAT of a given image. One is called digital thinning [65, 74, 44, 57, 58, 77]. The main feature of this approach is that the border of an image shape is traced and deleted, thereby successively shrinking the shape. With each border trace some MAT points are extracted. The speed of the thinning algorithm depends on the shape in the image. The other approach to extracting the MAT of a given image is via the distance transform of the image [14, 66, 47, 46]. A subsequent pass over the image finds maximal inscribed circles or squares, depending on which distance function used, and decides whether a specified pixel is an MA point by comparing the radius of the circle or square and with that of its neighbors. The algorithms may use Euclidean distance or chessboard or city block distance For this reason, the performance of the algorithm depends not only on the speed and the memory usage, but also on whether the extracted MAT satisfies the following properties:



(a) (b) (c)

Figure 2.1 Small change in boundary causes dramatic change in the MAT

1. Connectivity.
2. Rotation invariance.
3. Reconstructivity.
4. Single-pixel width.

Souza and Houghton [69] use the connectivity property of MAT to find the MAT by tracing. In their approach, a starting point of the MAT is found and the eight neighbor points are tested for mediality. They continue to test the next eight neighbor points in turn until no more MAT points can be found. A final refinement of the MAT points achieves greater positional accuracy.

A major drawback of the MAT as shape representation is that small changes in the boundary can produce dramatic changes in the MAT, as shown in Figure 2.1. Badler and Dane [1] smooth the boundary by interpolating a set of points using a parabolic blending technique, and find the MAT from the blended curves. In their method, the boundary will be smooth in the first step, so that small changes in the boundary will only slightly change the blended curve. So, the final MAT will be more stable.

2.2 The MAT from a Geometric Approach

The MA and the Voronoi diagram of a polygon are investigated in [43, 40, 63]. Although an $\mathcal{O}(n \log n)$ algorithm exists for finding the MA of a polygon, there has

been little work extending these algorithm to $\Re^3$. Most of the papers consider Voronoi diagrams for point sets in 3D. Stifter [72] generalized the 3D Voronoi diagram to sets which are open, connected, and bounded. Such sets $\mathcal{F}$ whose boundary are surfaces satisfy a special set of axioms. There axioms are:

There must exist a finite set $contour(\mathcal{F})$ with the following properties:

1. $contour(\mathcal{F})$ is a partition of the boundary of $\mathcal{F}$.

2. Moving on a line normal to some element in $contour(\mathcal{F})$ away from that element the distance to the element increases strictly.

3. Points with equal distances to two elements of $contour(\mathcal{F})$ assume the maximal distance to the boundaries of the elements.

4. Each point in $\mathcal{F}$ is uniquely projectable onto each element in $contour(\mathcal{F})$.

5. All elements in $contour(\mathcal{F})$ are continuously differentiable.

Stifter proved the following theorem:

> For each p, q on the Voronoi diagram of $\mathcal{F}$: If there exists a connected curve from p to q in $\mathcal{F}$, then there exists one on the Voronoi-diagram of $\mathcal{F}$; for all $p \in \mathcal{F}$, there exist a connected curve in $\mathcal{F}$ from p to the Voronoi-diagram of $\mathcal{F}$.

Notice that Stifter has different definition on Voronoi diagram from the Voronoi diagram defined in computational geometry. In computational geometry, the Voronoi diagram of $\mathcal{F}$ is consists of 2D surfaces. In her definition, she call such Voronoi diagram 2-skel of $\mathcal{F}$. The 1-skel of $\mathcal{F}$, which is the Voronoi diagram of $\mathcal{F}$ in her definition, is the intersection curve of two 2-skel surface of $\mathcal{F}$.

The MAT derived from geometric properties of the boundary was investigated in [6, 17, 40, 15, 43, 49, 50, 63]. Preparata [63] and Lee [43] gave algorithms for polygonal 2D domain. Preparata finds the MAT for convex simple polygon and nonconvex simple polygon by different algorithms. He uses edge elimination to derive

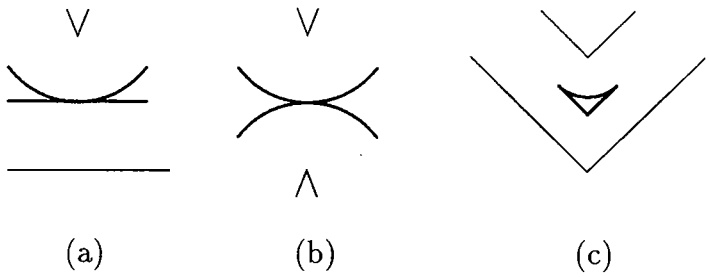


Figure 2.2 Concave vertex eliminate in Preparate's algorithm

the MAT for convex simple polygons. Consider shrinking the boundary of the domain with uniform speed. At some critical distance, one or more offset edge will have zero length and the offset polygon has less boundary edges. The offset polygon is also convex and the MAT segments that lie between the polygon and the offset is easy to find. Continuing this process until the offset polygon has zero area, he finds the MAT for the convex domain. Notice that the MAT of the convex polygon consists of line segments. This algorithm is $O(n \log n)$, where n is the number of the convex edges. For nonconvex simple polygons, Preparata uses edge elimination and concave vertex elimination to derive the MAT. Concave vertex elimination happens when different parts of the shrinking boundary have come into contact. After the contact, the remainder offset figure, which is not a polygon because there may be circular arcs on the boundary, can be either separated into two disjoint figures, as shown in Figure 2.2(a)(b), or one figure with a circular arc as part of its boundary, as shown in 2.2(c). Notice that the concave vertex is no longer influences offset figures, so the total number of edges and concave vertices for the offset figure is reduced. This algorithm is a $O(n^2)$ where n is the sum of the edges and concave vertices of the original polygon.

Lee [43] gave an algorithm for simple polygons that uses the divide and conquer paradigm. He computes the Voronoi diagram of a polygonal domain and deletes the Voronoi edges incident to concave vertices. The algorithm is $O(n \log n)$. Another $O(n \log n)$ algorithm for a subset of the Euclidean plane bounded by one or more

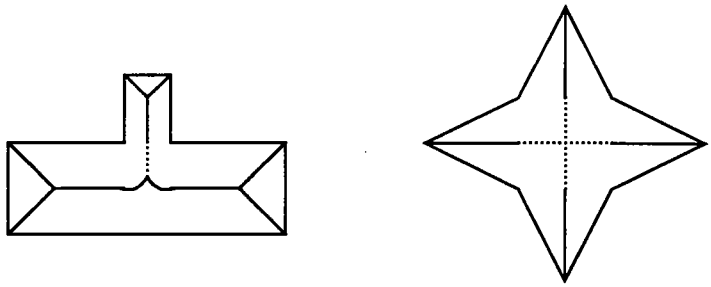


Figure 2.3 Dotted MA cannot be approximated through Delaunay triangulation

polygons, has been proposed by Kirkpatrick [40]. His algorithm is also based on the divide and conquer paradigm. Patrikalakis and Gürsoy [56] generalized Preparata's algorithm to domains bounded by circular arcs and line segments. The MA of such a domain contains line segment, parabolic arcs and elliptic arcs.

Given a set of points, a Delaunay triangulation of these points has the property that every Delaunay triangle does not contain other points of the triangulation. So, if we approximate the boundary by points with sufficient density, and find the Delaunay triangulation of these points. then the centers of the circumscribed circles of the Delaunay triangles that are contained in the domain, are approximate MA points [45, 78]. The exact MA can be found by applying Newton iteration from the approximate MA. Notice that this approach does not find some MA segments [33], as shown in Figure 2.3.

Three-dimensional problems have been investigated [32, 17, 50, 72, 54, 49]. Although we do not know of any algorithm for finding the MAT for convex polyhedra, we believe one can be devised by extending Preparata's algorithm to 3D, that is, by doing edge elimination and face elimination until the remainder offset volume is equal to zero. The MAT for nonconvex polyhedra is more complex because the elimination of concave vertices and concave edges is difficult. The MA for nonconvex polyhedra may consist of faces on planes, paraboloids, parabolic hyperboloid, parabolic cylinder, cone, etc. Notice that if the two concave edges are skew, the offset surfaces locally

associated with these two concave edges becomes tangent at one point, and this makes edge elimination impossible.

The MA for a general surface and space curve are difficult to visualize. For example, consider the volume bounded by $y^2 + z^2/4 = 4 - |x|$, or the space curve which is the intersection of a cylinder and a sphere, their MA points is hard to image. But, with the help of the computer graphic, we develop an algorithm to sketch the MA faces. The algorithm is based on dimensionality paradigm [35] will discuss in Chapter 4.

2.3 The MAT in Descriptive Geometry

Cyclographic maps have been introduced in [48] and used in computer science [34]. Every point in (x, y, z)-space is associated with an oriented circle in the (x, y)-plane by making the point the vertex of a double right cone whose axis is parallel to the z-axis and intersecting the cone with the (x, y)-plane. If the point has a z-coordinate greater than 0, the orientation of the circle is counterclockwise, otherwise, the orientation of the circle is clockwise. Consider an oriented curve C in 2D, and rotate all of its normals 45° about the tangent. Then we obtained a ruled surface called the cyclographic map $S(C)$ of C. The projection of the singular points of the surface onto the (x, y)-plane contains the MA of the curve. The internal MA is produced on one side (the side with z value greater than 0) and the external MA is on the other side.

If we do not care about the orientation of the circles, we can think of the cyclographic maps in another way. Let D be a closed simple domain, and consider the single cones whose apex is on the boundary and whose axis is parallel to the z axis where the interior of the cone is in the $z > 0$ region. Then, the envelope of the cones in the region $z \geq 0$ is a different new cyclographic map. The only difference is that the new map is in the positive z-half space. The MAT is contained in the set of the singular points of the envelope. The internal MAT and the external MAT are all on the same side of the (x, y)-plane. Consider all cones whose apex is on the MAT and

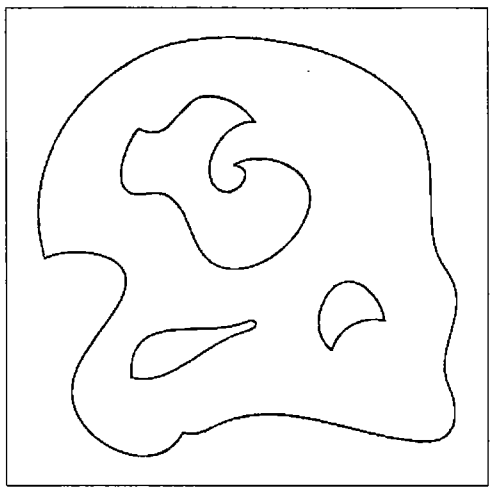
whose axis is parallel to z-axis. The union of all the circles that are the intersection of the cones with the (x, y)-plane are then the original domain D. The plane $z = r$ intersects with the maps produce the local r-offset of the original domain. Notice that every point in the half space $z \geq 0$ corresponds to a right cone in 3D or a circle on (x, y)-plane, which means that three-dimensional half-space can be representing by the set of circles in the plane. This notion that space could be treated as a collection of elements other than points was proposed by Plücker over 100 years ago [62].

We can restrict the surface $S(C)$ such that it is the graph of a function with the (x, y)-plane its domain. With each point(x, y), we associate the minimum $|r|$, so that the point (x, y, r) is on $S(C)$. Let us call this surface the restricted cyclographic map of C. The surface so defined is then the distance surface of Blum [4]. It follows, that the restricted cyclographic map of C can be determined approximately with the Euclidean distance transform.

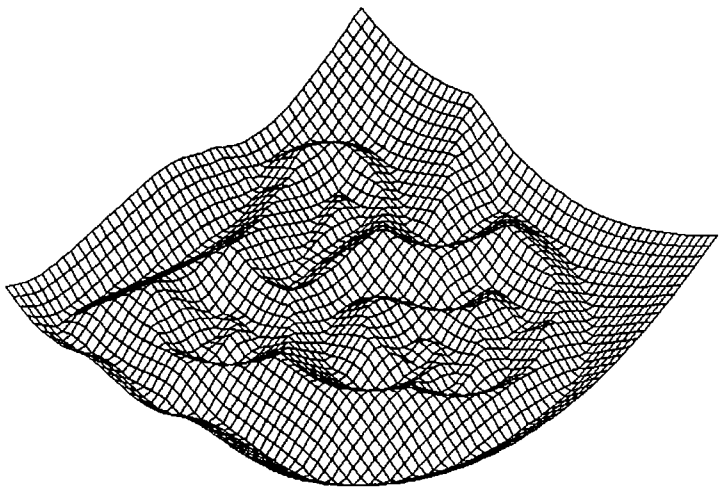
Notice the restricted cyclographic map of C can be obtained from the cyclographic map of C or from the modified cyclographic map of C. The cyclographic map can be easily extended to any closed domain with an oriented curve as its boundary, such as domains with holes. And, the modified cyclographic map can be easily extended to any finite set of curves. Figure 2.4(b) shown an example of the restricted (modified) cyclographic map of the boundary curve of Figure 2.4(a).

2.4 The MAT in Mechanics

Consider a wave front that propagates with uniform speed in a normal direction. At time $t = 0$ assume the front is at the boundary of a close domain T where we assume that the boundary is a smooth curve. The front propagated inward into T, and we denote with $S(x, y)$ the time when the front reaches the point (x, y) for the first time. In [55], it is shown that the MA of T is the shock produced by the propagating wave front. See also [42]. If we can solve the Hamilton-Jacobi equation, that is, $S_x^2 + S_y^2 = 1$ and locate the shocks in the solution, it follows that we can find the MA of the domain T. Similar discuss are in [42]. We can derive the equation of



(a)



(b)

Figure 2.4 A 2D domain and its restricted cyclographic map

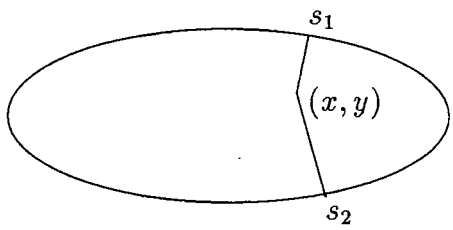


Figure 2.5 The local minimum distance to the boundary

S as follows:

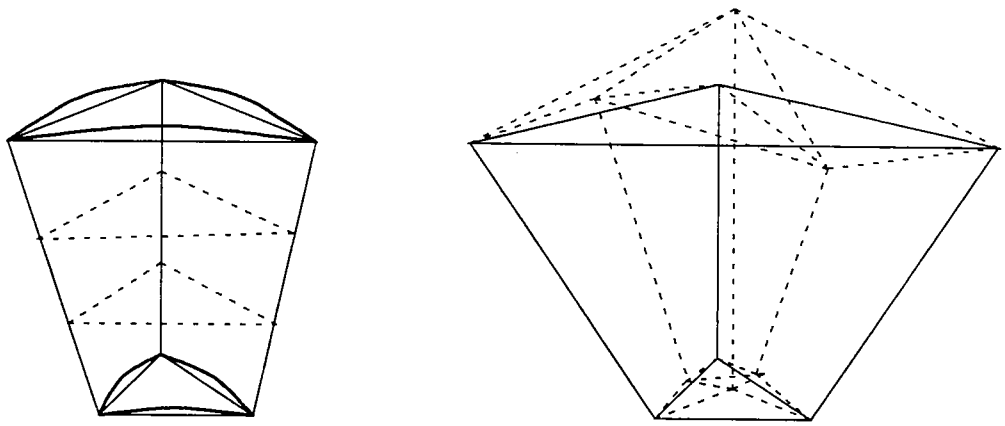
$$\Phi(s; x, y) = (X(s) - u)^2 + (Y(s) - y)^2$$

$$S(x, y) = \min \Phi(s; x, y)$$

where (X(s),Y(s)) is the parametrized closed curve that is the boundary of T, and (x, y) is a point in T. Because $\Phi(s; x, y)$ is a function of s, it will have a number of minima. For example, in Figure 2.5, the local minimum of S occurs at s_1 and s_2 for a fixed point (x, y). If we vary (x, y), we may expect that the global minimum will jump from one local minimum to the other at certain singular points. The closure of these singular points are exactly the shock wave produced by propagating the offset of the 2D contour, they also are the MA for the domain T. The surface $z = \sqrt{S(x, y)}$ is the same as the cyclographic map for the boundary of T.

2.5 Application of the MAT

There are many applications of the MAT of 2D and 3D solids. The interior MAT can be used in shape representation and shape recognition [4, 66, 59, 28, 61, 5], especially for objects whose width is relatively unimportant such as characters in character recognition, or chromosomes in microbiology. The medial axis has also been used in finite-element mesh generation [27, 56, 75, 76]. Many global shape characteristics can be extracted from the internal MAT of the shape, as explained in [27]. The external MAT can be used in motion planning [71]. For example, the



(a) (b)

Figure 2.6 The subdivision for finite-element mesh generation

external MAT of n obstacles can be used to define a collision-free path for robots that maximizes clearance from the obstacles. The MAT is also useful in casting design [11] and has potential applications in geometric tolerancing [34].

To work on finite-element mesh generation from the MAT is straightforward. Assume we have an MAT whose faces and its associated "footpoint faces", are approximated by triangular faces. The finite-element mesh generation can be obtained by meshing each truncated prism. For meshing each component, we have:

1. If the 3 vectors from the MA pointing to their footpoints are bounded by a cone with small angle, then partition the prism into n small components, where n depends on the longest vector of these 3 vectors. See Figure 2.6(a).

2. If the 3 vectors from the MAT pointing to their footpoint are not bounded by a cone with small angle, subdivide the MAT triangle into four triangles and find the associated footpoint faces. Then, repeat step 1 for each subcomponents. See Figure 2.6(b).

Notice that the new points in the subdivision are refined using Newton iteration.

3. THE PROPERTIES OF THE MEDIAL AXIS TRANSFORM FOR 2D AND 3D SOLIDS

We state some properties observed from Figures 1.11 and 1.12 without a proof. They are:

1. An end point of the MA may have only one foot point. For an example, see Figure 1.11(a). The end point may have infinitely many foot points if it is associated with a circular arc.

2. An MA point may be on the boundary of the region, which means that the maximal inscribed circle tangent to the boundary point of the area has radius zero. The origin in Figures 1.12(f)(g) are examples of this case.

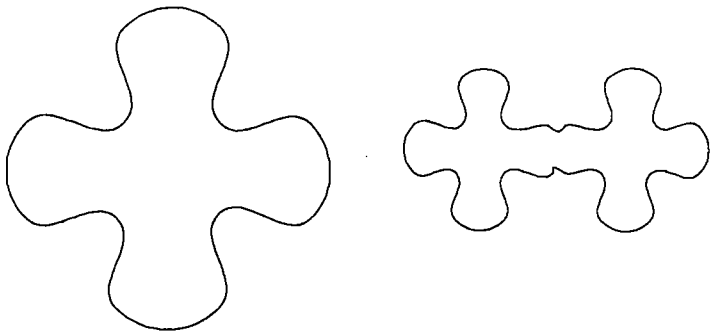
3. A connected unbounded region may have a disconnected MA. The region $f(x, y) < 0$ in Figure 1.12(e) is an example of this case.

4. A symmetric region has a symmetric MA. All pictures in Figure 1.11 and 1.12 are examples of a boundary curve symmetric with respect to the X axis, Y axis or to the origin.

5. If the region is symmetric about the X-axis (or Y-axis), it is possible that no points on X-axis (or Y-axis) are MA point. An example is shown in the region $f(x, y) < 0$ in Figure 1.12(e). Another example is the region $y - \cos x > 0$.

6. If the region is symmetric about a point, this point is not necessarily an MA point. The origin in Figure 1.12(b) is an examples.

7. It is possible that a point belongs to the MA of both the region $f(x, y) > 0$ and $f(x, y) < 0$. The origin in Figure 1.12(d)(g) are examples of this case. Notice here that the origin is the limit point of MA points which have two footpoints.



(a) (b)

Figure 3.1 Initial curve for limit curve

8. Assume the normal for the boundary of the region D points to the interior of D. Then boundary points with a zero curvature radius are also MA points of D. Examples are shown in Figures 1.12(f)(g).

The MAT may be very complex, for example, for the area $y - x \sin \frac{1}{x} > 0$. For such a region, it has probably infinitely many end points and infinitely many branches. Even a closed bounded domain with a structurally simple boundary can have a very complex MAT. For example, consider the curve in Figure 3.1(a) and call it G_1. G_1 has one branch point with four footpoints or two branch points with three footpoints. Consider the curve formed by gluing two half-scaled G_1 curves together, as shown in 3.1(b). Call this curve G_2. Then the curve has at least twice as many branch points as G_1 has. If we continue to glue G_1 onto the right of G_2, scaling the curve properly, we have a sequence of curves G_n, for $n > 0$. Let the curve $G = \lim_{n\to\infty} G_n$. It is easy to see that G is a closed and bounded domain with infinitely many branch points. Such kind of domains are excluded from consideration.

In this chapter, we define the concept of a simple domain in 2D so as to simplify the problem of structural properties of the MAT. There are four sections in this chapter. The first section defines simple domains. The second section proves the

uniqueness, divisibility, connectedness, and reversibility properties of the MAT for simple domains in 2D. It also extends the results to domains whose boundary is not differentiable everywhere. The third section extends the problem to domains with holes and the last section discusses extensions from 2D solids to 3D solids.

3.1 Simple Domain

We define a simple domain D to be a compact domain whose boundary is a simple, differentiable and almost twice differentiable, closed curve of bounded curvature variation. We give definitions for the boundary curve as follows:

Definition 4 A curve in the real Euclidean space $\Re^2$ is a map $\alpha : [a, b] \rightarrow \Re^2$. where $\alpha(t) = (x(t), y(t))$ and $x(t)$, $y(t)$ are continuous functions.

Definition 5 A curve $\alpha : [a, b] \rightarrow \Re^2$ is:

- simple if α is one-to-one.
- differentiable if $d\alpha/dt$ is defined and not zero, for all $t \in [a, b]$, where $d\alpha/dt$ is the derivative of α at t
- almost twice differentiable if its boundary has finitely many points which are not twice differentiable.
- closed if $\alpha(a) = \alpha(b)$.
- of bounded curvature variation if there are finitely many local extrema of the curvature, each of finite value. And, there are finitely many inflection points.

We restrict the boundary of a simple domain to be simple and differentiable, thereby excluding nonmanifold 2D solids and solids with corners. We will generalize simple domains later. Since we require closed boundary curves, simple domains and their MAs are contained in a bounding box. Notice that solids in solid modeling always satisfy the bounded curvature variation property. In particular, this is the case when the boundary of the domain consists of finitely many smooth segments of algebraic curves.

3.2 Uniqueness, divisibility, connectedness, and reversibility of the medial axis transform for 2D solids

Let us define $\mathcal{D}$ to be the set of compact domains whose boundary α is a simple, closed, differentiable and almost twice differentiable curve of bounded curvature variation. Let $\mathcal{S}$ be the set of points (x, y, z) in $\mathcal{R}^3$ where $z \geq 0$. We will interpret the point (x, y, z) as corresponding to the circle in $\mathcal{R}^2$ with center (x, y) and radius z. We further define two operators, $Skel$ and $Skel^{-1}$, where $Skel : \mathcal{D} \longrightarrow \mathcal{S}$ maps the domain $D \in \mathcal{D}$ to its MAT $S \in \mathcal{S}$ and $Skel^{-1}$ maps a set of 3D points (x, y, z) in $\mathcal{R}^3$ where $z \geq 0$, which represent a set of circle in $\mathcal{R}^2$, to the region consisting of the union of all circles and its interior. Notice that $Skel^{-1}(S)$ does not necessarily belong to $\mathcal{D}$. Using these notations, we prove four theorems in this section:

Theorem 3.1 For all $D \in \mathcal{D}$, $Skel(D)$ is unique.

Theorem 3.2 Let $D \in \mathcal{D}$. If there exists a circle $\mathcal{C}$ tangent to D at more than one point, then D can be subdivided into two region D_L and D_R, where the boundary of $D_L \cap D_R$ is the circle $\mathcal{C}$ and $Skel(D) = Skel(D_L) \cup Skel(D_R)$.

Theorem 3.3 If $D \in \mathcal{D}$, then $Skel(D)$ is connected. Furthermore, there are no loops in $Skel(D)$.

Theorem 3.4 Let $D \in \mathcal{D}$ and $S \in \mathcal{S}$, then $Skel^{-1}(Skel(D)) = D$.

3.2.1 The MAT for 2D simple domain is unique

We prove Theorem 3.1 in this section. Some properties of the MAT need to be proved before proving this theorem. Let D be a compact domain whose boundary curve α is a simple, closed, differentiable and almost twice differentiable curve of bounded variation. Intuitively, every boundary point is associated with a unique MA point, and every MA point is associated with one, two, or more boundary points. The MA end points of the domain are associated with a circle arc or a boundary point that is not twice differentiable or has local maximal curvature. We prove these intuitions step by step.

Assume $D \in \mathcal{D}$, and the normal of its boundary curve α points locally to the interior of α. We associate two circles with every point p on the curve α where p has positive curvature. One is the osculating circle of α at p and the other is the circle tangent to α centered at the MA point of which p is a footpoint. We call this circle the MA circle of α at p, or the MAT point of α at p. Note that we have not yet proved the existence and uniqueness of the MA circle.

For any three noncollinear points on α, we can find an unique circle passing through them. As these three points approach p, the circle approaches the osculating circle of α at p [70]. Define $\kappa(p), o(p), OC(p), RO(p)$ to be the curvature, the center of the osculating circle, the osculating circle, and the radius of the osculating circle of α at p. Note that $RO(p) = 1/\kappa(p)$. Consider the osculating circle at a point p and a sufficiently small neighborhood N of curve point p, as illustrated in Figure 3.2. A point in N whose curvature is greater than $\kappa(p)$ will be inside the osculating circle and a point in N whose curvature is less than $\kappa(p)$ will be outside the osculating circle. So, if p has locally maximal curvature in α, then no point of N is inside the circle, so the curve α is locally exterior to $OC(p)$, which implies $OC(p)$ and α intersect at p with even multiplicity, as shown in Figure 3.2(a). Notice that this does not mean that the osculating circle is contained in D, since the radius of the osculating circle at p can be larger than the radius of the MA circle at p. If p has locally positive minimal curvature, the curve α is locally in the interior of $OC(p)$, as shown in Figure 3.2(c). If p has positive curvature which is neither a local maximum nor local minimum, then the osculating circle $OC(p)$ intersects the curve α tangentially with odd multiplicity, so the curve α crosses $OC(p)$ from the outside to the inside, as shown in Figure 3.2(b). Here, the points on α with greater curvature locally are inside of $OC(p)$. The proof of these properties can be found in [73].

Assume that v is the unit normal of the boundary curve α at p and R is the radius of the osculating circle of α at p. Hilbert [12] states that for sufficiently small $\epsilon > 0$, there is a number $\delta(\epsilon) > 0$, s.t., if $r = R - \epsilon$, then there is a δ neighborhood of p that

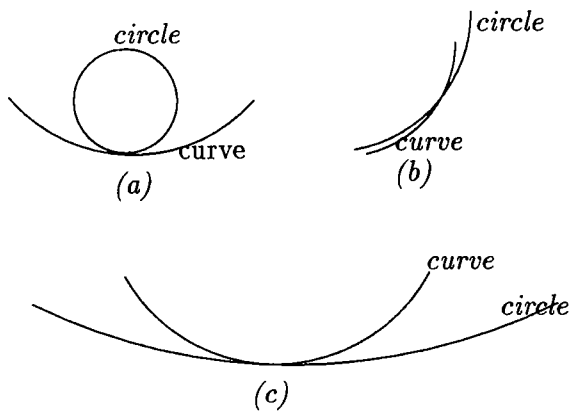


Figure 3.2 Tangent relation between the curve and the osculating circle

is outside the circle $\mathcal{C}(p+rv, r)$; similarly, if $r = R+\epsilon$, then there is a δ neighborhood of p that is inside the circle $\mathcal{C}(p + rv, r)$.

Assume the curve is twice differentiable at p. Let us consider the way we obtain $o(p)$, the center of the osculating circle at p. We consider three noncollinear curve points, and produce a unique circle through them. As these three points approach p, we find the osculating circle at p in the limit. No matter how these three points approach p, we always obtain the osculating circle at p. Consider two points coincident at p and a third point q approaching p, as shown in Figure 3.3. The bisector of the line segment $\overline{pq}$ intersects the normal of p at point o. As q approaches p, o will approach $o(p)$. If the curvature of the curve from q to p is monotonically increasing, then the radius of the approach circle is monotonically decreasing, as shown in Figure 3.3(a). If the curvature of the curve from q to p is monotonically decreasing, then the radius of the approach circle is monotonically increasing, as shown in Figure 3.3(b). Notice that if α is twice differentiable at p and p has neither local minimal nor local maximal curvature, then the osculating circle can be obtained from either side of p in the curve, as shown in Figure 3.3(c). This method of finding the osculating circle and the curvature has been proved in many differential geometry books, e.g, [70]. Now, consider the reverse approach, shown in Figure 3.3(b). If we shrink the osculating circle but keep the circle tangent to α at p, that is, we move the center of the shrinking circle, from $o(p)$ to o', with radius $\overline{o'p}$ in Figure 3.3(b), one of the intersection points of the circle and the curve moves from p to q', and the multiplicity of the circle intersection with the curve at p will become an even number. This is because the curve is now locally outside the circle. By treating tangency as double intersection, when shrinking the osculating circle a little bit, the closed curve α intersects with the circle in at least two other points, as long as p does not have local maximal curvature. This is because two closed curves intersect in an even number of points by the Jordan curve theorem, assuming the intersection multiplicity is properly accounted for.

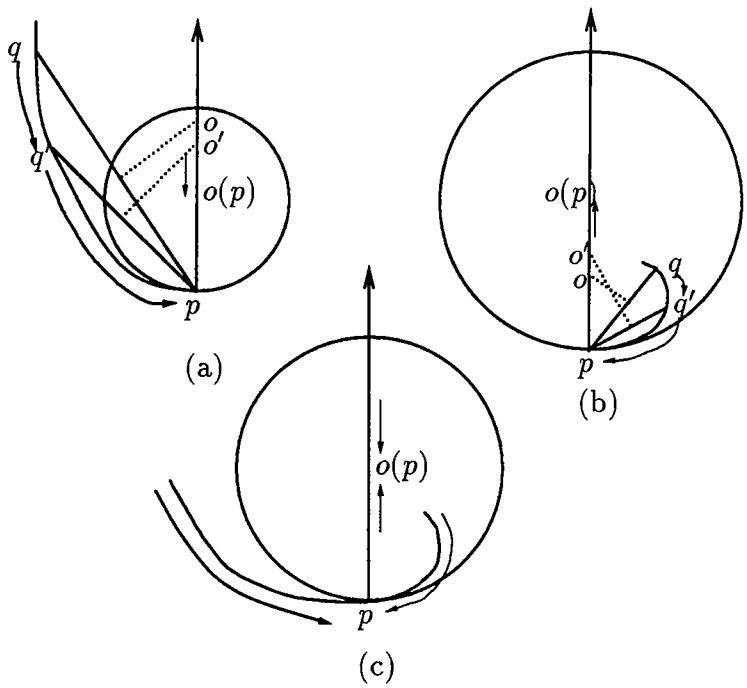


Figure 3.3 Obtaining the osculating circle by point q approaching point p

Now assume that the curve is twice differentiable in a neighborhood of p but is only once continuously differentiable at p. Approaching p from different sides of p will produce different limit circles. Let us define $o^+(p), o^-(p)$ to be the centers of these two "osculating" circles, and let $OC^+(p), OC^-(p)$ be the circles, and $RO^+(p), RO^-(p)$ be the radii of the circles. Without loss of generality, we further assume $RO^+(p) > RO^-(p)$. Note that if α is twice differentiable at p, then $o(p) = o^+(p) = o^-(p), OC(p) = OC^+(p) = OC^-(p)$, and $RO(p) = RO^+(p) = RO^-(p)$.

Let us recall some basic properties of circles: If two different circles intersect, they intersect in two points transversally, or else are tangent in one point. If they intersect in two points transversally, these two points separate a circle into two arcs, one arc is in the interior of the other circle and the other arc is in the exterior. If they are tangent in one point and have a common interior point, then one circle contains the other.

Define $Inside(\mathcal{C})$ as the interior region of a circle $\mathcal{C}$. Under this definition and the definition for MA circle, we establish some basic properties for MA circles.

Lemma 3.5 A boundary point of a simple domain cannot be interior to an MA circle. That is, no MA circle intersects the boundary of a simple domain transversally.

Proof:

Assume α is the boundary curve of the simple domain D. Assume also that p' is a point on α and $\mathcal{C}'$ is a MA circle that contains p' in the interior. Then there exists an $\epsilon > 0$, such that $\mathcal{C}(p', \epsilon) \subseteq \mathcal{C}'$, but $\mathcal{C}(p', \epsilon)$ is not contained D because p' is a boundary point of D, which implies $\mathcal{C}'$ is not contained in D. This contradicts the fact that $\mathcal{C}'$ is a maximal inscribed circle of D. $\diamond$

Lemma 3.6 If two MA circles intersect, then they intersect transversally iff there is a point that is interior to both circles. Otherwise they are tangent, neither containing the other.

Proof:

Assume the two circles have an interior point in common. Then if they are tangent, one contains the other, contradicting maximality. Therefore, the circles intersect transversally. If they have no interior in common, they must be tangent to each other and neither can contain the other. ◇

Lemma 3.7 There is at most one maximal inscribed circle of D which is tangent to p and centered on the normal of α at p.

Proof:

Assume two or more MA circles are tangent to α at p. Then we have two circles of different radii and both with center on the curve normal at p, so one would contain the other. This contradicts Lemma 3.6. So, we proved that if MA circle tangent to α at p exist, it must be unique. ◇

Define $arc_{\mathcal{C}}(p, q)$, where p, q are points on the circle $\mathcal{C}$, as the circle arc from the point p to q orienting the circle counter-clockwise. From this definition, $arc_{\mathcal{C}}(p, q) \neq arc_{\mathcal{C}}(q, p)$, for all $p \neq q$. The arc for the curve α from point p to q is defined similarly and denoted by $arc_{\alpha}(p, q)$.

Lemma 3.8 Let D be a compact domain and its boundary curve α be a simple, closed curve of bounded curvature variation. Assume there exists a circle tangent to α at p and the circle is locally inside the curve in a neighborhood of p. If the curve intersects the circle at a point other than p, then we can find a unique maximal inscribed circle of D that is tangent to α at more than one point. Notice that the center of the MA circle so found cannot be a end point.

Proof:

We prove the existence and uniqueness of the MA circle of D tangent to α at p. Let the circle be $\mathcal{C}(c, R)$. If no point of α is in the interior of $\mathcal{C}(c, R)$, then there exists at

least one point $q \in \alpha$ tangent to $\mathcal{C}(c, R)$ by the Jordan curve theorem. If we enlarge $\mathcal{C}$ to a circle $\mathcal{C}'(c', R')$ where $R' > R$, then the point q must lie inside $\mathcal{C}'$. Therefore, $\mathcal{C}'$ contains exterior points of the domain and is not an MA circle. So, we know $\mathcal{C}(c, R)$ is the maximal inscribed circle of D in this case.

Now, assume there is some point of α inside $\mathcal{C}(c, R)$. Since α is continuous, we can always find two points $q_1, q_2 \in \alpha \cap \mathcal{C}(c, R)$, where q_1 and q_2 are the first and last intersection points of α and $\mathcal{C}(c, R)$ starting from p going counter-clockwise, as shown in Figure 3.4. Let us define a function $f(r') = r' - \min \|c' - q\|$, where c' is the center of the shrinking circle and $q \in arc_\alpha(q_1, q_2)$ and $r' \in [0, R]$. $f(R) > 0$ because parts of α are inside or on the circle $\mathcal{C}(c, R)$. $f(0) < 0$ is trivial. Since f is continuous, by the mean value theorem, there must exist $\overline{r}$, such that $f(\overline{r}) = 0$, which means there exists a circle $\mathcal{C}(\overline{c}, \overline{r})$ tangent to α at p and q where q satisfies $\overline{r} = \min \|\overline{c} - q\|$ for $q \in arc_\alpha(q_1, q_2)$. So, we proved the existence of the MA circle of D tangent to α at p and other points.

The uniqueness of this MA circle is proved in Lemma 3.7. ◇

Lemma 3.9 Let D be a compact domain and its boundary curve α be a simple, closed curve of bounded curvature variation. Assume there exists a line L that passes through the point p, such that the exterior of α is locally on one side of L, as shown in Figure 3.5. Then, we can find a unique maximal inscribed circle of D that is tangent to α at more than one point.

Proof:

Let us draw a line L' starting from p, pointing to the interior of α, and orthogonal to L, as shown in Figure 3.5. Assume that v is the unit vector from p along the line L' and $M = \max \|q_1 - q_2\|$ where $q_1, q_2 \in D$. Since D is bounded, M is finite. Consider the circle $\mathcal{C}(p + Mv, M)$. Clearly the point $p + 2Mv \in \mathcal{C}(p + Mv, M)$ is in the exterior of α. Moreover, there is a neighborhood of p of the circle that is in the interior of α, so α must intersect the circle at at least one other point. By lemma 3.8, there is a unique maximal inscribed circle. ◇

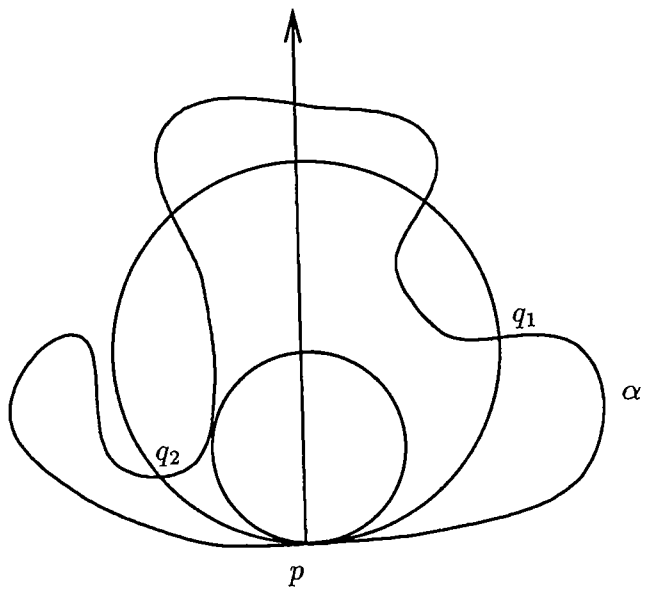


Figure 3.4 The existence of the skeleton circle

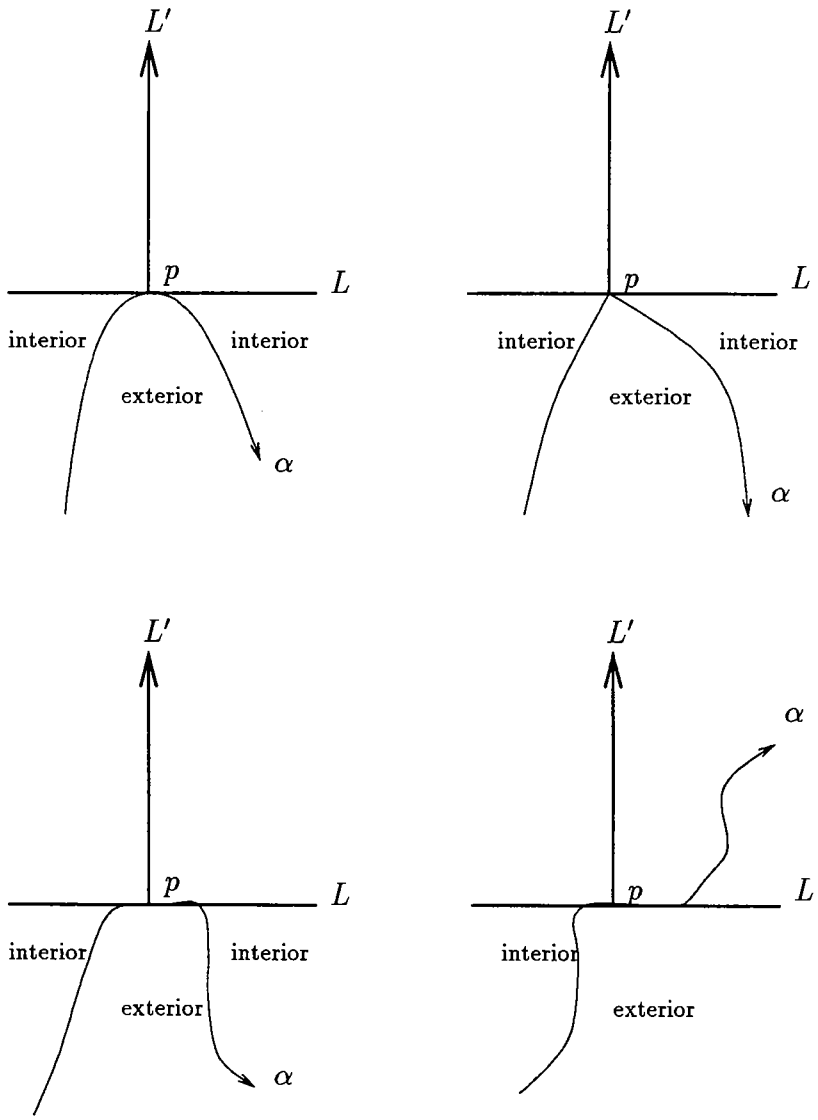


Figure 3.5 The neighborhood of p for the line L is locally inside of the curve α

Notice that if α is differentiable at p, then L is tangent to α at p. Notice also that if α is not differentiable at p, p must be a concave vertex. Otherwise, L does not exist.

Let D be a domain, α be its boundary curve and p be a point on α. Assume the normal of α points to the interior of D, and denote with v the normal of α at p and with $SC(\{p\})$ the set of MA circles of D associated with p. We have the following lemmas:

Lemma 3.10 Let D be a simple domain, α its boundary, and let p be a point of α with a normal that points to the interior of D. If α is differentiable at p, and has a second left and right derivative at p, then $SC(\{p\})$ is a singleton.

Proof:

If the second left and right derivative of α at p are coincident, that is, the curvature of α at p is defined, there are five cases we must consider:

1. $\kappa(p) < 0$.
2. $\kappa(p) > 0$ and α has local maximal curvature at p.
3. $\kappa(p) = 0$.
4. $\kappa(p) > 0$ and the curvature from one side toward p is monotonically decreasing.
5. $OC(p)$ is coincident with α at the left neighborhood or right neighborhood of p. That is, p is on a circular arc if $\kappa(p) \neq 0$, or p is on a line segment if $\kappa(p) = 0$. Notice that p can be end point of this circular point or line segment.

The proof for the first case is directly from lemma 3.9. We can select the tangent line of α at p as the line L in that lemma.

For the second case, $\kappa(p) > 0$ and $OC(p)$ is locally inside of α at p. If $OC(p)$ intersects α only at p, from what Hilbert stated, if we enlarge the circle, the new circle

will be locally outside of α at p. So, we know that $OC(p)$ is a maximal inscribed circle of D, $o(p)$ is an MA end point. By lemma 3.7, $SC(\{p\})$ is singleton. If $OC(p)$ intersects α at p and other point, furthermore, no points of α is in the interior of $OC(p)$, trivially, $OC(p)$ is the maximal inscribed circle of D, $o(p)$ is not an MA end point. And, by lemma 3.7, $SC(\{p\})$ is singleton. If there exist a point q of α in the interior of $OC(p)$, we know α intersect $OC(p)$ at one point other than p because the neighborhood of α at p is in the exterior of $OC(p)$ and q is in the interior of $OC(p)$. By lemma 3.8, $SC(\{p\})$ is singleton.

For the third case, α is tangent to $OC(p)$ and the curvature of α from one side of p toward p is monotonically decreasing, similar to Figure 3.3(b). Let us consider the reverse steps of finding the osculating circle we described before. We can find a point $q' \in \alpha$ inside the $OC^{-}(p)$ and the curvature from q' to p is monotonically decreasing. Then, the circle tangent to α at p and through q' is easy to find. We know the radius of this circle is less than $RO^{-}(p)$, call it $\mathcal{C}(p + rv, r)$ where $r = RO^{-} - \epsilon$ and $\epsilon > 0$. Consider this circle $\mathcal{C}(p+rv, r)$, and from the properties Hilbert stated, it is tangent to α at p and it inside the curve α in the neighborhood of p. Furthermore, it intersects the curve at two points other than p, because the intersection at p has even multiplicity and there is the intersection at q'. So, by the Jordan Curve theorem there is at least one more intersection. By lemma 3.8, we can find a unique MA circle whose center is a normal or a branch point.

There are three subcases in the fourth case. That is, p has local maximal curvature, local minimal curvature, or p is a inflection point. The proof for the subcase that p has local maximal curvature and local minimal curvature are the same as the proof for the first case and the third case, respectively. The proof for the subcase that p is a inflection point is the same as the second or the third case, depending on the curvature from the side which has positive curvature toward p is monotonically increasing or monotonically decreasing.

For the last case, if $\kappa(p) \leq 0$, the proof is directly from lemma 3.9. We can select the tangent line of α at p. If $\kappa(p) > 0$, we know $OC(p) \cap \alpha$ is a continuous circle arc. Except for this intersection, $OC(p) \cap \alpha$ could:

1. have no other intersection.

2. intersect and no point of α is in the interior of $OC(p)$.

3. intersect and some point of α is in the interior of $OC(p)$.

The proof for each subcase is the same as the proof in the second case.

We proved that $SC(\{p\})$ is a singleton if α is twice differentiable at p. Now, we want to prove $SC(\{p\}$ is a singleton if α is differentiable but not twice differentiable. In this case, the left and right derivative of α at p do not agree, and we only consider the side which has the higher curvature. The sign of the curvature is also counted. Consider these two osculating circle $OC^+(p)$ and $OC^-(p)$ with radius $RO^+(p)$ and $RO^-(p)$. Let $\kappa^- = 1/RO^-(p)$, there are five cases we need to discuss:

1. $\kappa^-(p) \leq 0$.

2. $\kappa^-(p) > 0$ and the curvature of α from the negative side toward p is monotonically increasing. In this case, $OC^-(p)$ is locally inside of α on the negative side, it also implies $OC^-(p)$ is locally inside of α on the positive side because $RO^+(p) > RO(p) > 0$.

3. $\kappa^-(p) > 0$ and the curvature of α from the negative side toward p is monotonically decreasing. In this case, $OC^-(p)$ is locally outside of α on the negative side and inside of α at positive side.

4. $\kappa^-(p) = 0$.

5. $OC^-(p)$ is coincident with α at the negative neighborhood of p.

The proof for these five cases is the same as the four case we proved for the twice differentiable boundary point p of α. $\diamond$

By lemma 3.10, every point p on the boundary is associated with exactly one MA circle. Let $RS(p)$ be the radius of the MA circle associated with p, if $RS(p) = RO^-(p)$, and $RO^-(p)$ is tangent to α at only one point, then p is associated with an end point of the MA. Otherwise, p is associated with normal or branch points.

We know the following corollary from the proof in 3.10:

Corollary 3.11 Let D be a simple domain, α its boundary, and let p be a point of α with a normal that points to the interior of D. The center of the unique circle in $SC(\{p\})$ could be an MA end point only if one of the following holds:

1. $\kappa(p) > 0$ and p has local maximal curvature.
2. $\kappa(p) > 0$ and p is on a circular arc.
3. $\kappa^-(p) \geq 0$ and the curvature from the negative side toward p is monotonically increasing.
4. $\kappa(p) = 0$, and p is a inflection point.
5. $\kappa^-(p) > 0$ and $OC(p)$ is coincident with α at the negative neighborhood of p.

We have proved every boundary points is associated with exactly one MA circle. Is every MA circle associated with at least one boundary point? We have the following lemma:

Lemma 3.12 Every MA circle is tangent to the boundary of the simple domain at one or more points.

Proof:

Assume there exists an MA circle $\mathcal{C}(c, r)$ in the simple domain D that is not tangent to α. It is easy to find a larger circle $\mathcal{C}(c, r + \epsilon)$ that contains $\mathcal{C}(c, r)$ and is contained in D. This contradicts the fact that $\mathcal{C}(c, r)$ is a maximal inscribed circle of D. $\diamond$

From the last two lemmas, we conclude that every boundary point is associated with exactly one MA circle and that every MA circle is tangent to the boundary of the domain at one or more points.

Because each boundary point is associated with exactly one MA circle, we have proved theorem 3.1 directly.

Now let us consider a domain with finitely many vertices. Assume T is an area whose boundary is a simple, closed curve of bounded curvature variation. We can easily find a sequence of $T_0, T_1, T_2, \cdots$, so that $T_0, T_1, T_2, \cdots \in \mathcal{D}$ and $\lim_{n\to\infty} T_n = T$. The construction of T_0 is based on finding a small circle tangent to two edges of the vertex p at q, q', and the circle totally inside or outside, depending on whether the vertex is convex or concave, in the area T. We approach q to p by some sequence, and find a corresponding q' to produce $T_1, T_2, \cdots$. It is easy to see that $\lim_{n\to\infty} T_n = T$, as shown in Figure 3.6. From this discussion, we know that if the vertex is convex, the end point for $T_0, T_1, T_2, \cdots$ will approach the vertex itself because the curvature of the circle becomes larger and larger, which means the vertex itself is also an MA point. In the limit, the maximal inscribed circle that correspond to the vertex is centered at the vertex and has radius zero.

If the vertex is concave, we can find a left limit normal and a right limit normal for the vertex, both assumed to be pointing into the interior. Consider a normal line L' between the left normal and right normal, the line L passing through the concave vertex and orthogonal to L' satisfies the assumptions of lemma 3.8. So, this vertex is associated with only one MA circle for each normal between the left limit normal and right limit normal. From this discussion, we have:

Corollary 3.13 The MAT of a domain whose boundary is a simple closed curve of bounded curvature variation is unique.

3.2.2 The MAT for 2D simple domain is divisible

In this section, we prove that the domain $D \in \mathcal{D}$ can be subdivided into two subdomains D_L and D_R, and the union of the MAT for D_L and D_R is the MAT for D. For the proof of the theorem, we rewrite the theorem as:

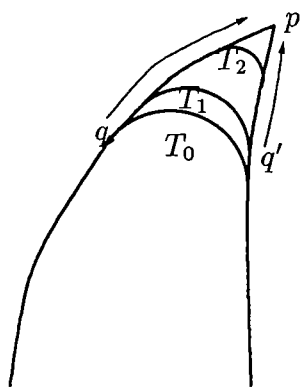


Figure 3.6 Smooth the vertex on the boundary

Theorem 3.14 Let $D \in \mathcal{D}$ and its boundary $\alpha : [a,b] \rightarrow \Re^2$ be a simple, closed, differentiable curve of bounded curvature variation. If there exists an inscribed circle $\mathcal{C}(c,r)$ with center c and radius r, tangent to α in at least two points $p = \alpha(t_1)$ and at $q = \alpha(t_2)$, where $p \neq q$, as shown in figure 3.7, then the points separate α into two curves β_L and β_R and separate the circle $\mathcal{C}$ into $\mathcal{C}_L$ and $\mathcal{C}_R$. Choose p and q so that $\mathcal{C}_R$ contains no boundary point of D. Assume p and q are in $\mathcal{C}_L$ and $\mathcal{C}_R$ and not in β_L and β_R. With α_L the union of β_L and $\mathcal{C}_R$ and α_R the union of β_R and $\mathcal{C}_L$, and with D_L the region bounded by α_L and D_R the region bounded by α_R, we have:

1. α_L and α_R are simple, closed, differentiable curves of bounded curvature variation.
2. $Skel(D_L) \subseteq Skel(D)$ and $Skel(D_R) \subseteq Skel(D)$.
3. $Skel(D_L) \cap Skel(D_R) = \{(x,y,z) |$ where $c = (x,y)$ and $r = z\}$.
4. $Skel(D_L) \cup Skel(D_R) = Skel(D)$.

From Theorem 3.14, we know the MA of the domain is cut by an MA circle and the center of the circle connects the pieces. So we have the following corollaries;

Corollary 3.15 MA circles cut the MA into separate pieces connected at the center, and these pieces can be constructed separately by considering only the subdomain bounded by α_R.

Corollary 3.16 Let $line(p)$ be the line segment from the boundary point p of D to its associated MA point. If p and q are two boundary points of D and $p \neq q$, then $line(p)$ and $line(q)$ do not intersect except possibly at the same MA point.

Lemma 3.17 Assume the hypotheses of Theorem 3.14. Let $\mathcal{C}'$ be any other MA circle of D. If $\mathcal{C}'$ intersects $\mathcal{C}_R$, then $\mathcal{C}'$ cannot intersect β_L. Moreover, $\mathcal{C}'$ is tangent to at least one point of β_R.

Proof:

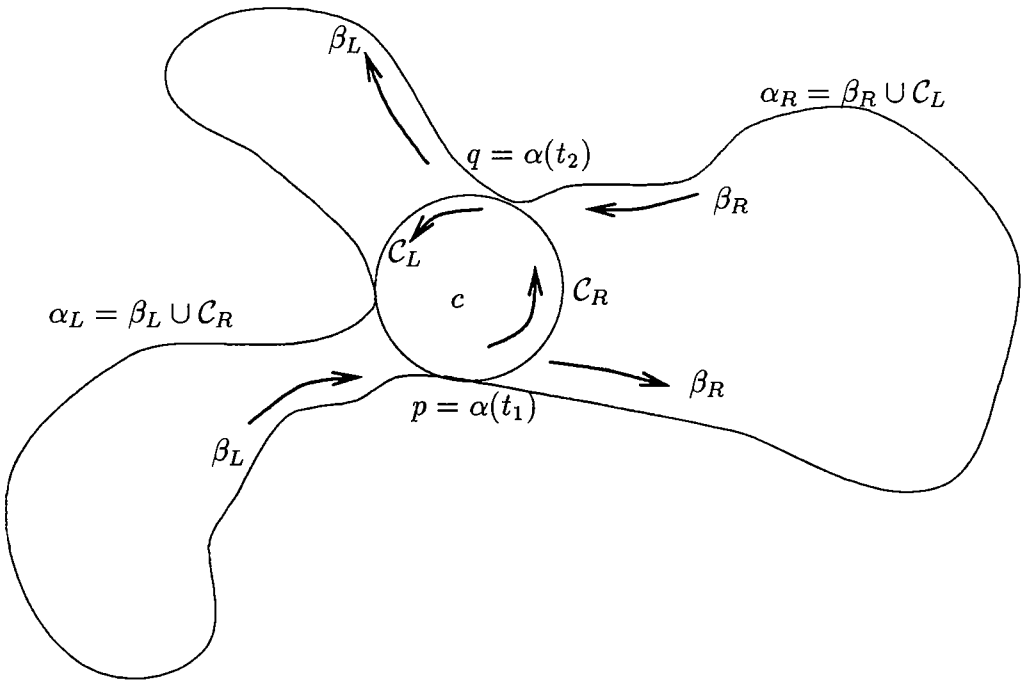


Figure 3.7 An MA circle of the curve α

By lemma 3.5, we know that if $\mathcal{C}$ and $\mathcal{C}'$ intersect, they intersect transversally or are tangent to each other, and neither contains the other. Assume β_L is tangent to $\mathcal{C}'$ at p' and $\mathcal{C}$ and $\mathcal{C}'$ are on the left side of β_L, then we can always find a point $q' \in \mathcal{C} \cap \mathcal{C}'$ so that $arc_{\beta_L}(p', p), arc_{\mathcal{C}}(q', p)$ and $arc_{\mathcal{C}'}(p', q')$ bound a closed region, as shown in Figure 3.8. Notice that $p' = q'$ in Figure 3.8(c). There are four cases:

1. β_L never leaves the bounded region. That contradicts the fact that D is compact because $D_L \subseteq D$ is not compact in this case.

2. β_L intersects itself. This contradicts that α is simple.

3. β_L goes into $\mathcal{C}$ or $\mathcal{C}'$. This contradicts the lemma 3.5.

4. β_L goes through the tangent point of $\mathcal{C}$ and $\mathcal{C}'$. This contradicts that both $\mathcal{C}$ and $\mathcal{C}'$ are on the left side of β_L.

Therefore, β_L cannot intersect $\mathcal{C}'$ if $\mathcal{C}'$ intersects $\mathcal{C}_R$.

If $\mathcal{C}'$ is not tangent to any point of β_R, obviously it is not a maximal inscribed circle of α, which contradicts the fact that $\mathcal{C}'$ is an MA circle. $\diamond$

Lemma 3.18 Assume the hypotheses of Theorem 3.14. If $\mathcal{C}'$ is tangent to β_R, then it cannot be tangent to β_L.

Proof:

Let $\mathcal{C}'$ be an MA circle that does not intersect $\mathcal{C}$, then $\mathcal{C}'$ will be in the region bounded by $\mathcal{C}_R$ and β_R or in a bounded by $\mathcal{C}_L$ and β_L. If $\mathcal{C}'$ is bounded by $\mathcal{C}_R$ and β_R, it must be tangent to β_R only. In the other case, it must tangent to β_L only. If $\mathcal{C}'$ intersects $\mathcal{C}$, the results follows from lemma 3.17. $\diamond$

We can now prove the theorem 3.14 with the following lemmas:

Lemma 3.19 Assume the hypotheses of Theorem 3.14, then α_L and α_R are simple, closed, differentiable curves of bounded curvature variation.

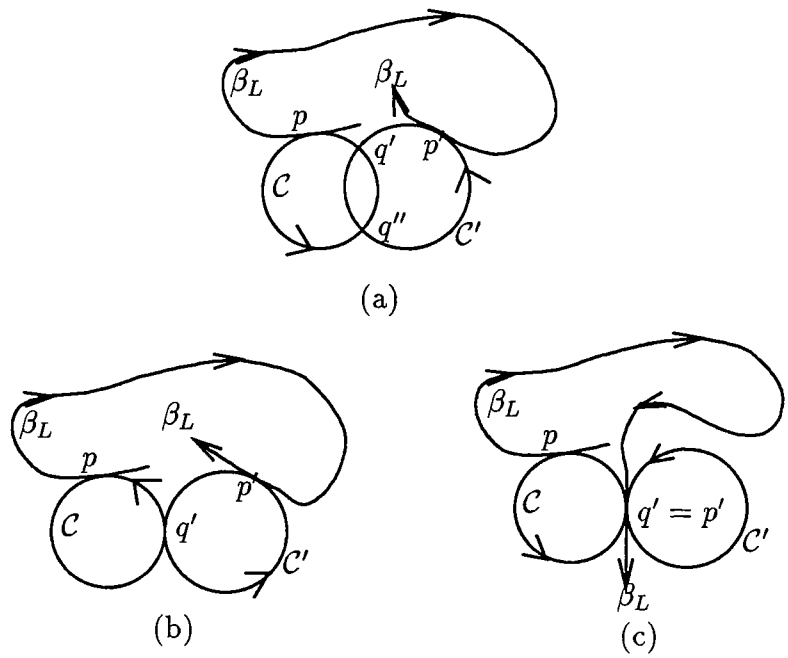


Figure 3.8 The curve β_L cannot be tangent to both C and C'

Proof:

Without loss of generality, we assume $a \leq t_1 < t_2 \leq b$. α is a simple, close regular curve.

$\Rightarrow$ $d\alpha/dt \neq 0, \forall t \in [a,b], \alpha$ is one-to-one function on the domain $[a,b)$ and $\alpha(a) = \alpha(b)$.

$\Rightarrow$ Let f be the one-to-one function which maps the interval $[t_1,t_2]$ onto the circle arc $\mathcal{C}_R$, then α_L can be defined by:

$$\alpha_L(t) = \begin{cases} \alpha(t) & \text{if } t \in [a,t_1) \text{ or } t \in (t_2,b] \\ f(t) & \text{if } t \in [t_1,t_2] \end{cases}$$

$\Rightarrow$ $d\alpha_L/dt = d\alpha/dt \neq 0 \ \forall t \in [a,t_1) or (t_2,b]$. $d\alpha_L/dt \neq 0 \ \forall t \in (t_1,t_2)$ because every point in the arc of circle is differentiable. Since the circle $\mathcal{C}(c,r)$ is tangent to α at $\alpha(t_1)$ and $\alpha(t_2)$, $\mathcal{C}_R(c,r)$ has the same tangent line as α at $\alpha(t_1)$ and at $\alpha(t_2)$, which means $d\alpha_L/dt$ at t_1 and at t_2 exists, and proves that α_L is regular. We want to prove for all $r_1, r_2 \in [a,b)$, if $r_1 \neq r_2$, then $\alpha_L(r_1) \neq \alpha_L(r_2)$. if $r_1, r_2 \in [a,t_1)$ or (t_2,b) and $r_1 \neq r_2$, then $\alpha_L(r_1) \neq \alpha_L(r_2)$ because β_L is one-to-one. If $r_1, r_2 \in [t_1,t_2]$ and $r_1 \neq r_2$, $\alpha_L(r_1) \neq \alpha_L(r_2)$ because f is one-to-one. if $r_1 \in [a,t_1)$ or (t_2,b), and $r_2 \in [t_1,t_2], \alpha_L(r_1) \neq \alpha_L(r_2)$ because $\alpha_L(r_1) = \alpha(r_1) \in \beta_L$, $\alpha_L(r_2) = f(r_2) \in \mathcal{C}_R$ and $\beta_L \cap \mathcal{C}_R = \phi$. So, we proved that α_L is a simple curve. With the property $\alpha_L(a) = \alpha(a) = \alpha(b) = \alpha_L(b)$, we proved that α is a closed curve. That α_L has bounded curvature variation is trivial from the fact that if α_L is not of bounded curvature variation, then β_L is not of bounded curvature variation, α is not of bounded curvature variation either. $\diamond$

Lemma 3.20 Assume the hypotheses of Theorem 3.14, then $Skel(D_L) \subset Skel(D)$ and $Skel(D_R) \subset Skel(D)$.

Proof:

If $\mathcal{C}' \in Skel(D_L)$ is an MA circle that is tangent to $\mathcal{C}_R$and is not $\mathcal{C}$, then this contradicts to the fact that $\mathcal{C}'$ is maximal if $\mathcal{C}'$ is inside $\mathcal{C}$ and contradicts to the fact that $\mathcal{C}'$

is an inscribed circle if $\mathcal{C}$ is inside $\mathcal{C}'$. So $\mathcal{C}'$ must be equal to $\mathcal{C}$. Hence, there is no MA circle $\mathcal{C}'$ of α_L, where $\mathcal{C}' \neq \mathcal{C}$, that is tangent to $\mathcal{C}_R$. If $\mathcal{C}' \neq \mathcal{C}$, then $\mathcal{C}'$ is associated with one or more points of β_L. With $\mathcal{C}' \in Skel(D_L)$, $\mathcal{C}'$ is associated with one or more points in β_L or $\mathcal{C}' = \mathcal{C}$, so we proved $\mathcal{C}' \in Skel(D)$. Therefore, $Skel(D_L) \subseteq Skel(D)$. The proof of $Skel(D_R) \subseteq Skel(D)$ is analogous. $\diamond$

Lemma 3.21 Assume the hypotheses of Theorem 3.14, then $Skel(D_L) \cap Skel(D_R) = (x, y, z)$ where $c = (x, y)$ and $r = z$.

Proof:

Assume that $(x', y', z') \in Skel(D_L)$ where $c' = (x', y')$ is the center of the MA circle $\mathcal{C}'$ and $z' = r'$ is its radius. Assume further that c' is in the interior of the sector bounded by $\mathcal{C}_R, \overline{qc}$, and $\overline{cp}$; see also Figure 3.7. Since $\mathcal{C}'$ is closer to some point l on $\mathcal{C}_R$ than to either p or q, c' is associated with an interior point of $\mathcal{C}_R$. Then $\mathcal{C}'$ cannot be maximal. If c' is on the segment $\overline{pc}$ or $\overline{qc}$, but is not c, a similar argument shows that the associated MA circle is not maximal. Therefore, the center of the MA circles in D_L are in the region bounded by $\beta_L, \overline{pc}$, and $\overline{cq}$, excluding the segment $\overline{cp}$ and $\overline{cq}$, but not the point c. By symmetry, the center of the MA circles in D_R are in the region bounded by $\beta_R, \overline{qc}$, and $\overline{cp}$, excluding the two segments $\overline{cp}$ and $\overline{cq}$ but not c. These regions intersect only in c. Clearly, $(x, y, z) \in Skel(D_L)$ and $(x, y, z) \in Skel(D_R)$, which establishes the lemma. $\diamond$

Lemma 3.22 Assume the hypotheses of Theorem 3.14, then $Skel(D_L) \cup Skel(D_R) = Skel(D)$.

Proof:

$Skel(D_L) \cup Skel(D_R) \subseteq Skel(D)$ by Lemma 3.20. Assume $\mathcal{C}'((x', y'), z')$ be an MA circle of D and $(x', y') \neq c$. By lemma 3.18, $\mathcal{C}'$ must be associated with one or more points in β_L, which implies $(x'y'z') \in Skel(D_L)$ or $\mathcal{C}'$ must be associated with one or more points in β_R, which implies $(x', y', z') \in Skel(D_R)$. Therefore, if $(x', y', z') \in$

$Skel(D)$, then $(x', y', z') \in Skel(D_L) \cup Skel(D_R)$, which means $Skel(D) \subseteq Skel(D_L) \cup Skel(D_R)$. $\diamond$

Corollary 3.23 An MA circle of a simple domain whose center is not an end point splits the MA into a finite number of connected parts, each in turn the MA of a simple domain.

We now consider the subdivision problem for a domain whose boundary has vertices. Notice that the differentiability of the boundary was only used in the proof of Lemma 3.19. Therefore, we have the following:

Corollary 3.24 Let T be a domain whose boundary is closed, simple curve of bounded curvature variation. If there exists a circle tangent to T at more than one point, then T can be subdivided into two region T_L and T_R. The boundaries of T_L and T_R are also closed, simple curves of bounded curvature variation. Furthermore, the boundary of $T_L \cap T_R$ is a circle and $Skel(T) = Skel(T_L) \cup Skel(T_R)$.

3.2.3 The MAT for simple domain is connected with tree structure

In this section, we want to prove that the MAT for simple domain is connected with tree structure. After that, we would like to expand the theory to the domains with finitely many vertices on the boundary. We prove theorem 3.3 from the following three lemmas:

Lemma 3.25 Let α be a oriented continuous open curve whose curvature is monotonically increasing or monotonically decreasing. There is no oriented circle tangent to α at two or more points with the same orientation as *alpha* at all points, such that the interior of the circle contains no point of α.

Proof:

Let p and q be two points at which the sought circle $\mathcal{C}_0$ is tangent to α. Consider the osculating circle $\mathcal{C}$ at p. Because the curvature of α is monotone, the segment of α

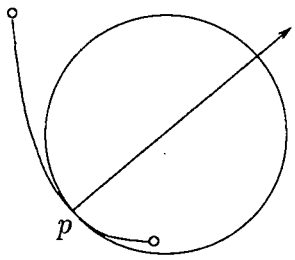


Figure 3.9 Only one circle tangent to the curve with strictly monotone curvature

with higher curvature than the curvature at p will be in the interior of $\mathcal{C}$. Since $\mathcal{C}_0$ is assumed to contain no point of α, it must be smaller than $\mathcal{C}$. But then q cannot be a point of lower curvature than p (see Figure 3.9) because all those points are outside $\mathcal{C}$. So the curvature of α at q is greater than at p. By symmetry, $\mathcal{C}_0$ must be smaller than the osculating circle at q. But then p would be outside $\mathcal{C}_0$ as it has lower curvature. Therefore, $\mathcal{C}_0$ does not exist. $\diamond$

Lemma 3.26 The MAT of the domain $D \in \mathcal{D}$ is connected.

Proof:

We would like to prove the MA of D is connected first and then prove the MAT of D is connected.

For a simple domain, we can subdivide the domain into many subdomains via the MA branch point, so that each subdomain contains no branch point. If the MAT of each subdomain is connected, so is the original domain. So, we can assume the domain D does not contain MA branch points in this proof.

Let $m(p)$ be the MA point associated with the boundary point p and $n(p)$ be the normal of α at p. We want to prove $m(p)$ is connected for all $p \in \alpha$. Notice that $m(p)$ is always on $n(p)$.

Assume the MA for D is not connected at p and we want to prove the contradiction. Without loss of generality, we further assume that $m(p)$ is not continuous from its

right side, that is $\lim_{x \to p^+} m(x) \neq m(p)$. Notice that if p is on a circular arc, $m(p)$ is continuous because $m(p) = m(p^+) = m(p^-)$.

We know $\lim_{x \to p^+} m(x)$ is on $\lim_{x \to p^+} n(x) = n(p)$ because α is differentiable.

Assume $m(p) - \lim_{x \to p^+} m(x) = \epsilon$ where $\epsilon > 0$, as shown in Figure 3.10(a). We consider a boundary curve segment from p to its neighbor points $p + \epsilon'$, where the curvature of this curve segment is monotonically increasing or monotonically decreasing. Because $m(p) - \lim_{x \to p^+} m(x) = \epsilon$, and by the previous lemma, we know the MA point $m(p')$, where p' is a point between the point p and $p+\epsilon'$, associated with only one boundary p'. But, the point p' does not have local maximal curvature or is not on the circular arc. This contradicts the fact that MA end points are associated with points on a circular arc or points with local maximal curvature.

Now consider the case that $\lim_{x \to p^+} m(x) - m(p) = \epsilon$, where $\epsilon > 0$, as shown in Figure 3.10(b). The circle $\lim_{x \to p^+} \mathcal{C}(m(x), \|m(x) - p\|)$ is contained in D because D is a compact domain, and is contains the circle $\mathcal{C}(m(p), r)$. So, we find that $\mathcal{C}(m(p), r)$ is not a maximal inscribed circle of D.

From the above discussion, we proved $\lim_{x \to p^+} m(x) = m(p)$. That is, the MA points for D are connected.

Now, assume the MA points for D are connected and the MAT points for D are not connected. There is one MAT point (x, y, z) that is not connected but after projection to the $z = 0$ plane, the MA of D is connected at (x, y). So, there is one branch of the MAT points approaching the point (x, y, z') where $z' \neq z$, so that after the projection, the MA point of D is connected. Therefore, one of these two MAT points (x, y, z) and (x, y, z') does not have an associated MA circle that is a maximal inscribed circle of D. So, we proved the MAT of D is connected. $\diamond$

We proved the connectedness property for $D \in \mathcal{D}$. That is, $m(p)$ is continuous function for $p \in \alpha$. Now we like to know the differential property of the radius function. Let r be the radius of the MA circle and θ be the angle between the tangent of the MA and the line connecting the MA point and its associated boundary point, as shown in Figure 3.11. Assume the MA is parametrized by arc length. Blum

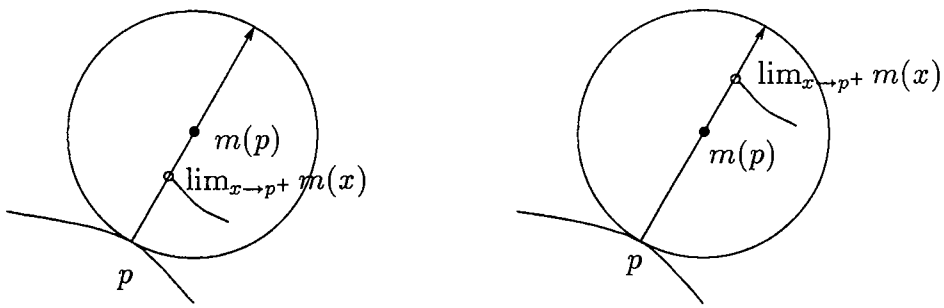


Figure 3.10 The connectedness of the MAT of simple domain

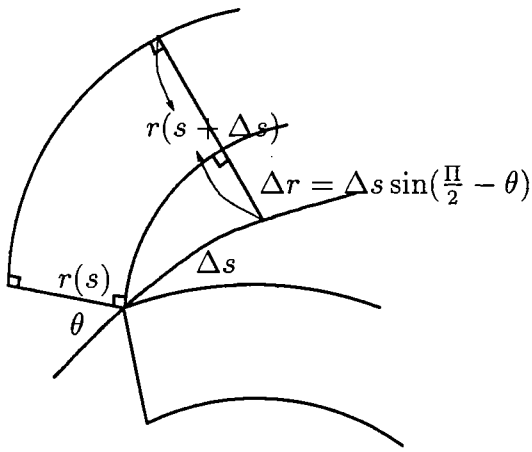


Figure 3.11 The derivative of the radius function

[4] proved:

$$\frac{dr}{ds} = \lim_{\Delta s \to 0} \frac{r(s+\Delta s) - r(s)}{\Delta s} = \lim_{\Delta s \to 0} \frac{\Delta r}{\Delta s} = \sin(\frac{\Pi}{2} - \theta) = \cos\theta$$

if the MA point with parameter s is not a branch point.

Lemma 3.27 The structure of the MAT for the domain $D \in \mathcal{D}$ has no loop.

Proof:

Let S be the set of boundary points which have extremal curvature, or are inflection points, or are end points of a circular arc of the simple domain D. Then, all points in S separate the curve into curve segments, such that the curvature of each curve segment is monotonically increasing, monotonically decreasing, or constant. Let $S' = \{p | p \in S$ and p is associated with a branch or normal MA point$\}$. We want to prove that the MA of D has no loop by induction on the number of elements in S'. Then, we prove the MAT of D has no loops.

1. Basis: $|S'| = 0$

Consider a simple domain D whose associated set S' is empty. Then, α, which is the boundary of D, is constructed by m curve arcs which are monotonically increasing

or monotonically decreasing, and m circular arcs as shown in Figure 3.12(a)(b). The length of a circular arc can be zero as in Figure 3.12(a). We want to prove the MA of D has no loop when $m > 1$. Assume there is a loop in the MA of D. Consider two MA points a_0 and a_1 on the loop, each with one footpoint, a_0' and a_1' respectively, on the same noncircular curve segment, as shown in Figure 3.12(c). The MA points a_0 and a_1 separate the MA loop into two arcs, $arc_{MA}(a_1, a_0)$ and $arc_{MA}(a_0, a_1)$. Consider an MA point $a_2 \in arc_{MA}(a_1, a_0)$ which is the midpoint for a_0 and a_1. Because a_2 is in the region bounded by $arc_\alpha(a_1', a_0')$, line segment $\overline{a_0' a_0}$, $arc_{MA}(a_0, a_1)$ and line segment $\overline{a_1 a_1'}$, any line segment with end point a_2 and a point $a_2' \in \alpha$ will be characterized by one of the following cases:

1. a_2' is on $arc_\alpha(a_1', a_0')$.

2. The line segment $\overline{a_2 a_2'}$ intersect $\overline{a_0' a_0}$ or $\overline{a_1 a_1'}$. If a_2' is the footpoint of a_2, this contradicts Corollary 3.16.

3. The line segment $\overline{a_2 a_2'}$ intersect $arc_{MA}(a_0, a_1)$ at a point q. If a_2' is the footpoint of a_2, this contradicts with the fact that the MA circle center at q is a maximal inscribed circle of D.

Thus, we know the footpoints of a_2 must be on the boundary of $arc_\alpha(a_1', a_0')$. If a_2 has more than one footpoint on the $arc_\alpha(a_1', a_0')$, we find a circle tangent to the open curve $arc_\alpha(p_1, p_0) - \{p_1, p_0\}$ at two points, and the curvature of this open curve is monotonically increasing or monotonically decreasing. This contradicts Lemma 3.25. We find that each MA point in $arc_{MA}(a_1, a_0)$ is associated with only one boundary point, which means there are infinitely many MA end points in D. This contradicts the fact that D has finitely many MA points. The proof that a simple domain has finitely many MA end points is directly from Corollary 3.11 the definition of the simple domain.

So, we proved that the MA of D has no loop when $|S'| = 0$.

2. Hypothesis: $|S'| < n$

Assume the MAT of the simple domain D whose associated S' has less than n element has no loop.

3. Induction: $|S'| = \mathrm{n}$

Let us subdivide the domain into two subdomains via a point $p \in S'$. After subdividing the domain, D_L and D_R are simple domains and each subdomain has an associated set S' that has less elements than n. From Theorem 3.15, any path from an MA point of D_L other than p to an MA point of D_R other then p must pass through p. Because D_L and D_R has no loop, so does D.

So, we have proved the MA of D has no loop by induction. Now, we want to prove the MAT of D has no loop. Let $M = \{(x, y, z)|z \geq 0\}$ be the set of MAT of D. If S has loops, the MA of D which is $M' = \{(x, y)|(x, y, z) \in M\}$ must have loops. This contradicts with the fact that MA of D has no loop. $\diamond$

The proof of Theorem 3.14 is directly from Lemma 3.26 and 3.27.

Now we want to extend the connectedness problem to domains whose boundaries have vertices. The proof for a convex vertex is similar to the proof in Theorem 3.26. Here, $\lim_{x \to p^+} m(x) - m(p) = 0$ is trivial because both of the radius associated with the boundary p which is $\|p - m(p)\|$ and the limit radius associated p which is $\lim_{x \to p^+} \|m(x) - p\|$ are equal to zero.

For the concave case, we know for every direction between p's left limit normal and right limit normal, there is only one medial axis point associated with p. Let $m(p, \theta)$ be the MA point associated with p and on the line through p rotated from the right limit normal counterclockwise by θ degrees. We want to prove that $\lim_{x \to \theta}(m(p, x)) = m(p, \theta)$, where θ is between 0 and the angle between p's right limit normal and left limit normal. The proof is similar to the proof in Theorem 3.26. The proof that the MAT of such domain has no loop is similar to the proof in Lemma 3.27. So, we have:

Corollary 3.28 The MAT for a domain whose boundary is a simple, closed curve of bounded curvature variation is connected with tree structure.

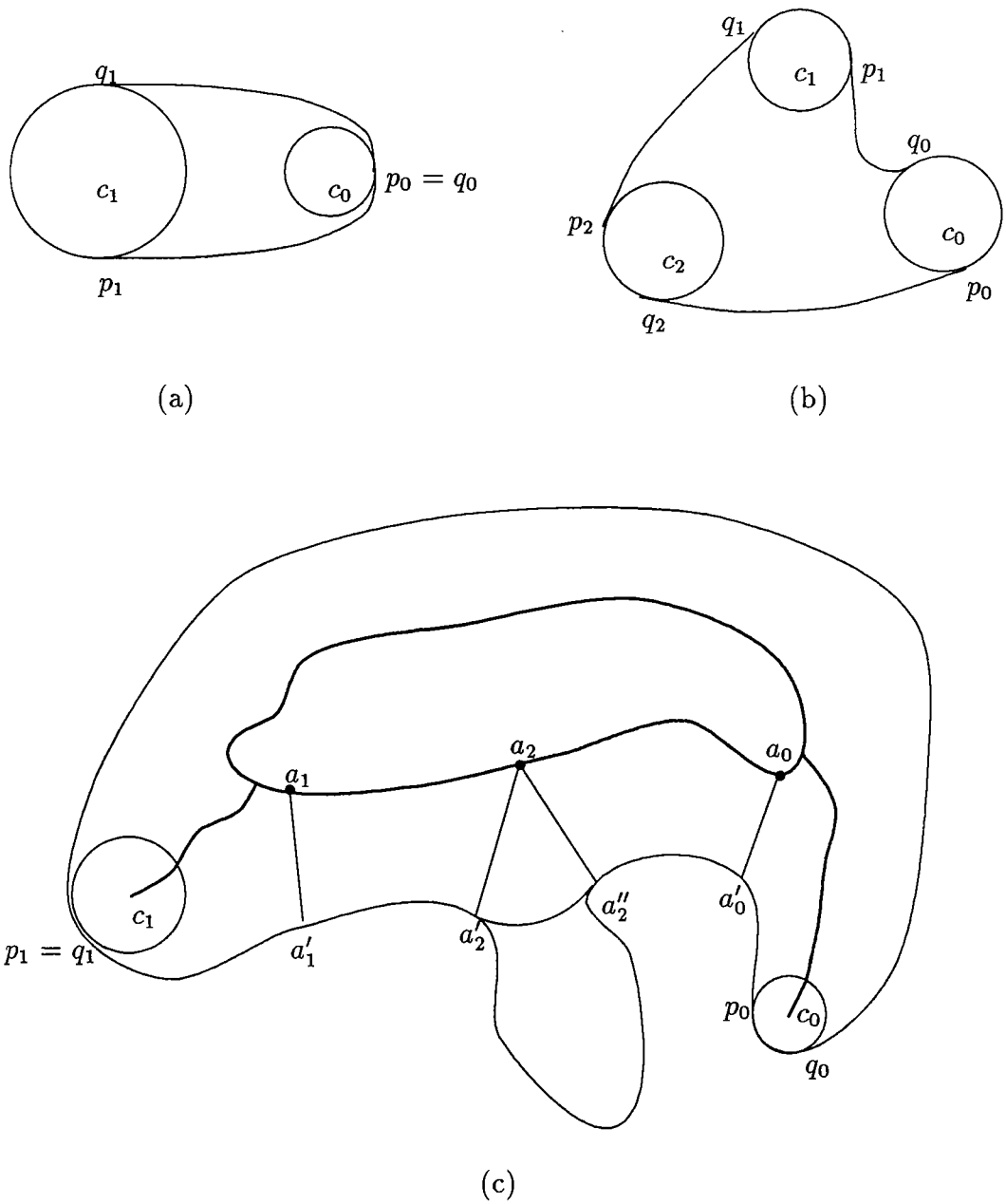


Figure 3.12 The MA for the simple domain has no loop

3.2.4 The simple domain is recoverable from the MAT

We prove theorem 3.4 in this section. Duda and Hart [16] proved $Skel^{-1}(Skel(D)) = D$ informally, and our proof is based on their informal ideas.

Assume $d \in D$. We can find a footpoint of d on the boundary of D, call it d'. From the previous sections, we know d' is associated with a MA circle in $Skel(D)$. It is trivial that d is contained in this MA circle and that the MA circle is contained in $Skel^{-1}(Skel(D))$, so $d \in Skel^{-1}(Skel(D))$. We proved $D \subseteq Skel^{-1}(Skel(D))$. Assume $d \in Skel^{-1}(Skel(D))$, d is contained in a circle of $Skel(D)$ and this circle is contained in D, so $d \in D$. we proved $Skel^{-1}(Skel(D)) \subseteq D$. So, $Skel^{-1}(Skel(D)) = D$. ◇

3.3 The MAT for domain with hole, multiple shell domain, and nonmanifold domain

We consider more complicated domains in this section. For example, domains with holes, multiple shell domains and nonmanifold domains. Multiple shell domain can be considered as union of different domains with or without holes, which reduces the problem to a simpler one. For example, the domain in Figure 3.13 can be considered as three simpler domains which have zero, one and two holes respectively.

For the nonmanifold domain, we need to pay special attention to nonmanifold vertices. Notice that smoothing the nonmanifold vertex and finding the MAT for the domain in the limit is not going to work here. For example, consider two ellipses tangent to each other, as shown in Figure 3.14(a). We smooth the nonmanifold vertex so that the resulting boundary curve bounds a manifold region. Let D be the domain bounded by these two ellipses, there is a sequence of domain $D_0, D_1, D_2, \cdots$, all of them are manifolds, and $\lim_{n\to\infty} D_n = D$. Notice that the boundary is simple, closed and differentiable for each D_n. From theorem 3.3, the MAT for each manifold domain D_i is connected. But, the MAT for D is not connected. From this discussion we have:

Corollary 3.29 The MAT for nonmanifold domain is not necessarily connected.

Figure 3.13 The multiple shell domain

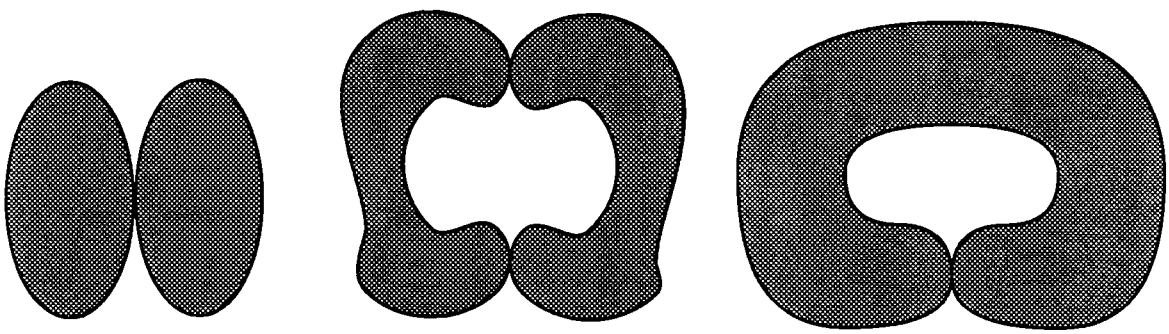


Figure 3.14 The nonmanifold domain

Because the nonmanifold domain cannot be analyzed with the limit approach, we use a different approach as follows:

- If the domain can be separated into two or more pieces at the nonmanifold vertices, such as the domain in Figure 3.14(a)(b), then we consider the subdomains separately.

- In the case that the domain cannot be separated at the nonmanifold vertices, such as the nonmanifold domain in Figure 3.14(c), we can find an arbitrary MA circle which is tangent to its boundary at more than one point. Then, we separate the domain into two or more subdomains with the MA circle. In this process, we reduced the number of nonmanifold vertices.

In our approach, we simplify the nonmanifold domain and multishell domain into manifold domain with holes, that are considered separately.

Let a compact domain with genus n be a compact domain with n holes, such that no hole contains any other hole of the domain. In our approach, we subdivide the domain so that the genus of the domain reduces. Consider the subdivision process in Figure 3.7. Notice that $\mathcal{C}_L - \{p, q\}$ and $\mathcal{C}_R - \{p, q\}$ are associated with the same MA circle as the boundary point p and q. So, $\mathcal{C}_L - \{p, q\}$ and $\mathcal{C}_R - \{p, q\}$ can be ignored if we know the MA is bounded by $\beta_L, \overline{pc}, \overline{cq}$ for α_L and is bounded by $\beta_R, \overline{qc}, \overline{cp}$ for α_R. So, to consider the MAT for α_L is the same as considering the MAT for β_L, and trimming the MA point by the line segments $\overline{pc}$ and $\overline{cq}$. The MA point for α_L can be defined as the closured of points which have more than one foot point on $\beta_L \cup \{p, q\}$, and the MA point is in the region bounded by β_L, $\overline{pc}$ and $\overline{cq}$. So, we define $\overline{pc}$ and $\overline{cq}$ as fake boundary for the domain bounded by β_L, $\overline{pc}$ and $\overline{cq}$. And we define $\beta_L \cup \{p, q\}$ is the true boundary for this domain. Under this modification, theorems 3.1, 3.3 and 3.14(2)(3)(4) are still true for the domain with proper fake boundaries produced by subdivision process.

Consider a genus 1 compact domain D, as shown in Figure 3.15(a). We can find two MA circles $\mathcal{C}_1$, $\mathcal{C}_2$, both of the MA circles have footpoints on the exterior boundary

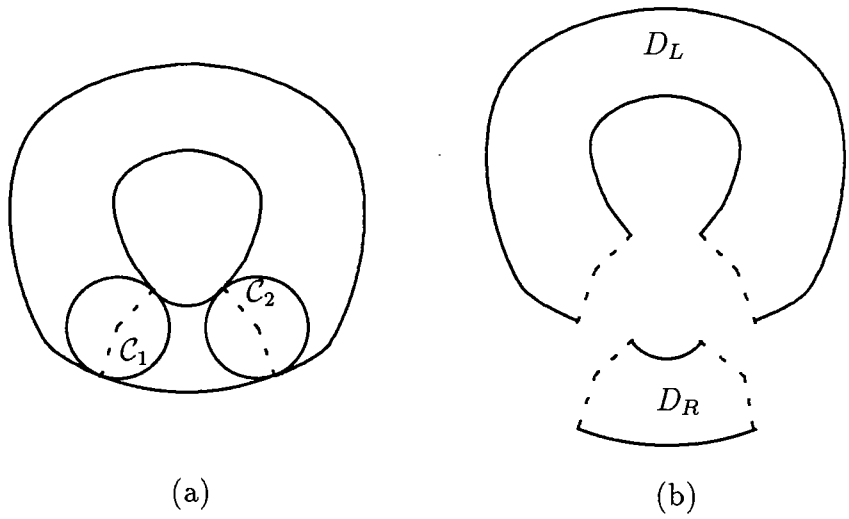


Figure 3.15 Reducing the genus of the domain

and the boundary of the hole. By subdividing the domain using the false boundary, two subdomains with genus zero are obtained, and we call them D_L and D_R. See also Figure 3.15. From theorem 3.3, the MAT of D_L and D_R are connected with tree structure. And, there are exactly two paths to connect $\mathcal{C}_1$ and $\mathcal{C}_2$, one via D_L, and the other via D_R. After combining D_L and D_R, we know that the MAT of D is connected and has only one loop. Notice that the MAT of D is unique because the MAT of D_L and D_R are unique.

Because fake boundaries have been introduced, even though MA circles $\mathcal{C}_1$ and $\mathcal{C}_2$ intersect each other, the subdivision process still works fine. Otherwise, D_L still has a loop after subdivision because D_L still has genus 1.

Consider a genus n compact domain. We can find two MA circles, where both of the circles have footpoints on the exterior boundary and the boundary of the same hole. By subdividing the domain using fake edges, we reduce the domain to two subdomains, one has genus zero and the other has genus $n-1$. Continuing the subdivision process and by induction, we have

Corollary 3.30 The genus n compact domain has a unique, connected MAT, and the MA has exactly n simple loops, each loop surrounding a hole in the domain and all loops are surrounded by the exterior boundary of the domain. Furthermore, no loop contains other loops.

4. APPLICATION - THE MAT OF 2D AND 3D SOLIDS

There are two sections in this chapter. The first section describes algorithms for extracting the MAT of 2D solids and the second section describes algorithms for extracting the MAT of 3D solids. The algorithms are based on the maximal disc criterion or on the equal distance criterion.

4.1 Proposed methods in 2D and their comparisons

There are many algorithms for finding the MAT of 2D solids. We discuss two algorithms, one due to Danielson, the other due to Rosenfeld and Pfaltz. Both algorithms are based on the maximal disc criterion. Algorithms based on thinning the domain are not discussed in this section because most of them do not use Euclidean distance. Three algorithms are proposed to find the MAT for 2D solids. They are the interpolation/extrapolation method, the Newton and march and the grid edge interpolation method. The first method is based on the maximal disc criterion and the others are base on the equal distance criterion.

The preprocessing of the first three algorithms is the same. The first step is to find the distance transform of the domain using Danielson's 8SED algorithm. After that, each pixel has distance amplitudes (a, b) which indicates that the distance to the boundary is $\sqrt{a^2 + b^2}$. It also means the maximal inscribed circle centered at the pixel has radius $\sqrt{a^2 + b^2}$. The comparison between amplitudes (a, b) and (a', b') is based on the distance to the boundary.

Each method has a different strategy to extract the MA grid points after the distance transformation has been computed. We will discuss them one by one in each subsection.

4.1.1 Rosenfeld and Pfaltz's Method

Rosenfeld and Pfaltz [66] use city block distance to extract the distance transform. After that, all of the local maxima of the distance transform are extracted as the MA of the picture. Assume $T = (t_{i,j})$ is the distance transform of an image. Test whether the pixel at (i, j) is an MA point as follows:

> If none of $t_{i-1,j}, t_{i,j-1}, t_{i+1,j}, t_{i,j+1}$ is equal to $t_{i,j} + 1$, then (i, j) is an MA point.

We use the Euclidean distance transform and extend this idea by checking the distance from 4 neighbors to 8 neighbors. That is,

> If none of $t_{i-1,j}, t_{i,j-1}, t_{i+1,j}, t_{i,j+1}$ is greater than or equal to $t_{i,j} + 1$, and none of $t_{i-1,j-1}, t_{i-1,j+1}, t_{i+1,j-1}, t_{i+1,j+1}$ is greater than or equal to $t_{i,j} + \sqrt{2}$, then (i, j) is an MA points.

The strategy used above is not good for the Euclidean distance transform because the $t_{i,j}$ value is the distance to the boundary and the associated boundary value could come from any direction, not only from vertical or horizontal directions. Consider Figure 4.1: if $a \neq b$, then the values for d_1, d_2, d_3 and d_4 are all less than 1.0 although the grid distance of the point to the vertical and horizontal neighbors is 1.0. An analogous result can be derived for the diagonal neighbors. See Figure 4.2 for a example using this strategy. In this Figure, all circled numbers are MA points.

Predictably, the MA point for the domain in Figure 2.4 will have a lot of noise, as shown in Figure 4.3(a). The picture has 500 $\times$ 500 grid points. Notice that there is less noise if the pixel has a footpoint that lies in the vertical, horizontal or diagonal directions.

Using Montanari's idea [46], we give a threshold value to identify an MA point. So, the new strategy becomes:

> If none of $t_{i-1,j}, t_{i,j-1}, t_{i+1,j}, t_{i+1,j+1}$ are greater than or equal to $t_{i,j} + 1.0 * threshold$, and none of $t_{i-1,j-1}, t_{i-1,j+1}, t_{i+1,j-1}, t_{i+1,j+1}$ is greater than or equal to $t_{i,j} + \sqrt{2} * threshold$, then (i, j) is an MA points.

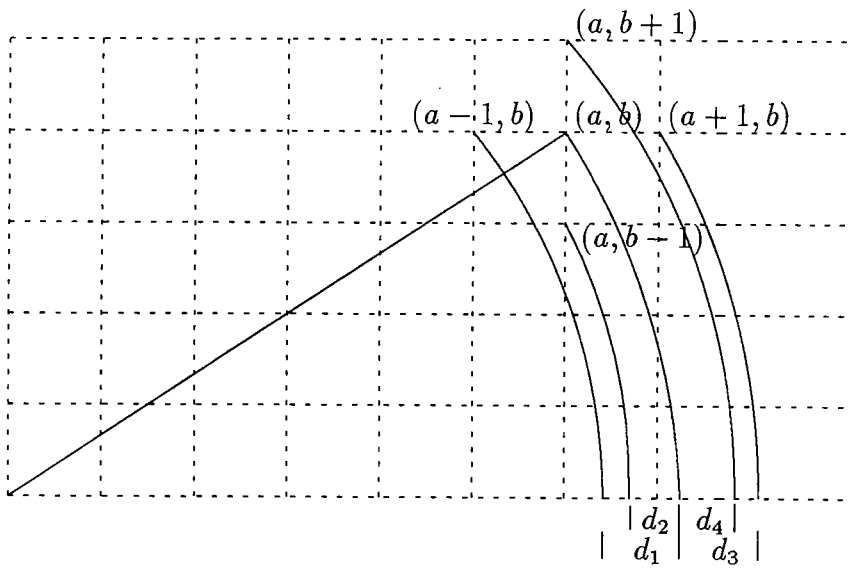


Figure 4.1 The radii of a circle and its neighbor circles

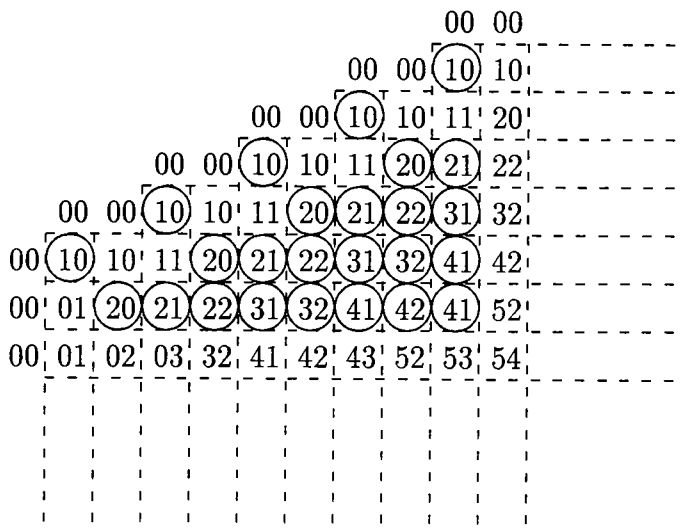
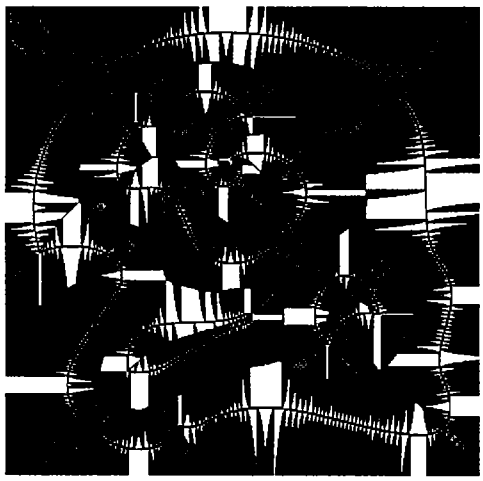
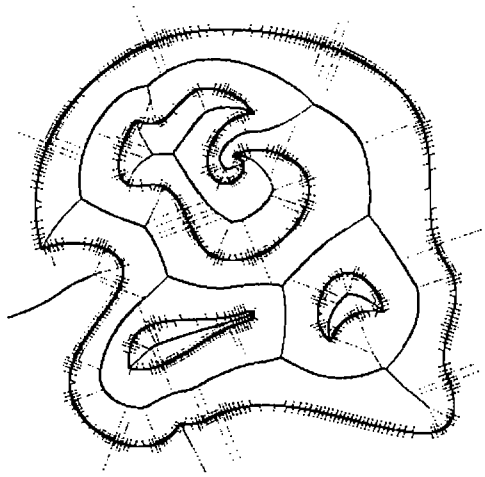


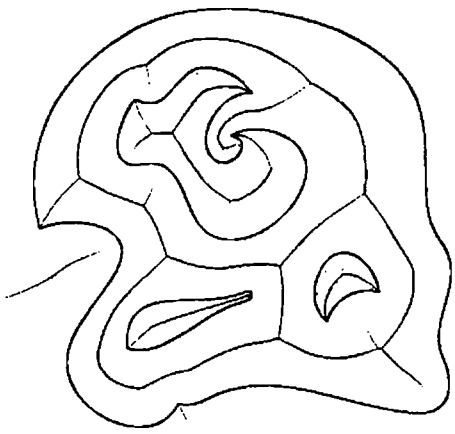
Figure 4.2 The MA points extracted by modifying Rosenfeld and Pfaltz's Method



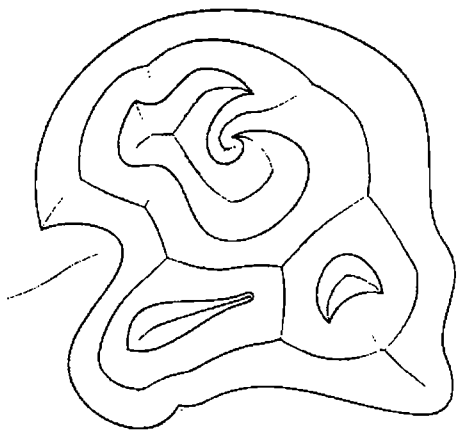
(a) no threshold value



(b) threshold = 0.9



(c) threshold = 0.8



(d) threshold = 0.7

Figure 4.3 The MA points found by using Rosenfeld and Pfaltz's method

The MA points for the domain in Figure 2.4 is shown in Figure 4.3(b),(c),(d) for threshold values equal to 0.9, 0.8 and 0.7 respectively. Notice that some MA points have been eliminated with small threshold value.

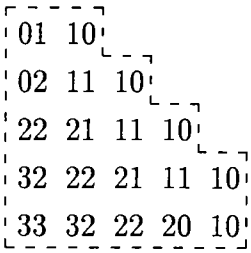
4.1.2 Danielson's Method

Danielson [14] defined different discrete circles for each distance amplitudes. Let the discrete circle $DCircle_{(i,j)}(a,b)$ be centered at a grid point whose coordinates are (i,j) with distance amplitudes (a,b). If the coordinates of the grid are not important, we write $DCircle(a,b)$ instead. Figure 4.4 shows as example, the quarter circles $DCircle(4,4), DCircle(5,0), DCircle(4,3)$ and $DCircle(3,3)$ for 8SED.

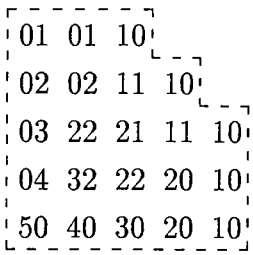
Given a discrete circle $DCircle_{(i,j)}(a,b)$, the table $g4(a,b)$ stores the squared radius of largest discrete circle that may be centered at any one of the four rectilinear neighbors of (i,j) and is contained in $DCircle_{(i,j)}(a,b)$. Similarly, $g8(a,b)$ stores the largest squared radius possible for a contained discrete circle centered at the diagonal neighbors of (i,j). For example, $g4(4,4) = (5,0) = (4,3) = 25$ and $g8(4,4) = (3,3) = 18$, as shown in Figure 4.4. Because of the error produced by the discretization, it is possible that two circles with different squared radius produce the same discrete circle.

Whether the grid point (i,j) with distance amplitudes (a,b) is an MA point is decided by its grid neighbors. If the discrete circle associated with the grid point (i,j) is not contained in one of the 8 discrete circles associated with its neighbors, then it is a MA point. That is,

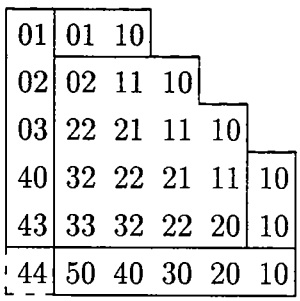
> If any one of $(a4,b4)$ or $(a8,b8)$, where $(a4,b4)$ and $(a8,b8)$ are the distance amplitudes of the vertical/horizontal and diagonal neighbors of (i,j) respectively, satisfies $g4(a4,b4) \geq (a,b)$ or $g8(a8,b8) \geq (a,b)$, then the grid point (i,j) is not an MA point because its associated discrete circle is contained in one of the discrete circles associated with its neighbors. Otherwise, it is an MA point.



(a) DCircle(3,3)



(b) DCircle(5,0)
DCircle(4,3)



(c) DCircle(4,4)

Figure 4.4 The Discrete quarter Circle with 8SED-mapping

The tables for $g4$ and $g8$ are not easy to produce because of the discretization. We derive a formula for the entries of the $g4$ and $g8$ tables. Although the values are the same for the examples shown in Danielson's [14] paper, it is not proved that for all grid point (i, j) with distance amplitude (a, b), the circle $DCircle_{(i,j)}(a, b)$ is contained in the circles $DCircle_{(i,j+1)}g4(a, b)$ and $DCircle_{(i+1,j+1)}g8(a, b)$. The formula we use is:

$$S(a,b) = \{(a',b') | b' = 0, 1, 2, \cdots \text{where } a' \text{ is maxima,}$$

$$\text{such that } a' \geq 0, b' \geq 0 \text{ and } (a',b') < (a,b)\}$$

$$S'(a,b) = \{(a'',b'') | b'' > 0 \text{ and } (a'',b''+1) \in S(a,b)\}$$

$$g4(a,b) = \min S(a,b)$$

$$g8(a,b) = \min S'(a,b)$$

The MA point using Danielson's method with the $g4$ and $g8$ table produced by the formula above for the domain of Figure 2.4 is shown in Figure 4.5(a).

If we consider the shape of the real circle instead of discrete circle, the equations produced for $g4$ and $g8$ will be different. Consider a circle centered at the origin with radius amplitudes (a, b), and let the difference of its radius with the radius of the circles centered at the origin with amplitudes $(a-1, b), (a, b-1), (a+1, b)$ and $(a, b+1)$ be $d1, d2, d3$ and $d4$, respectively, as shown in Figure 4.1. We define an error bound $e4$ as half of the minimum value of d_1, d_2, d_3 and d_4. Considering the eight neighbors, we can define $e8$ similarly. With these two values, $e4$ and $e8$, as an error bound for converting continuous circles to discrete circles, we produce $g4(a, b)$ and $g8(a, b)$ as follows:

$$r4 = \sqrt{a^2 + b^2} - 1 + e4$$

$$r8 = \sqrt{a^2 + b^2} - \sqrt{2} + e8$$

$$g4(a,b) = \max\{(a',b') | \text{ where } a' \geq 0 \text{ and } b' \geq 0 \text{ and } a'^2 + b'^2 \leq r4\}$$

$$g8(a,b) = \max\{(a',b') | \text{ where } a' \geq 0 \text{ and } b' \geq 0 \text{ and } a'^2 + b'^2 \leq r8\}$$

The MA points using Danielson's method with the $g4$ and $g8$ tables produced by the above formula for the domain in Figure 2.4 is shown in Figure 4.5(b). In this picture, some noise exists near the boundary of the domain.

We know the discretization for a curve has large relative errors nearby. When the grid point is far away from the boundary, the relative error produced by discretizing the domain boundary is reduced, as evidenced by Figure 4.5(b). With this idea, we produce $g4$ and $g8$ with different error bounds:

> If $a^2+b^2 > cut$, then use the strategy above to produce the $g4(a,b)$ and $g8(a,b)$ entry. Otherwise, use the strategy with a doubled error bound.

The MA points using this strategy with $cut = 20$ is shown in Figure 4.5(c).

Notice that the strategy used in Figure 4.5(a), call it the discrete circle strategy, produces less noise near the boundary whereas the strategy used in Figure 4.5(b), call it the continuous circle strategy, produces less noise for grid point far from the boundary. So, we have a hybrid strategy to produce the $g4$ and $g8$ table.

> If $a^2+b^2 \leq cut$, then produce the $g4(a,b)$ and $g8(a,b)$ entry using the discrete circle strategy. Otherwise, produce them using the continuous circle strategy.

A good value cut is determined by the experiment. However, we find that this approach produces new noise in the diagonal direction at distance cut. The MA points for $cut = 50$ are shown in Figure 4.5(d).

Now, let $g4c$ and $g8c$ be the tables produced by continuous circle strategy and $g4d$ and $g8d$ be the tables produced by discrete circle strategy. Let d be the squared distance from the grid (i,j) to the boundary of the image. We set two values low and $high$ where $low < high$, so that the discrete and continuous circle strategy will used for the grid whose squared distance to the boundary are $0 \leq d < low$ and $high < d < \infty$, respectively. For the case that $low \leq d \leq high$, both strategy will be used. The grid points (i,j) is an MA point only if the both strategies agree that (i,j) is an MA point. So, this "overlap" strategy is:

If $d < low$, then the $g4d$ and $g8d$ table are used to determine that whether (i, j) is an MA point or not.

If $d > high$, then the $g4c$ and $g8c$ table are used to determine that whether (i, j) is an MA point or not.

If $low \leq d \leq high$, the grid point (i, j) is an MA point only if both tables, $g4c$ and $g4d$ for vertical/horizontal neighbors and $g8c$ and $g8d$ for diagonal neighbors, are agree that the grid point (i, j) is an MA point.

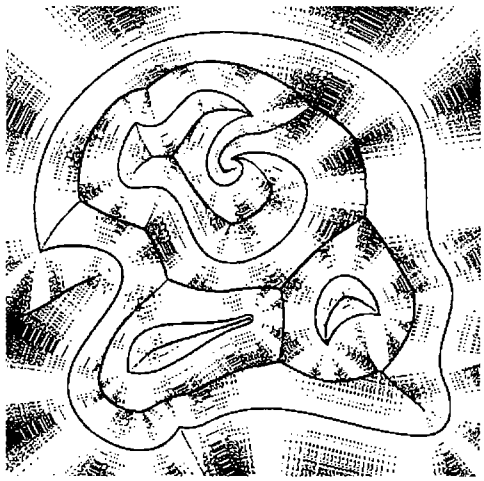
The results of this strategy with $low = 15$ and $high = 25$ are shown in Figure 4.5(e).

4.1.3 Interpolation/Extrapolation Method

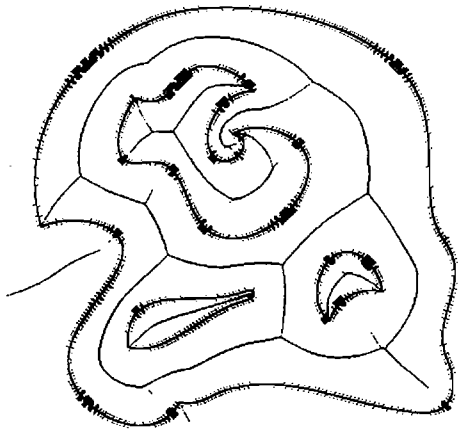
It is possible that non-MA points are not bounded by the maximal inscribed circle centered at any of its eight neighbors [33], as shown in Figure 4.6. In this Figure, the maximal inscribed circle centered at a is not bounded by the maximal inscribed circle centered at its neighbors. But, it is bounded by the maximal inscribed circle centered at i. In this situation, the maximal inscribed circle centered at a is not a maximal inscribed circle of the domain. In this algorithm, we want to find the circle centered at b, and the radius of this circle is interpolated by the nearby grid points, marked c and e. Let the interpolated radius be d_2. The distance from the boundary to the grids point a be d_1 and the length from a to the interpolated point b be l. We decide in this algorithm on MA points as follows:

If $d_2 - d_1 < l$, then the grid point is an MA point.

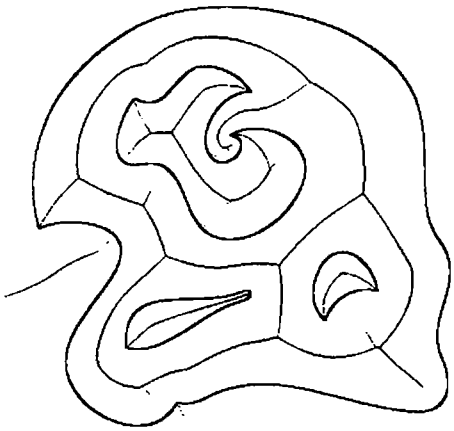
The idea of producing the equation $d_2 - d_1 < l$ comes from the cyclographic map. Before describing how we produce this equation, we would like to describe the relationship between points in 3D half space and a circle in 2D [62, 4]. Consider the point p in 3D half space and its associated circle $Circle(p)$ in 2D, as shown in Figure 4.7. Let us called the upper cone and lower cones with apex p, $UC(p)$ and $LC(p)$ respectively. A point q in 3D half space will have one of the following properties:



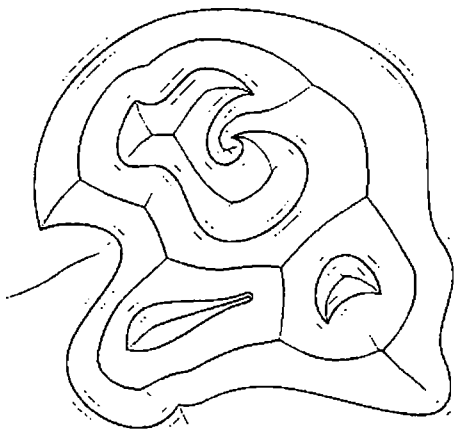
(a) discrete circle strategy



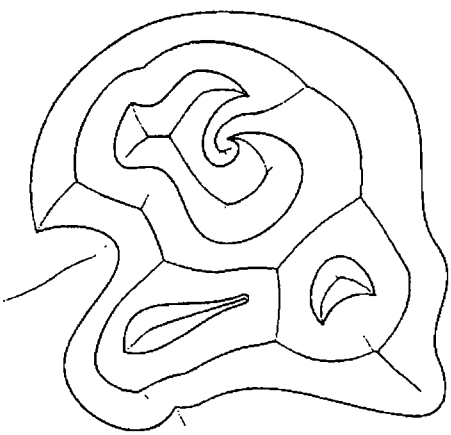
(b) continuous circle strategy



(c) continuous circle strategy with different error bounds



(d) hybrid strategy



(e) overlap strategy

Figure 4.5 The MA points obtained by different strategy

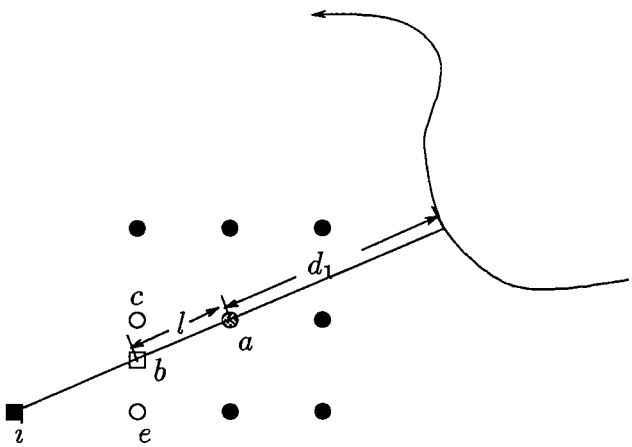


Figure 4.6 The maximal inscribed circle in the interpolation/extrapolation method

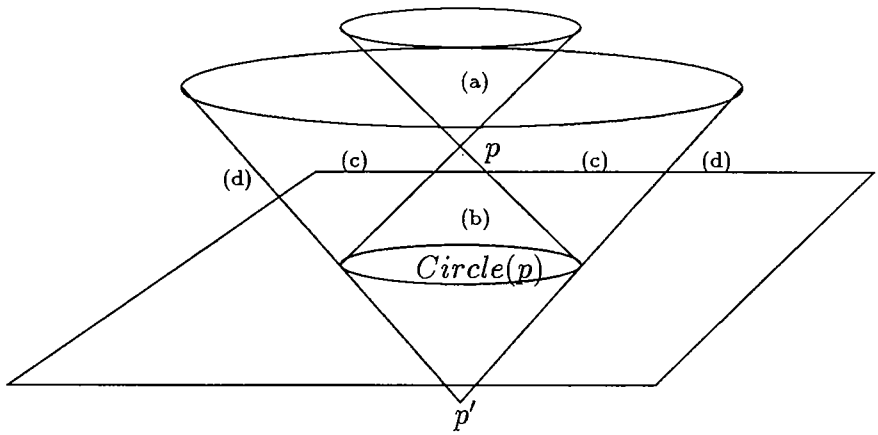


Figure 4.7 The relation between points in 3D half space and circles in 2D

1. If the point q is in the region (a) or on $UC(p)$, then $Circle(p) \subseteq Circle(q)$.
2. If the point q is in the region (b) or on $LC(p)$, then $Circle(q) \subseteq Circle(p)$.
3. If the point q is in the region (c), then $Circle(p)$ intersects $Circle(q)$ transversally.
4. If the point q is on $UC(p')$, then $Circle(p)$ tangent to $Circle(q)$ and neither contains the other.
5. If the point q is in the region (d), then $Circle(p)$ and $Circle(q)$ do not intersect.

So, if the point p is an MAT point of a domain, there are no other points q on the cyclographic map in the region (a) or on the boundary of the double cone whose apex is p. Otherwise, the $Circle(p)$ or $Circle(q)$ is not a maximal inscribed circle of the domain.

Now, consider Figure 4.8(a). We extend the picture into 3D so that the third dimension represents the distance to the boundary for each grid point, as shown in Figure 4.8(b). From this picture, if the grid point p is a MAT point, the θ angle, $cot^{-1}\frac{d_2-d_1}{l}$, should be greater than 45 degree. Otherwise, the $Circle(p) \subseteq Circle(q)$ and p is not an MAT point.

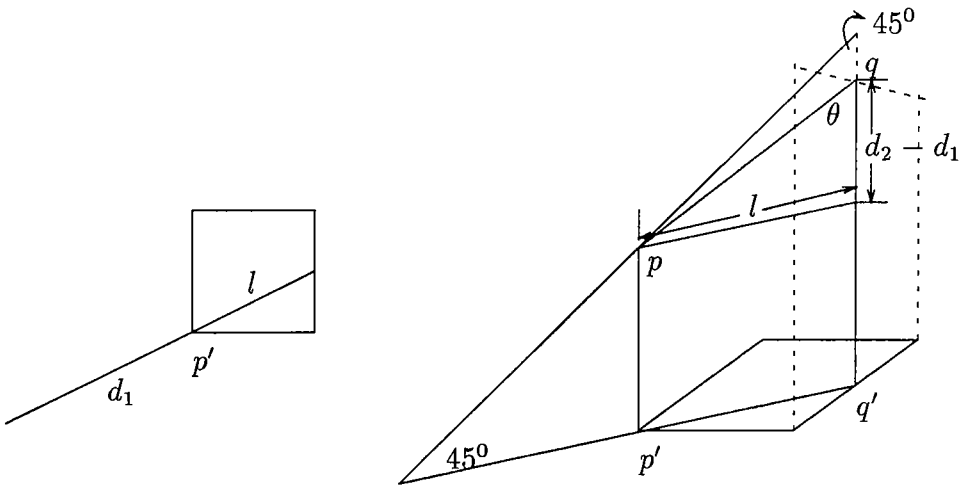


Figure 4.8 The interpolated radius d_2 and extrapolated radius $d_1 + l$

Notice that linear interpolation is used in Figure 4.8, and in our implementation. The output of this algorithm for the domain in Figure 2.4 is shown in Figure 4.9(a).

A threshold value is used to decrease the noisy MA points in the picture. The strategy we now use is:

If $d_2 - d_1 < l * threshhold$, then the grid point is an MA point.

Notice that the strategy we used before is $d_2 - d_1 < l * cot(\phi)$ where ϕ is 45^0. Now, we set $threshold = cot(\phi)$, so that a smaller threshold value has a large ϕ angle. Notice that if p is the grid point under consideration as in Figure 4.7, the larger the angle ϕ is, the larger the region (a) is, which then bounds more points and reduces more noise. But, it also deletes some MAT points, especially MA points whose footpoints lie in nearly the same direction.

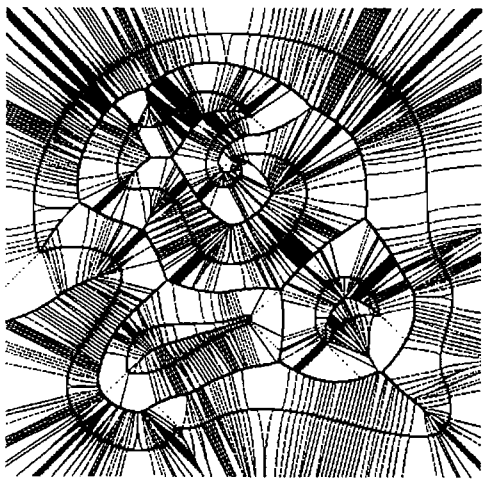
The MA points for the domain in Figure 2.4 using this approach with $threshold = 0.90, 0.80, 75$ are shown in Figure 4.9(b)(c)(d) respectively.

4.1.4 Newton and March

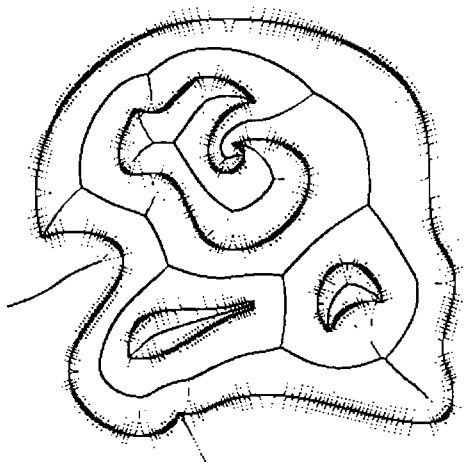
Form the theory we derived in Chapter 3, we know that a closed bounded domain has a connected MAT. If we can find an MA point for this domain, we can find all the MA points with a good strategy for walking along MA curves.

Given the boundary of a domain, we compute the distance transform for the discretized domain as preprocessing step. With every grid point we associate the index of a nearest edge or a concave vertex, and the direction and distance to that edge or concave vertex. The main purpose of this steps is to solve the proximity problem, and we will discuss it again later. We assume that the primitive curves from which the boundary of the domain has been constructed are smooth. The connection between two primitive curves can be a concave or a convex vertex. We call a primitive curve and a concave vertex a primitive object.

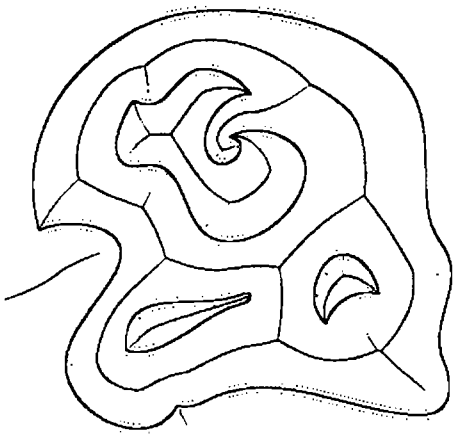
In this section, we describe:



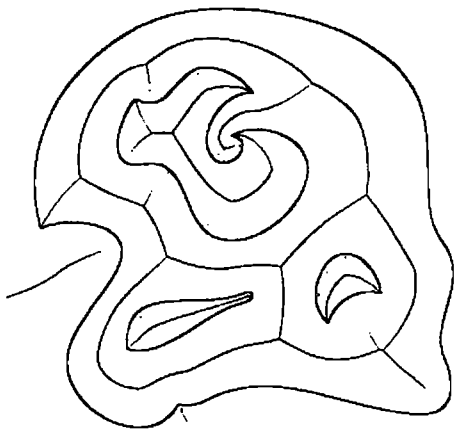
(a) no threshold value



(b) threshold = 0.9



(c) threshold = 0.8



(d) threshold = 0.75

Figure 4.9 The MA points found by using the interpolation/extrapolation method

1. How to formulate the system of equations for Newton iteration for one branch of the MA points.
2. How to determine whether there is an MA point in a grid square.
3. How to find a starting point.
4. How to find the next MA point.
5. How to find a branch point of the MA.
6. How to deal with the end points of the MA.

4.1.4.1 How to formulate the system of equations for Newton iteration

Assume (u_1, v_1), (u_2, v_2) are points on the curve $f(x, y) = 0$, $g(x, y) = 0$ respectively, and that the normal of f at (u_1, v_1) and the normal of g at (u_2, v_2) contains a point (x, y), so that the distance of (x, y) from (u_1, v_1) is equal to the distance from (u_2, v_2). Let f_x denote the partial differential of f with respect to the variable x. The points (u_1, v_1), (u_2, v_2) and (x, y) satisfy the following system of equations: [31]:

$$f(u_1, v_1) = 0$$

$$g(u_2, v_2) = 0$$

$$f_y(u_1, v_1) * (x - u_1) - f_x(u_1, v_1) * (y - v_1) = 0$$

$$g_y(u_2, v_2) * (x - u_2) - g_x(u_2, v_2) * (y - v_2) = 0$$

$$(x - u_1)^2 + (y - v_1)^2 - (x - u_2)^2 - (y - v_2)^2 = 0$$

After solving the system, ordinarily, (u_1, v_1) and (u_2, v_2) would be footpoints of the point (x, y), but under certain circumstances, they are not because there is another boundary point that is closer than (u_1, v_1) or (u_2, v_2) are, as shown in Figure 4.10. The solution of this system defines the equal distance points of f and g. The distance here is local distance, but the MA points are with respect to the global distance.

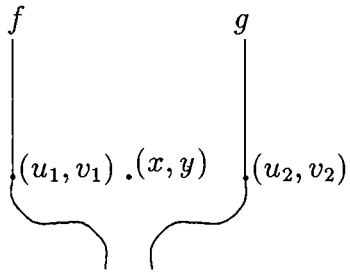


Figure 4.10 A solution point for newton iteration that is not an MA point

That is why the solution of (x, y) is a superset of the MA points of f and g. The difference between local distance and global distance will be introduced in chapter 5.

There are 5 equations with 6 variable. So, the solution of this system of equations has 1 degree of freedom in general. The first two equations state that points (u_1, v_1) and (u_2, v_2) should be on f and g, respectively. The third and fourth equations assure the perpendicularity properties when measuring the distance from the MA point to the curve. The lines through the points (x, y) and (u_1, v_1), and through the points (x, y) and (u_2, v_2) will be perpendicular to the curves f and g respectively. The last equation states that the distance of the MA point from f and from g are equal.

The system of equations for the equal distance points of a vertex (u_2, v_2) and a curve f is:

$$f(u_1, v_1) = 0$$

$$f_y(u_1, v_1) * (x - u_1) - f_x(u_1, v_1) * (y - v_1) = 0$$

$$(x - u_1)^2 + (y - v_1)^2 - (x - u_2)^2 - (y - v_2)^2 = 0$$

The equal distance curve for two points (u_1, v_1) and (u_2, v_2) is a line. The linear equation is obtained by simplification from equal distance equation below:

$$(x - u_1)^2 + (y - v_1)^2 = (x - u_2)^2 + (y - v_2)^2$$

Many curves have MA points, and we call them self-MA points. Let (u_1, v_1) and (u_2, v_2) be two points on f so that the normal of f at (u_1, v_1) intersects the normal

of f at (u_2, v_2) at (x, y). Furthermore, the distance between (u_1, v_1) and (x, y) is equal to the distance between (u_2, v_2). The system of equations for finding the equal distance points of the curve f is [35]:

$$f(u_1, v_1) = 0$$

$$f(u_2, v_2) = 0$$

$$f_y(u_1, v_1) * (x - u_1) - f_x(u_1, v_1) * (y - v_1) = 0$$

$$f_y(u_2, v_2) * (x - u_2) - f_x(u_2, v_2) * (y - v_2) = 0$$

$$(x - u_1)^2 + (y - v_1)^2 - (x - u_2)^2 - (y - v_2)^2 = 0$$

$$(1 - a * (u_1 - v_1)) * (1 - a * (u_2 - v_2)) = 0$$

There are 6 equations with 7 variables. Let us consider only the first five equations for the case $(u_1, v_1) = (u_2, v_2)$. When $(u_1, v_1) = (u_2, v_2)$, the first five equations reduce to 2 equations with 4 variables, namely the first and the third equation. Any point (x, y) on the normal of f at (u_1, v_1) satisfies these two equations. So, we add a sixth equation which asserts that the two points (u_1, v_1) and (u_2, v_2) are not equal. Notice that if points (u_1, v_1) and (u_2, v_2) are equal, then there is no a that satisfies the last equation.

4.1.4.2 How to know whether there is an MA point in a grid square

Every grid point has associated with it the direction and distance to the boundary of the domain. So, we can find the range of direction angles for the four vertices of the grid square. Intuitively, if this range is small, as shown in Figure 4.11(a), there is no MA point in this grid square. If the angle is large, as shown in Figure 4.11(b)(c), then there are likely MA points in the grid square. Because we can find two grid points that produce a large angle range, we can also find the associated primitive objects and formulate the system of equations for Newton iteration. The start point for the Newton iteration is easy to find.

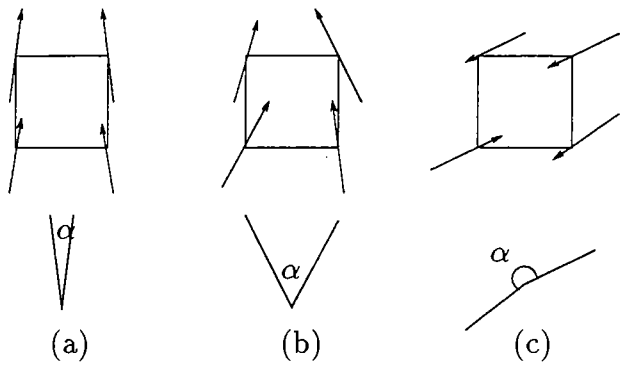


Figure 4.11 The angle range of the vector from the boundary to the grid points

This idea can be extended into 3D. A cone which contains all vectors of the 8 corner points of a grid cube is determined. Such a cone can be represented by a axis vector and an angle. When the angle is large, an MA point is expected in the grid cube. When it is small, no MA point is expected.

4.1.4.3 How to find a starting point

If the boundary of the domain has a convex vertex, this convex vertex is a start point for the MA of the two adjacent primitive curves that intersect at this point. If this domain has no convex vertex, we can find a grid point p, so that among all grid points, it has maximal distance to the boundary of the domain. Consider those four grid squares of which p is a corner. Then not all of the grid points in these four grid squares can have similar directions from the boundary. Otherwise, it contradicts the fact that p has maximal distance to the boundary. So, one of the neighbor grid square of p contains an MA point. After identifying the grid square with MA point in the interior, a start point is easy to find.

4.1.4.4 How to find the next MA point

After finding one MA point, we would like to find the next MA point so that the we can march along the MA branch. The new approximate point can be found by

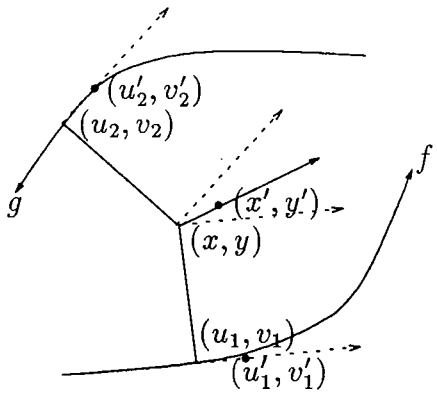


Figure 4.12 Next new estimated points for newton iteration

using the tangent line of the primitive curves. Consider the Figure 4.12; here f and g are two curves and we are marching. The points $(x, y), (u_1, v_1)$ and (u_2, v_2) are an MA point of f and g, a nearest point on f and a nearest point on g. Let the boundary of the curve be oriented counterclockwise. The tangent line of the point (u_1, v_1) can be subdivided into two half lines with (u_1, v_1) an end point. We define the half tangent line as positive tangent line if it has same orientation as the curve, otherwise, it is a negative tangent line. In Figure 4.12, f has a positive tangent line and g has a negative tangent line shown dashed.

The new estimated point $(x', y', u_1', v_1', u_2', v_2')$ for Newton iteration can be found easily. The points (u_1', v_1') and (u_2', v_2') can be obtained from the point (u_1, v_1) and (u_2, v_2), walking along the positive tangent line and negative line respectively, with distance *step*, as shown in Figure 4.12. The point (x', y') can be obtained from bisector of these two tangent line, which is always passing through the point (x, y), with the distance *step* to the point (x, y). From the new estimated point with Newton iteration, we can find the new candidate MA point and its associated footpoints.

4.1.4.5 How to find a branch point of the MA

With every step of the Newton iteration, we find a new candidate MA point, which has equal distance to two primitive objects. Whether this candidate MA point

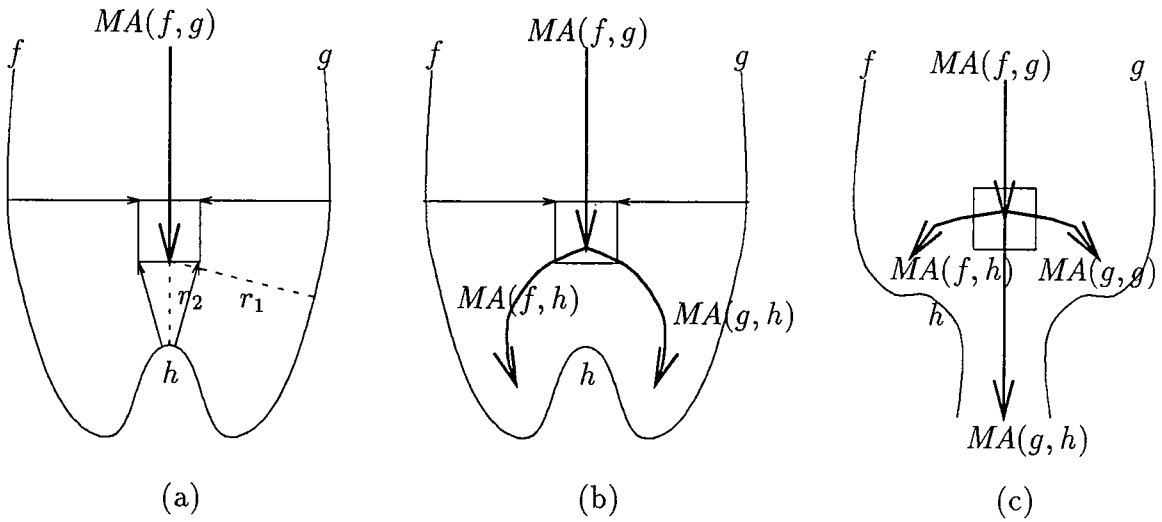
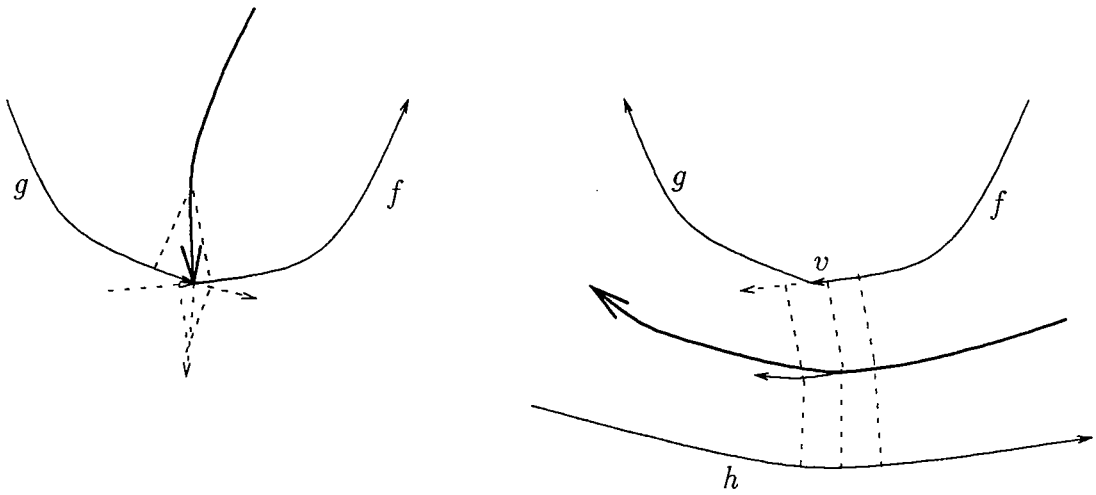


Figure 4.13 Crossing the branch of the MA

is in fact an MA point for the domain can be determined by the distance to the two primitive objects, call it r_1, and the distance to the boundary of the domain, call it r_2. If $r_1 > r_2$, then the candidate MA point is not an MA point of the domain, as shown in Figure 4.13(a). In this case, the MA curve $MA(f, g)$, whose footpoints are on f and g, has reached a branch point. Let the candidated MA point be in the grid square B. From the four corner grid points of B, we can find the associated new primitive object, so that the new branch of the MA can be traced. If the new primitive object is a primitive curve h, as shown in Figure 4.13(b), then the new MA branch will be $MA(f, h)$ and $MA(g, h)$. It is possible that two new primitive objects are found for the new MA branch, as shown Figure 4.13(c). In that case, we deal with one primitive curve at a time. For example, if we deal with h first, the new MA branches will be $MA(f, h)$ and $MA(g, h)$. After the first step extracting $MA(g, h)$, we will find that the new candidate MA point is not an MA point of the domain, so that two new MA branches can be found immediately, namely $MA(g, g)$ and (g, h).

Now, consider the case $r_1 < r_2$. In this case, the footpoint for one primitive curve is in the interior of the domain, as shown in Figure 4.14. Then, the system of



(a) (b)

Figure 4.14 The parameter of f or g is out of range

equations should be changed from the curve/curve case to the curve/vertex case, or, from the curve/vertex case to the vertex/vertex case.

4.1.4.6 How to deal with the end points of the MA

From the theory in chapter 3, an end point of the MA is always at a convex vertex, or at the maximal curvature center of a primitive curve. Moreover, we know that the end points of the MA are the closure of the MA points which have two or more footpoints. If the MA curve approaches an end point of the MA, then those two footpoints approach to each other. So, the maximal curvature center can be identified by two footpoints approaching each other, as shown in Figure 4.15. Notice that the end point approach to the convex vertex also has the property that its two footpoints approach to each other.

4.1.4.7 Notes on the implementation

We have implemented this algorithm assuming a boundary that is a piecewise Bézier curve. The system of equations is simpler because one parameter for a primitive

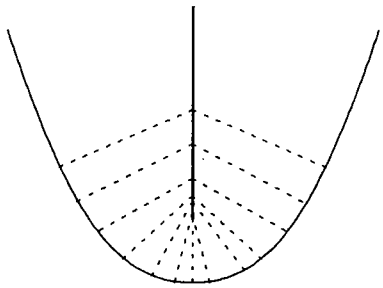


Figure 4.15 The approach from normal point to end point

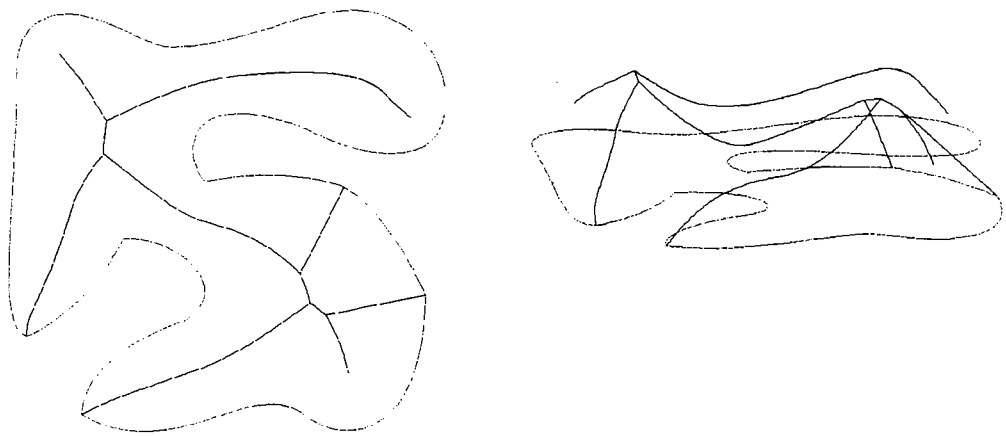
curve identifies a footpoint. Let $f(s) = (f_1(s), f_2(s))$ and $g(t) = (g_1(t), g_2(t))$ be two Bézier curves and let $f_{1,s}$ be the partial differential of f_1 with respect to the parameter s. Then the system of equation for these two curves is:

$$(f_{1,s}, f_{2,s}) \cdot (x - f_1(s), y - f_2(s)) = 0$$

$$(g_{1,t}, g_{2,t}) \cdot (x - g_1(t), y - g_2(t)) = 0$$

$$(x - f_1(s))^2 + (y - f_2(s))^2 = (x - g_1(t))^2 + (y - g_2(t))^2$$

There are 3 equations with four variables. The point $(f_1(s), f_2(s))$ and the vector $(f_{1,s}, f_{2,s})$ can be easily derived from the control points of the curve f [2]. Whether the footpoints of the candidate MA point are on the boundary of the domain is easy to decide by checking the value of the parameter. If the parameter value is between 0 and 1, then it is on the boundary, otherwise it is not. If both the parameters of f and g are out of range, then the MA curve went past a convex vertex, as shown in Figure 4.14(a). If only one of the parameter is out of range, as shown in Figure 4.14(b), a new system of equations for Newton iteration should be considered. One difficult decision in the algorithm is to choose a tolerance for this parameter. If we let $-\epsilon < s < 1.0 + \epsilon$ to mean that the point is on f, the true distance from $f(0)$ to $f(-\epsilon)$ depends on the length of the curve f between $f(0)$ to $f(1)$. So, if f is a very long curve and g is a very short one, it is not correct to use the same ϵ for the two



(a) (b)

Figure 4.16 The MA points found by using Newton and march

curves. For estimating the length of the curve, we can use the length of the control polygon of the curve. From it, a better ϵ for each curve can be determined.

One example domain with its MA for this algorithm is shown in Figure 4.16(a). Considering the radius of the maximal inscribed circle as the third dimension, we get the Figure 4.16(b).

4.1.5 Grid edge interpolation

The systems of equations in the previous section always has n variable with $n-1$ equations. If we can add one more equation, such as the equation for a grid line, then only finite number of solutions exist. Assume that the grid square is small enough so that every grid square has at most one branch MA point in the interior, and the MA points for two primitive objects intersect the boundary of all grid square in at most 2 points, each point on a different grid edge. Under those assumptions, instead of marching step by step, we can find the MA points square by square. Consider the figure 4.17(a)(b). After finding the MA points on the grid line, we can do linear

interpolation for the MA points in this grid square, and the neighbor grid square will be the next box we considered. It is possible that there are three neighbor grid boxes if the new candidate MA point is a grid point, as shown in Figure 4.17(b). In this case we need to check all these three boxes. Notice if there are no MA points on the grid edge, the Newton iteration will diverge.

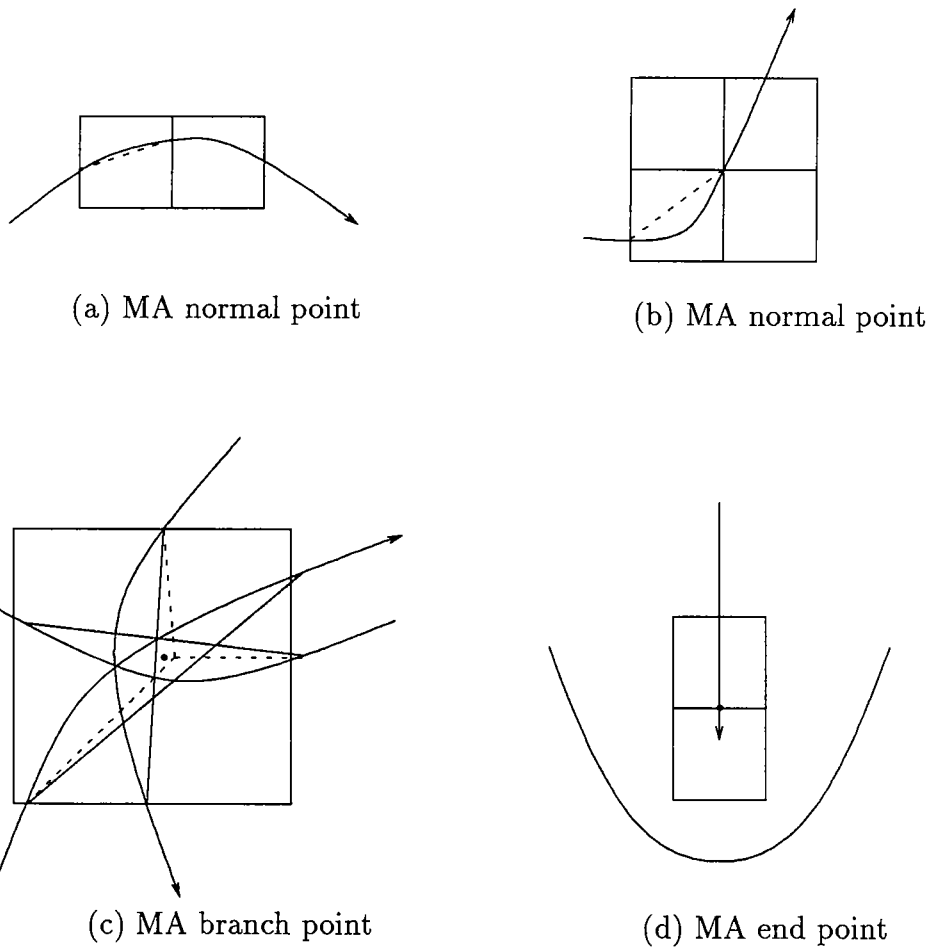
Assume that a branch point of the MA is in a box and connects three branches, call them $MA(f,g), MA(f,h)$ and $MA(g,h)$, as shown in Figure 4.17(c). Their associated approximated MA line is shown dashed. The MA branch point can be approximated by the center of the triangle formed by the intersection of these dashed lines. If the MA branch point connects more than three branches, we can find all of the intersection points from the approximated MA lines, and find the "center" of these points. If the points are located at (x_i, y_i), for $i = 1, n$, then their center can be defined as $\sum_{i=1}^{n}(x_i, y_i)/n$. After finding the approximate MA branch point, the connection between the MA branch points to each different branch is trivial.

If we consider the MA points of a parabola, the end point of the MA will be approximated by the last MA point intersection with a grid edge, as shown in Figure 4.17(d).

4.2 Proposed methods in 3D

We define some terms for 3D solids T before giving the algorithms. Let us consider an edge e of T. Assume this edge is the intersection of two CSG primitive surfaces f and g, whose normals point to the interior of T. Let the minimum angle and maximum angle for the normals of f and g at the points of e be θ_{min} and θ_{max}. If $\theta_{max} < \pi$, we call e a convex edge. If $\theta_{min} < \pi$ and $\theta_{max} \geq \pi$, we call e a singular edge. If $\theta_{min} \geq \pi$ and $\theta_{max} \geq \pi$, we call e a concave edge.

There are 3 different types of MA components of a 3D CSG object T, Namely an MA face, an MA edge and an MA vertex. The normal MA points comprise MA faces. The MA face, MA edge and MA vertex are locally homeomorphic to the plane, line,



(a) MA normal point

(b) MA normal point

(c) MA branch point

(d) MA end point

Figure 4.17 The MA points found by Grid edge interpolation

and point respectively. Because the dimensionality of these MA, we call them 2-MA, 1-MA and 0-MA for MA face, MA edge and MA vertex respectively.

Notice that any combination of these three MA components is possible. Consider the radius function that is approximately a small constant function, for the MA shown in Figure 4.18. Then the original CSG objects shown in the Figure are bounded by:

(a) a solid sphere.

(b) a solid torus.

(c) the difference of two solid spheres with same center and different radius.

(d) the union of a solid cylinder with two solid spheres.

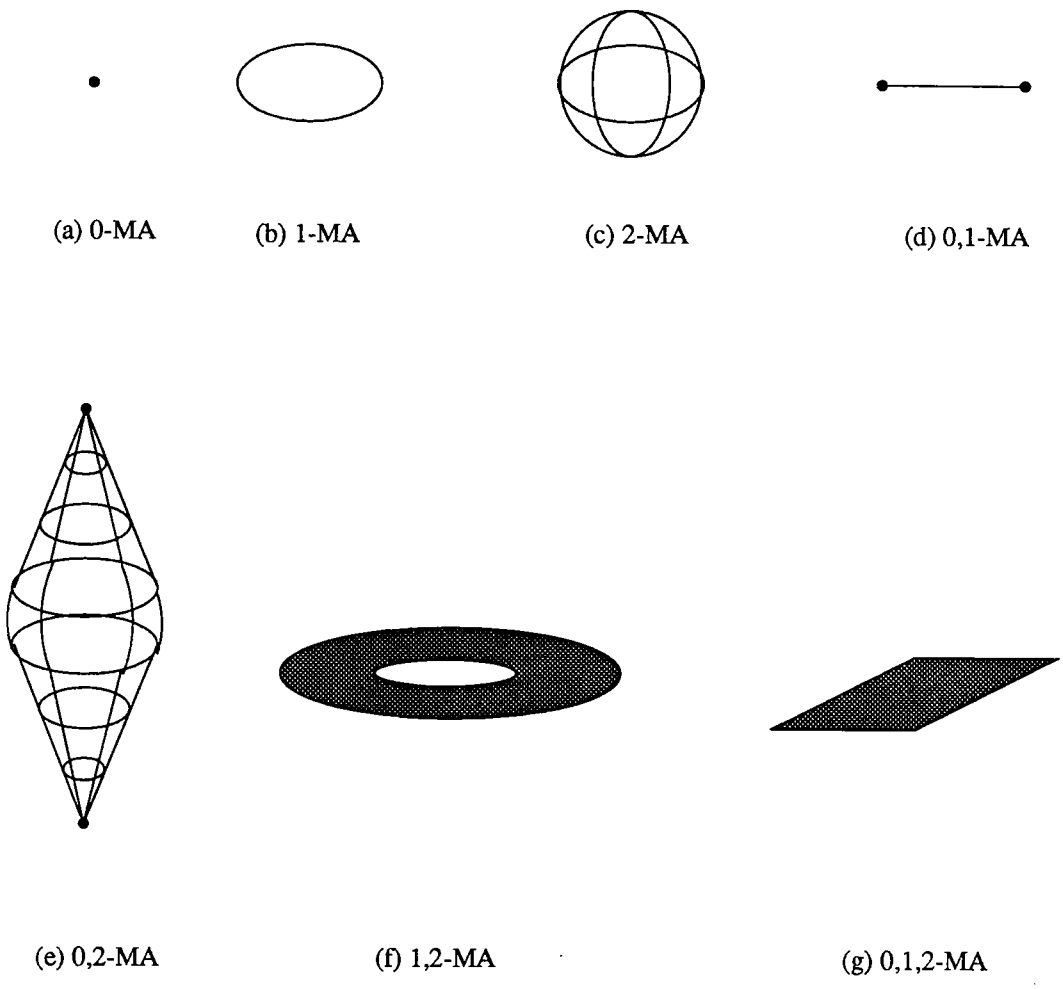
(e) proper difference, union operation on solid cone and solid torus, as shown in Figure 4.19. Because the symmetry of the MA in Figure 4.18(e), we only draw the domain which intersects a plane passing through the axes of cones and tori. In this figure, the vertices which are marked "a" should be on the center of those two circles in which the plane intersects the torus so that the MA curves associated with the vertices "a" are smooth.

(f) proper difference, union operation on solid cylinder and solid torus, as shown in Figure 4.20. As in (e), we only draw the domain which intersects a plane passing through the axes of the cylinders and tori.

(g) proper union operation on solid block, solid sphere and solid cylinder.

Although we believe the MA for T is connected, the 1-MA alone is not connected. Figure 4.18(f) is an example that the 1-MA alone is not connected. The MA for a long cylinder, as shown in Figure 4.23 is an example that the 2-MA alone is not connected.

We use a grid in our algorithm. In the very beginning, a bounding cube of T is found and every side of the cube is subdivided into 2^n line segments, where n is a natural number. This induces $(2^n)^3$ grid cubes.



(a) 0-MA (b) 1-MA (c) 2-MA (d) 0,1-MA

(e) 0,2-MA (f) 1,2-MA (g) 0,1,2-MA

Figure 4.18 The combination of MA

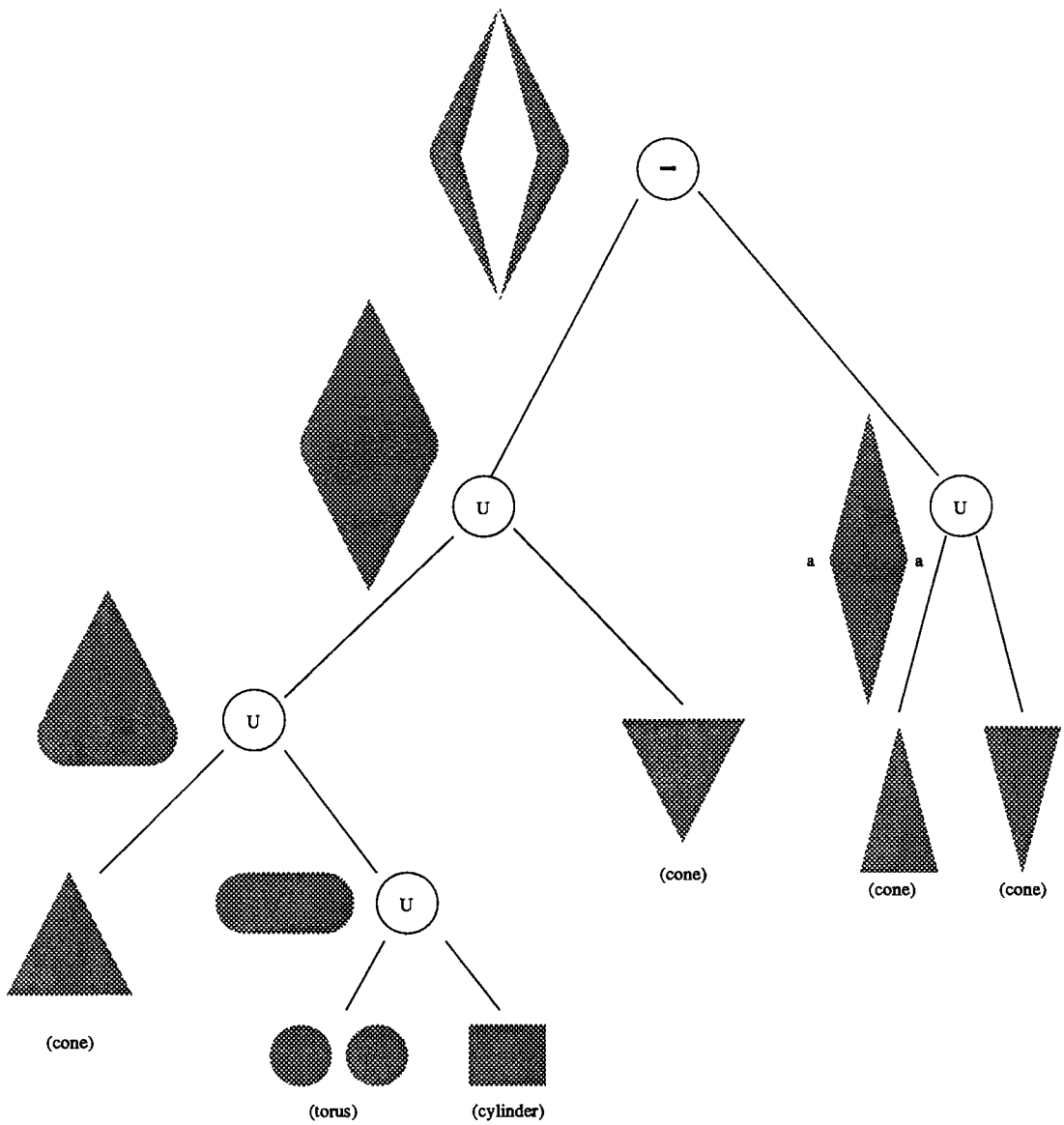


Figure 4.19 The domain for 0,2-MA

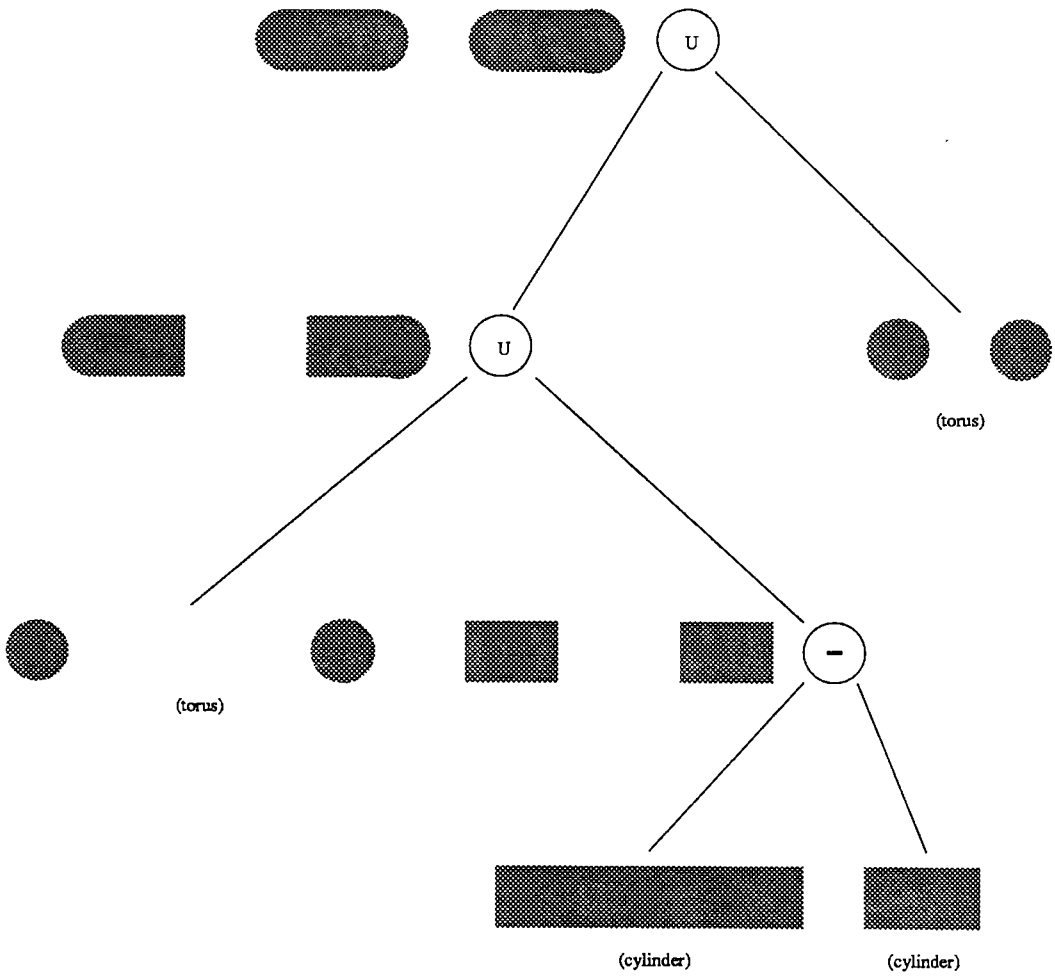


Figure 4.20 The domain for 1,2-MA

Our algorithms restricts T as follows:

1. There are no singular edges on T.

2. There is at most one 0-MA in each grid cube.

To simplify the data structure of the algorithms, we consider that every grid cube has three faces, call them $Xface, Yface$, and $Zface$, depending on its normal, three edges, call them $Xedge, Yedge$ and $Zedge$, depending on its direction, one vertex v, and the interior i, as shown in Figure 4.21. We call these fields position indices of the grid cube. We also call the vertex v a grid point.

The self-MAT is the MAT established from the primitive elements. For example, the self-MAT for a sphere, cylinder, cone and torus are a point, a line, a half line and a circle respectively. The self-MAT for a circle in space is a line. There is no self-MAT for a concave vertex.

Let $cube(i,j,k)\_Xface$ denotes the $Xface$ for the grid cube indexed by (i,j,k), then the 6 faces, 8 edges and 12 vertices that bound the interior region of a grid cube can be founded easily. For example, those 6 faces which bound $cube(i,j,k)\_i$ are $cube(i,j,k)\_Xface$, $cube(i+1,j,k)\_Xface$, $cube(i,j,k)\_Yface$, $cube(i,j+1,k)\_Yface$, $cube(i,j,k)\_Zface$, and $cube(i,j,k+1)\_Zface$.

Let p be point. We define the neighbor grid cube of p to be the grid cube whose closure contains p. So, if p is in the vertex, edge, face, or interior of a grid cube, there are 8,4,2,1 neighbor grid cubes for p respectively.

4.2.1 Rosenfeld and Pfaltz's Method, Danielson's Method and Interpolation/Extrapolation Method

The algorithms that extend Rosenfeld and Pfaltz's method, Danielson's method and the interpolation/extrapolation method into 3D are straightforward. We have implemented the interpolation/extrapolation method. With the experience in 2D, we can find a good threshold value to extract the MAT for 3D object. Because there is no geometry information for the boundary of the 3D solids in the purely discrete

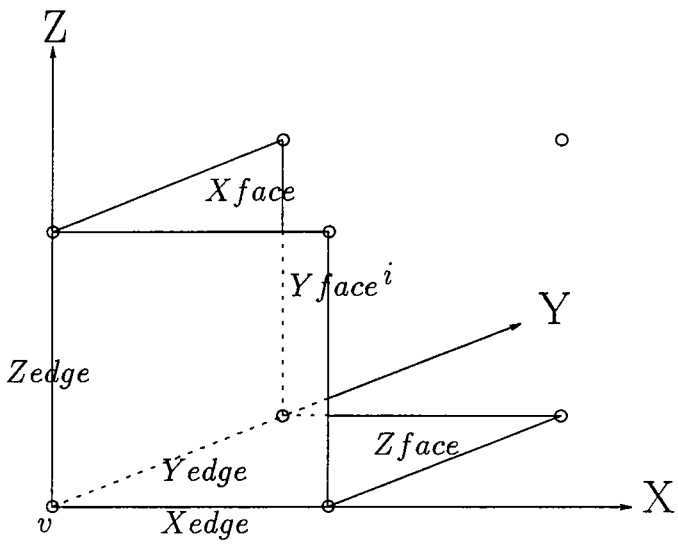


Figure 4.21 The data for each grid cube

algorithms, the connectivity of the MA is unknown. That is why the output of this algorithm is a set of grid points, and no shadow can be given in the output picture.

4.2.2 Finding nearest approach pairs on the primitive surfaces and marching section by section

This algorithm uses the following approach [32]:

1. By considering all pairs of boundary elements, determine for each pair the points that are equidistant from both elements and have minimum distance.
2. Sort those points by their distance from the boundary.
3. Processing the points by increasing distance, construct the MA by tracing the arising edges and faces.

To simplify the book keeping, we establish a table called primitive element table (PET). The primitive elements of T are the primitive surfaces, concave edges and concave vertices of T. And, PET stores the information of all primitive surfaces, concave edges, and concave vertices of T.

The edges and faces of T are subsets of space curves and surfaces, respectively. We call the corresponding curve or surface the carrier of the boundary element. Consider two primitive elements of T, f and g. The closest approach pairs of f and g are two points $p \in f$ and $q \in g$ so that the distance between p and q is locally minimal. The midpoint of p and q is called the nearest candidate MAT point of T. We further define the MAT point of f and g a candidate MAT point of T. Notice that the candidate MAT point or the nearest candidate MAT of T is not necessarily an MAT point of T. If the nearest candidate MAT point of T is an actual MAT point of T, we call the point a nearest MAT point of T.

The information stored in the PET are:

1. If the element is a primitive surface:
   (a) Exact representation of its carrier. The geometric information is stored. For example, a center point with radius represents a sphere.

(b) The implicit equation for the carrier of the primitive surface. The normal always points to the interior of T. For example, if T is the difference of two spheres centered at the origin with radius 4 and 1 respectively, the representation for these two spheres are $-x^2 - y^2 - z^2 + 4 = 0$ and $x^2 + y^2 + z^2 - 1 = 0$.

(c) The continuous faces which approximate the surface. These faces are produced from some solid modeler and they are only an approximation of the surface.

2. If the element is a concave edge:

   (a) Exact representation of its carrier. The only thing we need to store are the index of two CSG primitive surfaces which indicate the edge is the intersection of these two surfaces.

   (b) The continuous line segments which approximate the edge. These line segments are produced by some solid modeler.

3. If the element is a concave vertex, then we store the coordinates of the vertex.

The MAT entry (ME) stores some information for a specified MAT point:

1. The MA point coordinates.

2. The MA point position. That is, the index of a grid cube with proper position index.

3. The distance to the boundary of T.

4. A linked list that stores the coordinates of all its footpoints and the associated primitive elements.

The local MAT entry list (LMEL) stores the MAT entry in the same grid cube. The order of this list does not matter. The global MAT entry list (GMEL) stores

all MAT entries whose associated MAT in its associated grid cube have not yet been calculated. The associated distance to the boundary of T is increasing in this list.

Each grid cube stores:

1. The MA type in the grid cube. If it is marked 2-MA, this means that a 2-MA element has been found in the current grid cube. If it is marked 1-MA, this means that there are 1-MA and/or 2-MA elements in this grid cube currently. The 0-MA mark is defined similarly. Null-MA is used for a grid cube for which no MA point is known.

2. Geometric shape of the MA in this grid cube. This is approximated by a triangular face and it is used for computing an output sketch.

3. A primitive elements pairs list (PEPL). Each element of this list stores the index of two primitive elements, which means that the MAT for these two primitive elements in the grid cube has been calculated before.

4. Local MAT entry list.

The grid stack (GS) is a stack that stores the grid cubes we currently consider for finding MA faces. Every entry of this stack contains a grid index, the index of 2 primitive elements and a starting point for finding the MA face for these two primitive elements in the specified grid cube.

The input of our algorithm is the CSG tree for T and the output will be the MAT for T. The main structure of this algorithm comes from [32].

Algorithm:

1. Initialize the PET, LMEL, and GMEL.

2. Find the nearest MAT point of T and modify LMEL and GMEL.

3. If GMEL is empty, stop and go to step 8. Otherwise, pop an entry from GMEL, and call it D. Assume the MAT entry D stores the primitive elements f and g

and its neighbor grid cubes are $C_1, C_2, \cdots, C_n$. If the MA face for f and g is in grid cube C_i, where $i = 1, n$, and has not been found yet, which can be checked from PEPL in grid cube C_i, push C_i into GS.

4. If GS is empty, go to step 3. Otherwise, pop an entry from GS, and call it G. Do newton iteration in the grid cube G, based on the information in D, to find a new MA face. [10].

5. Modify every field of the grid cube G by:

   (a) change to its proper MA type.

   (b) store the shape of the MA element in this grid cube.

   (c) insert the element pairs f and g into PEPL.

   (d) Delete some ME from LMEL and GMEL if the candidate MAT point in that ME is not an MAT point of T.

6. For each new candidate MAT point on the boundary of the closure of the grid cube G, establish a new MAT entry and insert the entries into the LMEL of the correct grid cube. Also, insert these entries into GMEL.

7. Go to step 3

8. Connect all self-MAT of the primitive surface of T with the MAT we found.

For finding the nearest MA point of T in step 2, we need to consider the nearest MA point for every two primitive elements. The method we use to find the nearest MA point for two primitive elements of T is in appendix A.

Step 5 is best to explained using examples. In our implementation, triangular faces are used to approximate each MA face. To simplify the explanation, we use a plane to represent the MA face. Consider the Figure 4.22, and assume the plane whose corners are d_5, d_6, d_7, d_8 is the MA face we found before and the plane whose corners are d_1, d_2, d_3, d_4 is the MA face we find now for the primitive boundary elements f

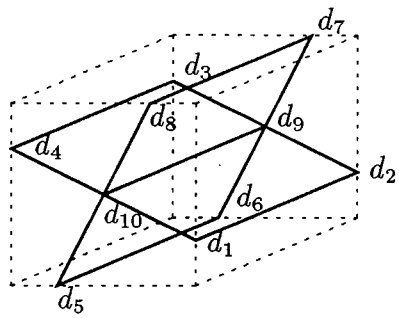


Figure 4.22 The construction for 1-skeleton

and g. We also assume that the starting point we use for finding the new MA face is d_1. Let the corresponding distance of the point d_i from the boundary of T be r_i for $i = 1, 8$. If $r_1 < r_4$ and $r_5 < r_8$, we know the resulting MA faces are in the plane with corners d_1, d_2, d_9, d_{10} union the plane corners d_5, d_6, d_9, d_{10}. So, we change the MA type from 2-MA to 1-MA, and delete ME which corresponds to the candidate MA point d_7 and d_8. Now, we have new candidate MA points d_2, d_9, d_{10}, which will be used in step 6. The change of 1-MA to 0-MA is similar.

Notice that from steps 1 to 7 of this algorithm, we only consider the MAT between two primitive elements of T, so the resulting MAT will always have type 2-MA elements. We need to consider the self-MAT too. For example, consider the MAT for a long cylinder, then the primitive elements for it will be an unbounded cylinder f and two planes Π_1 and Π_2, as shown in Figure 4.23. All MAT points of Π_1 and Π_2 are not MAT points of T. From step 1 to step 7 of this algorithm, we find the 2-MA for f and Π_1, and f and Π_2. We connect these two pieces in step 8.

This algorithm does not work when T has self-MA elements for concave edges. For example, consider the domain bounded by the union of a sphere and a cylinder, the self-MA point for the intersection curve of the sphere and cylinder is not extracted by our algorithm.

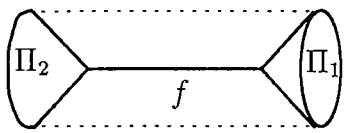


Figure 4.23 The MAT for a cylinder

4.2.3 Newton and March, and Grid edge interpolation for each domain box

The Newton and march method and the grid edge interpolation method have the same difficulties when extending from 2D to 3D. The geometry of the two boundary points which share the same MA points is more complex. In 2D, we have only 3 cases, namely vertex/vertex, vertex/curve and curve/curve. And, when marching on the MA points, there are only 6 possible ways to change the system of equations we are using. They are from vertex/vertex to vertex/curve, vertex/vertex to curve/curve, vertex/curve to vertex/vertex, vertex/curve to curve/curve, curve/curve to vertex/vertex and curve/curve to vertex/curve. In 3D, the geometry of the two boundary points which share the same MA points has 6 cases, and there are 30 possible ways to change the system of equations. Newton and march is more difficult because we must explore surfaces and the resulting MA points have to be connected properly. In the grid edge interpolation, we can interpolate the MA points we find in one grid box, which makes the problem simpler. We only list the algorithm for grid edge interpolation.

We would like to describe the data structure for this algorithm first. The Newton entry (NE), Newton table (NT), same primitives Newton list (SPNL), same grid Newton list (SGNL) and primitive element table (PET) are used. We will describe the detail information for them. Assume we want to find the MAT for the 3D CSG object T.

The Newton entry store:

1. Index pairs of two primitive elements. These two indices can be identical. In that case, we are dealing with a self-MAT element.

2. Starting point for Newton iteration for the two primitive elements.

3. Index to a grid cube. We will do Newton iteration based on the above information with the edge of these grid cubes. The MAT we find is in this grid.

4. Three pointers to Newton entries. One links the local Newton list and others link the global Newton list. So, the global Newton list is a doubly linked list.

The information stored in the PET is:

1. If the element is a primitive surface, we store the implicit equation of the carrier of the primitive surface.

2. If the element is a concave edge, we store the exact representation of its carrier. Here we store the index of two CSG primitive surface indicating that the edge is the intersection of these two surfaces.

3. If the element is a concave vertex, we store the coordinates of the vertex.

Notice that given a Newton entry, we can find the associated primitive elements we are interested in. And from the PET, we can derive the system of equations for Newton iteration. The starting point of the Newton iteration can be found in the Newton entry.

SPNL stores a doubly linked list and all entries in SPNL index the same primitives element pair. SGNL stores a link list and all of the entry in SGNL index the same grid cube.

The Newton table has dimension $n \times n$ where n is the number of primitive elements of T. Every entry index by f and g in the Newton table stores a pointer to the SPNL where every Newton entry in the SPNL has indexed the same primitive element pair.

Each grid cube stores:

1. The MA type in the grid cube. If it is 2-MA, it means that one 2-MA element has been found so far. If it marks 1-MA, it means that there is a 1-MA or several 2-MA in this grid cube currently. The 0-MA case is similar. Null-MA is used for grid cubes which have no MA point, so far.

2. Geometric shape of the MA in this grid cube. This is approximated by triangular faces and it is for the output sketch.

3. A primitive elements pairs list (PEPL). Each element of this list stores the index of two primitive elements, which means the MAT for the two primitive elements in this grid cube has been calculated before.

4. In/on/out classification and distance to the boundary of T from $cube(i,j,k)_v$. The footpoint or footpoints list is also stored.

There are some restrictions on T in the following algorithm:

1. There is at most one 0-MA element in each grid cube.

2. T has at least two primitive surfaces.

Algorithm:

1. Initialize the PET and NT.

2. Find a grid point which is inside of T and has maximal distance to the boundary of T. Find its neighbor grid point which will produce the MA elements on this grid box. Decide which two primitive elements we are dealing with. Produce the proper Newton entry and store it in the Newton table.

3. If one of the entries in the Newton table is not empty, then find the associated primitive elements, call them f and g. Otherwise, stop and exit.

4. If NT(f,g) is not empty, then pop an entry E from NT(f,g). Otherwise, go to step 3.

5. We know the two primitive elements in E are f and g. Assume the index of the cube in E is (i,j,k) and the starting point for Newton iteration in E is p. Let the MA point associated with p be w and its footpoint be r' on f and r'' on g. If $f \neq g$ or f and g are a concave curve which is not a circle, then do Newton

iteration from E. Otherwise, find the self-MAT by the geometric properties of f. Push (f, g) into PEPL of the $cube(i, j, k)$.

6. For each new MA point q which we found in step 5, find the distance from q to the boundary of T and its footpoints list and associated primitive element indices list. Because of numerical error, this list could be a singleton even when we have an MA point of T.

7. For each point q' in the footpoints list and its associated primitive element f'. If $f' \neq f$ and $f' \neq g$, or, $f = f'$ and the angle between $\vec{r'r}$ and $\vec{q'q}$ is large, or, $g = f'$ and the angle between $\vec{r''r}$ and $\vec{q'q}$ is large, then establish two new NE for function pairs (f', f) and (f', g) on the neighbor cubes of q. Push them into the proper entries of NT and link them into SPNL and SGNL. Otherwise, establish NE from the new MAT point on those neighbor cube where f and g pairs are not in the PEPL, link them into SPNL and SGNL, and go to step 5.

5. OTHER APPLICATION – OFFSETS FOR 2D CURVE

There are 5 sections in this chapter. The first section introduces the notions of local distance, global distance, local offset and global offset. In the second section, we review approaches to computing offsets devised by computer-aided geometric design (CAGD) and explain their local nature. In the third section, we reorganize the Euclidean distance transform, thereby deriving two algorithms that are better suited to computing the global offset. Both algorithms are further restructured in the fourth section adding interpolation and iteration, thus increasing the accuracy of the first two algorithms without changing the mesh size. The last section discusses the implementation.

5.1 Introduction of local distance, global distance, local offset and global offset

Given a plane curve C, its *offset* by a distance d is a curve $Off(C, d)$ such that the points of $Off(C, d)$ are at distance d from C. Similarly, given a surface S in 3-space, its offset by a distance d is a surface $Off(S, d)$ such that the points of the offset surface are at distance d from S. These informal definitions can be made more precise in one of two ways, depending on whether the distance is measured locally or globally.

Denote the Euclidean distance of two points p and q with $d(p, q)$. Let C be a curve in $\mathcal{R}^2$, and p any point. Define the *global distance* of p to C by

$$dist_g(p, C) = \inf\{d(p, q) \mid q \in C\}$$

where $d(p, q)$ is the Euclidean distance of two points. For a smooth curve C, a *local distance* can be defined as

$$dist_l(p, C) = d(p, q)$$

where q is on C and the line $\overline{pq}$ is perpendicular to the tangent of C at q. For most reasonable curves C these definitions make sense. The offsets for surfaces are defined analogously.

In the following, we assume that the curve C is simple and the surfaces to be smooth except at finitely many points or curves, and that any straight line intersects the surface in finitely many points, in the same sense. We call the curve or surface to be offset the *base curve* or *base surface*, respectively.

The local and global offsets can be defined as follows. In the global offset, the global distance is used, so that

$$Off_g(C,d) = \{p \in \mathcal{R}^2 \mid dist_g(p,C) = d\}$$

$$Off_g(S,d) = \{p \in \mathcal{R}^3 \mid dist_g(p,S) = d\}$$

Local offsets are simple to define analytically, given a parametric or implicit representation of the curve or surface[21, 29, 37]. Moreover, the local offset of an algebraic curve or surface again is an algebraic curve or surface.

It is convenient to think of geometric optics: Consider the surface of a lake, and image a curve C is drawn on the surface of the lake. If we hit the surface of the lake on the points of a curve C simultaneously, then, at time t, the peak of water wave generates the $d = tc$, local offsets, where c is the speed with which the front propagates. It is assumed that the front propagates with uniform speed and locally normal to the front. A physical analogue of the global offset is a grass fire front[3]. Briefly, image a curve C is drawn on a prairie, the fire begins along C simultaneously , and spreads burning uniformly. At time t, the fire front is the global offset at distance tc, where c is the speed with which the front propagates.

Notice that the water wave will "flow-through" after the self-intersection of the offset curve, but the grass fire will not. Figure 5.1 illustrates the local offsets of a ellipse and Figure 5.2 illustrates the global offsets of the same ellipse.

Note that the offset has two real components, but only one of these is shown in the figure.

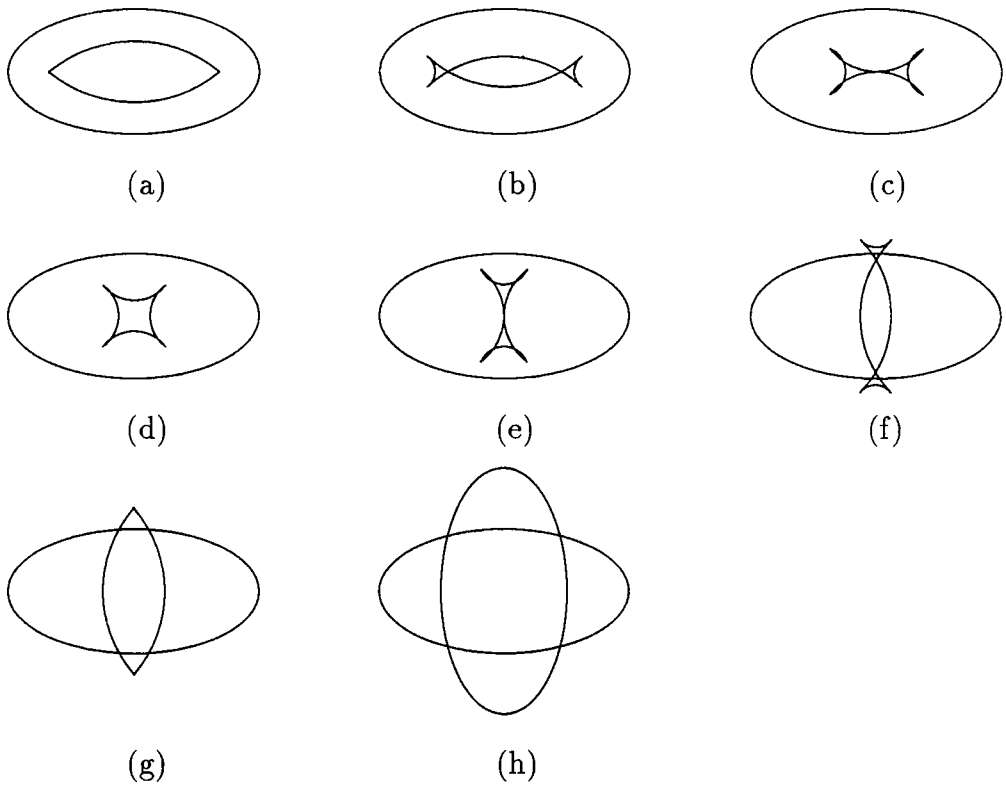


Figure 5.1 Local (one-sided) offset of a ellipse

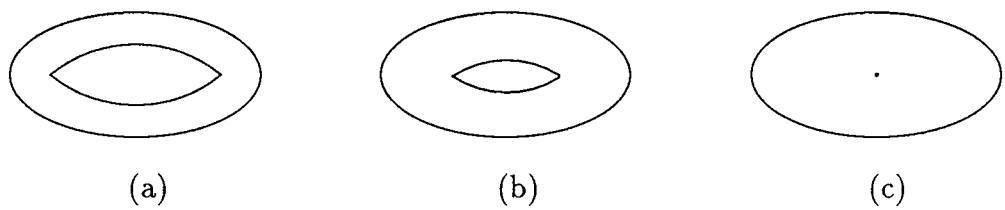


Figure 5.2 Global (one-sided) offset of a ellpise

Consider the curve C with the implicit equation $f(x,y) = 0$. Let (u_1, v_1) be a point on f and (x,y) be a point on the normal of f at (u_1, v_1). Let f_x denote the differential of f with respect to x. In 2D, the system of equation for the local offsets with distance d to the boundary in the dimensionality paradigm is [33]:

$$f(u,v) = 0$$

$$f_x(y-v) - f_y(x-u) = 0$$

$$(x-u)^2 + (y-v)^2 = d^2$$

The first equation states that the point (u_1, v_1) is on f. The second equation assures the fact that (x,y) is on the normal of f at (u_1, v_1). And, the third equation asserts that the distance from the boundary curve to offset curve is equal to d. Notice that in the general case, the degree of freedom is 1, so that the offset of the curve should be a curve in 2D.

In 3D, the system of equations for the local offset surface can be derived similarly. It will have 4 equations with 6 variables. One equation states that a specified the point (u,v,w) is on the surface $f(x,y,z) = 0$, two equations assure that an associated point (x,y,z) is on the normal of the surface at the point (u,v,w), and the last equations assure that the distance between the two points is equal to the offset distance. So, the system of equations is:

$$f(u,v,w) = 0$$

$$f_x(y-v) - f_y(x-u) = 0$$

$$f_y(z-w) - f_z(y-v) = 0$$

$$(x-u)^2 + (y-v)^2 + (z-w)^2 = d^2$$

Now, let us consider the system of equations which generates the local offset surface of a space curve. Assume that the curve is the intersection of two surfaces $f(x,y,z) = 0$ and $g(x,y,z) = 0$, and that they intersect transversally. Let (x,y,z)

be a point on the offset surface and (u, v, w) be a point on the curve. The system of equations for the offset surface of $f \bigcap g$ is:

$$f(u, v, w) = 0$$

$$g(u, v, w) = 0$$

$$((f_x, f_y, f_z) \times (g_x, g_y, g_z)) \cdot (x - u, y - v, z - w) = 0$$

$$(x - u)^2 + (y - v)^2 + (z - w)^2 = d^2$$

The offset surface of a space curve is also called a canal surface. [42, 51, 52, 53].

Consider the modified cyclographic map $S(C)$ of a curve C, and the restricted cyclographic map $\overline{S}(C)$ of C of chapter 2. Then the intersection curve of $z = d$ and $S(C)$ and of $z = d$ and $\overline{S}(C)$ are the local and global offsets curve of C, respectively, with offset distance equal to d.

Local offsets play a role in geometric optics[23]. Global offsets, in contrast, are a subset of the local offset, and are important in engineering applications[37]. Since global offsets involve global distance computations, they are algorithmically more difficult to determine. Briefly, the accepted strategy for computing global offsets is:

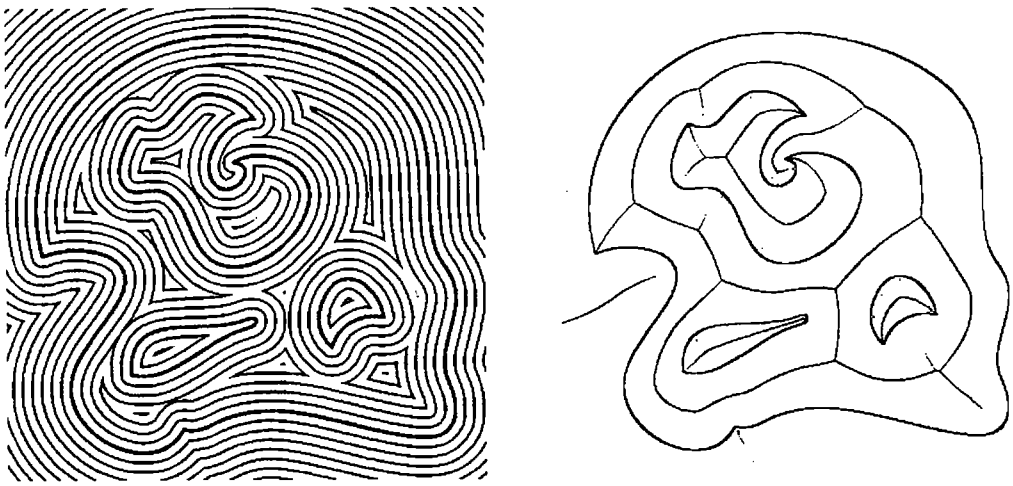
> Determine the local offset. Then remove all those parts that are not at global distance d, by trimming certain branches or regions bounded by self-intersections[37].

We propose as alternative the following approach:

> Integrate the global distance function up to distance d[14], and extract the global offset as a level set of this function.

Our approach requires discretizing the curve or surface in the ambient space.

The locus of the self-intersections at which local offset regions border global ones can be characterized by the MA. Computing the intersection of the MA with a d-offset is equivalent to determining those self-intersections at which the traditional approach to computing global offsets will cut away nonglobal offset segments or regions. Figure



(a) (b)

Figure 5.3 Global offset and MA for the same domain

5.3(a) shows the global offsets of the curve shown in Figure 2.4. For comparison, we show the MA points for the same domain in Figure 5.3(b). Notice that the self-intersection points, or the cusp, in the offsets curves generate the MA curve.

Since our algorithm discretizes the ambient space, and then computes the global distance based on this discretization, it is fairly insensitive to the problem dimension. We have implemented two-dimensional versions, for offsetting curves, and only small changes would be needed for offsetting surfaces in 3D with the same approach.

5.2 CAGD Approach to offset computations

Computer-aided geometric design (CAGD) has produced an extensive literature on offsets of curves and surfaces owing to the importance of the subject [18, 19, 23, 21, 30, 38, 67]. The majority of the methods constructs approximate representations for local offsets of curves and investigate methods to trim them back to global offsets, or restrict applicability to those cases in which the local and global offsets coincide.

When approximating the offset, special properties of the base curve representation are customarily exploited. For example, offsets from (rational) parametric curves and surfaces can be constructed approximately by transforming the control points [19].

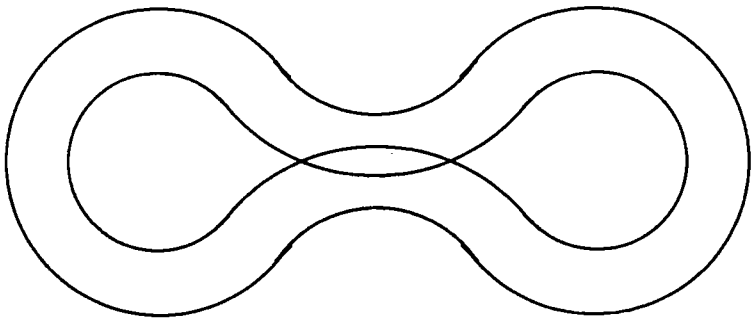


Figure 5.4 A Difficult Self-Intersection

Exact offset representations that belong to the same class of curves or surfaces usually do not exist in the parametric case, except in very special cases [22]. Error estimates for the deviation of the approximation from the true offset can be given [18]. If the error is unacceptably large, the base curve or surface can be suitably subdivided and the offset approximated anew.

In Figure 5.1(a) to (b), the offset distance exceeds the smallest radius of curvature of the base curve. This always results in self-intersection[36], and there are robust and efficient mathematical criteria that can be used to detect such situations [18]. In contrast, self-intersections such as the one shown in Figure 5.4 are harder to detect, and no efficient and comprehensive strategies for detecting them appear to be known. In consequence, it is not attractive to compute global offsets by trimming local offsets.

If the base curves and surfaces are algebraic, then the (local) offset is again algebraic, and in principle one can derive an implicit representation of the local offset using elimination techniques [29]. In practice, this approach is unattractive for several reasons:

- The symbolic computation required to derive the implicit form can be extremely expensive [30].

- The algebraic degree of the offset can be very high, even for base curves and surfaces that have low algebraic degree [20]. Thus, subsequent computations with the implicit form could be numerically difficult.
- The implicit equation represents the local offset and offers no advantage for finding self-intersections.

Some of these difficulties are circumvented by the dimensionality paradigm [30, 31]. Again, the local offset is obtained and has to be trimmed if the global offset is wanted. Trimming could be based on global surface evaluation algorithms [10]. This approach is largely unexplored as yet.

5.3 Discrete Algorithms for Global Offsets

If the offset distance is small, that is, if the area enclosed by the base curve and its offset is small in comparison to the total domain area, then the fixed sweeps of Danielson's method result in more work than necessary. We therefore reorganize the algorithm so as to eliminate unnecessary computations.

We wish to reorganize the computation so that the grid points are processed by increasing distance from the boundary. If this can be done efficiently, then the offset of a closed curve can be computed in time that depends on the offset distance and the curve length, rather than on the entire enclosed area.

We process a grid point by assigning its distance. Based on this distance, a candidate distance is proposed for each of its neighbors, and each neighbor becomes a candidate for processing. There will be a queue for candidate grid points. Note that a grid point can appear several times in the queue, with proposed distances determined from different neighbors.

To process grid points by increasing distance, we add them to a priority queue indexed by the squared distance from the boundary. Initially, the discretized base curve grid points are added to this queue, with a zero distance key. Grid points to be added at a later time will have nonzero distance keys. Along with the distance

key and the grid point coordinates, we store the distance amplitudes. The distance amplitudes Δx and Δy are signed, so we know the direction in which the nearest boundary point lies. By convention, the orientation of the vector $(\Delta x, \Delta y)$ is from a nearest boundary point (i_0, j_0) to the grid point $(i_0 + \Delta x, j_0 + \Delta y)$.

When removing the grid point (i, j) from the queue, it is processed as follows: Let (i, j) have distance amplitudes $(\Delta x, \Delta y)$ according to our sign conventions. Enter the distance of (i, j) into a matrix as element $M[i, j]$, unless $M[i, j]$ has already been assigned. Note that an entry already assigned has already the correct distance to the boundary, because all queue elements are processed by increasing distance. If $(\Delta x)^2 + (\Delta y)^2 < d^2$, where d is the offset distance, then add to the priority queue the entries $(i + r, j + s)$, with distance amplitudes $(\Delta x + r, \Delta y + s)$, where $r = -1, 0, 1$ and $s = -1, 0, 1$ and r and s are not both zero, and such that $(\Delta x + r)^2 + (\Delta y + s)^2 > (\Delta x)^2 + (\Delta y)^2$.

This scheme will still require time proportional to the matrix size unless we can initialize the entire matrix in constant time, independent of its size, access its entries in constant time, and determine for each accessed entry whether it has been reassigned since initialization. This can be done with standard techniques summarized in the appendix B.

The work done by the algorithm to assign M amounts to examining each grid point (i, j) that is at a distance no greater than the offset distance and computing the distance amplitudes for its immediate neighbors. It is thus proportional to $N = A/h^2$, where A is the area enclosed by curve and offset, and h is the grid spacing. The priority queue updates require an additional logarithmic factor, so that the asymptotic complexity of the algorithm is $O(N \log N)$. We summarize the final algorithm.

Algorithm 1

1. Initialize M to *unassigned* and Q to consist of all curve points with distance amplitudes $(0, 0)$.
2. While Q is not empty, delete the next (i, j) from Q and perform step 3.

3. Let (a, b) be the distance amplitudes of (i, j). If $M[i, j]$ is already assigned, do nothing. Otherwise, assign the distance of (i, j) to $M[i, j]$. Enter into the queue Q all entries $(i + r, j + s)$, where $r = -1, 0, 1$ and $s = -1, 0, 1$, r and s are not both zero, and $(a + r)^2 + (b + s)^2 > a^2 + b^2$.

Note that Step 3 is slightly different for boundary points, unless we compute the offset on both sides.

The offset curve can be extracted from M by examining all its entries. Alternatively, let a frontier point be any grid point with at least one neighbor whose distance exceeds the offset distance d. Clearly the offset is near frontier points, and can be extracted by processing them and their neighbors, without examining other matrix entries. Frontier grid points can be registered when they are entered into M.

The offset could be left in discretized form, or an approximation of it can be constructed by, say, interpolation. Greater accuracy is obtained by iteratively refining frontier grid points to true offset points[31].

Note that Algorithm 1 generalizes to offsetting surfaces in three dimensions. Here we have to work with a three-dimensional array and three distance amplitudes, $(\Delta x, \Delta y, \Delta z)$.

Processing grid points by increasing distance is accomplished by a priority queue. Since distance amplitudes, and hence the squared distances are integer-valued when measured in multiples of the grid spacing, we can reduce the processing time by a factor of $\log N$ if we can find the next closest unprocessed grid point in constant time. This can be done as follows.

Note that every distance key of interest is an integer between 1 and D^2, where $D = \lceil d \rceil$ is the offset distance measured in multiples of the grid spacing. We create an index array I, where position $I[k]$ is reserved for the distance key k. The entry $I[k]$ heads a list of unprocessed grid points, all at a conjectured squared distance k.

We add the grid point (i, j) with distance amplitudes (a, b) to the list headed by $I[a^2 + b^2]$. Now it is clear that we can process the grid points by first processing the list at $I[1]$, then the list at $I[2]$, and so on. Since distances are nondecreasing, we will

never add a grid point to any list that is closer to the boundary than the points in the list we are currently processing. In all other respects, the algorithm remains the same.

It would seem that the array I is in size proportional to D^2. However, note that the distance of an unprocessed grid point to be added is not too far from the grid point currently processed. When processing a point at the squared distance u_1, a new point to be added is at a squared distance no greater than $u_2 = (a+1)^2 + (b+1)^2$, where $a^2 + b^2 = u_1$ and $a, b \leq D$. Furthermore, every point at a squared distance less than u_1 has already been processed. But

$$u_2 - u_1 \leq 4D + 2$$

so that at any time there are at most $4D + 2$ nonempty lists. Thus, it suffices to allocate for I an array of size $4D + 3$ as long as this space is used as a circular queue [41]. Therefore, the space requirements for the index vector I are proportional to the offset distance.

Note that we have to initialize entries in I when they are used for the first time, as well as the intervening, empty items of smaller squared distance. Initialization requires $O(D^2)$ steps, but D^2 is dominated by N. We summarize this algorithm as follows, without going into the details of managing the vector I as a circular queue.

Algorithm 2

1. Initialize M to *unassigned*, and I to *empty*. Add the curve points to the list $I[0]$.
2. For k from 0 to D^2 do Step 3.
3. For each entry in list k do the following. If $M[i,j]$ is already assigned, do nothing. Otherwise, assign the distance of (i,j) to $M[i,j]$. Add all entries $(i+r, j+s)$ to list $I[(a+r)^2+(b+s)^2]$, where (a,b) are the distance amplitudes of (i,j), and r and s have been selected as in Algorithm 1.

The time required by Algorithm 2 is $O(N)$, and so improves Algorithm 1 by a factor of $\log N$.

5.4 Iteration and Interpolation Algorithms

Since Algorithms 1 and 2 compute distances from grid points only, the discretization of the base curve or surface into a set of grid points automatically introduces errors proportional to the grid spacing. We now diminish these errors using both iteration and interpolation. The main consequence of the changed approach is that integer arithmetic is no longer appropriate. Also, the work per grid point processed increases, but this increase can be offset by working with a coarser grid.

For each grid point, we will determine the following information: As before, we compute the signed distance amplitudes Δx and Δy and the squared distance $(\Delta x)^2 + (\Delta y)^2$. In addition, we identify the boundary segment or patch on which a nearest point lies. For parametric boundary elements we also record the parameter value(s). For grid points near the boundary, the additional data is determined first, say by linear interpolation of the intersection of the boundary with the grid lines. Then the distance amplitudes and the squared distance are computed from it. More precisely, with s_1 and s_2 the parameter values at the intersection with the grid lines, the secant is drawn and the perpendicular to it through the grid point is determined. Suppose the footpoint intersects the secant in the ratio $v : u$. Then the parameter is approximated by $s = s_1 + v(s_2 - s_1)/(u + v)$, and the distance amplitudes by Δx and Δy. See also Figure 5.5. The grid point data could be refined by Newton iteration.

When processing the neighbor grid points, either with Danielson's algorithm or with our Algorithms 1 or 2, the computations updating the distance amplitudes are as before. To increase accuracy, we can store the distance amplitude $mh + \delta$ as the pair (m, δ), since distance amplitudes increase by integer multiples of the grid spacing h. The additional data describing the nearest boundary point are copied. This part of the algorithm is the first phase.

Having so constructed an approximate distance of grid points, the frontier grid points are now "polished" using Newton iteration. In consequence, their distance to the boundary will be as accurate as possible. Then, the intersection of the offset

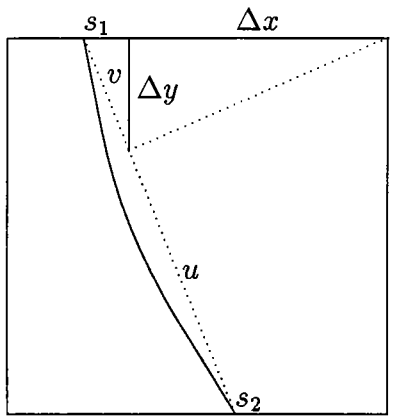


Figure 5.5 Grid Point Data Near Boundary

with grid lines is determined by linear interpolation. Again, Newton iteration can be used to polish the point. These iteration and interpolation steps comprise the second phase of the algorithm.

When a grid or offset point lies close to a self-intersection, the different distances proposed by several neighbors will be close. This fact should be recorded, for the iteration in Phase 2 changes distances slightly, and so the point may ultimately lie closer to another part of the boundary. Thus, the originally proposed footpoint on the boundary may not be the true one. In this situation we have to iterate the distance from each footpoint associated with a neighbor who proposed approximately the same distance. The algorithm so derived and based on the data structures of Algorithm 1 will be called Algorithm 3.

5.5 Experimental Results

We implemented Algorithms 1 and 2, and a version of the iteration and interpolation method based on Danielson's algorithm. Input and output can be provided in several ways. We can interface to //ELLPACK [39] which provides us with graphical tools for defining 2D domains bounded by Bézier curves of arbitrary degree and with holes allowed. Alternatively, we can draw any shape with Framemaker, a commercial document preparation system. The shapes consist of cubic Bézier splines, ellipses, and line segments, and need not form domains.

We timed the execution of Danielson's method and of Algorithms 1 and 2 on the three sample shapes shown in Figure 5.6(a)(c)(e), using a grid size of 500×500. Their associated offsets, which have distance to the boundary of 10,20,30, and so on, are shown in Figure 5.6(b)(d)(f). Danielson's method processed all $250,000$ grid points, whereas Algorithms 1 and 2 processed only grid points up to the offset distance. The performance timed on a SUN SPARC 2 workstation is summarized in Table 5.1. All computations correspond to the 8SED version. The results show that Algorithm 2 strictly improves Algorithm 1.

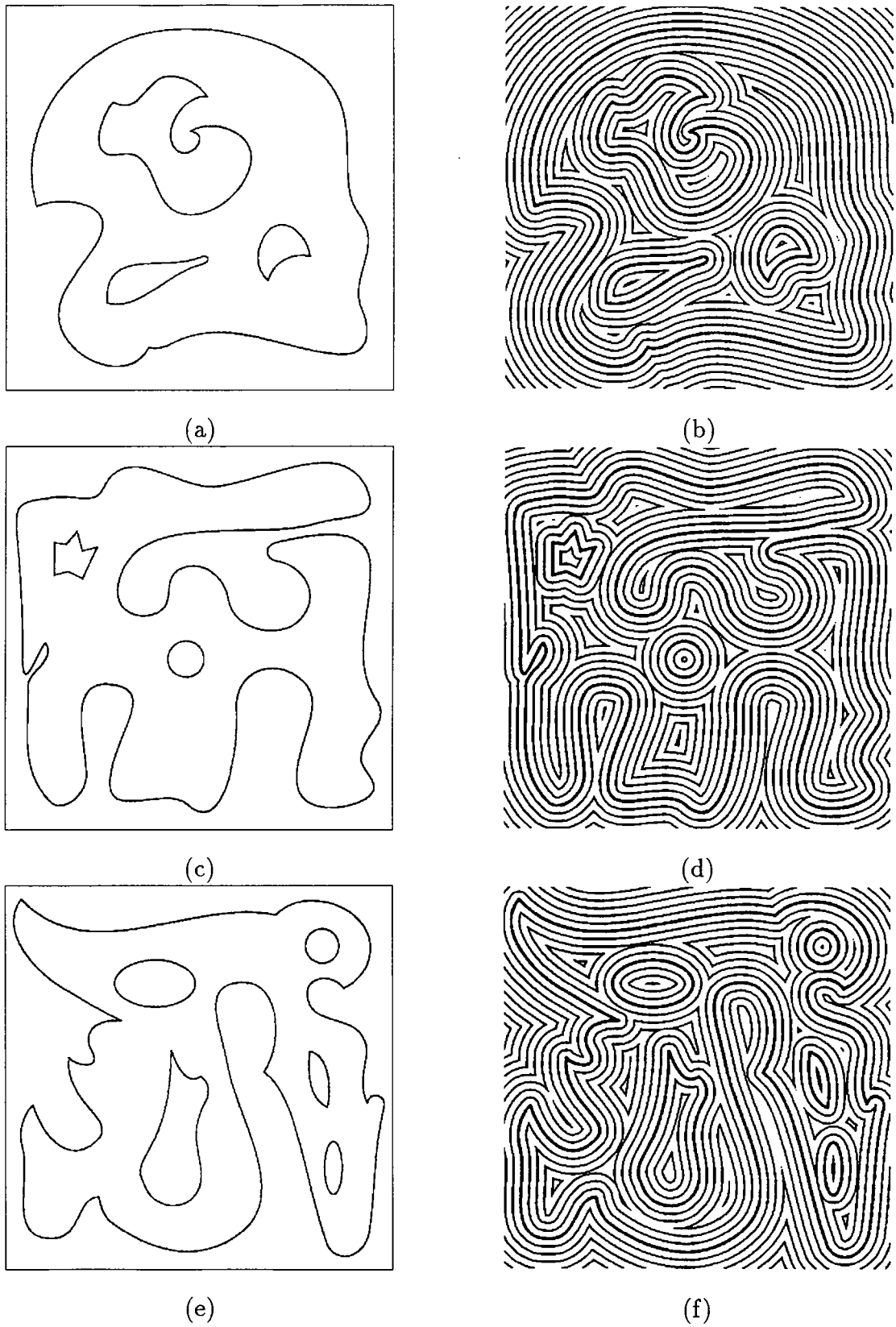


Figure 5.6 Sample domains and their offsets

Table 5.1 Performance of Danielson's algorithm and of Algorithms 1 and 2

Danielson's Algorithm

Domain A: 16.85 sec	Domain B: 17.61 sec	Domain C: 16.86 sec

Algorithms 1 and 2

Domain	Offset dist	N	Time Alg 1 (sec)	Time Alg 2 (sec)
A	10	32,718	14.27	9.13
	20	62,648	28.80	19.85
	30	91,558	42.56	28.42
B	10	41,587	19.20	12.50
	20	78,932	36.91	25.78
	30	111,797	51.60	33.60
C	10	47,419	21.20	13.61
	20	87,918	41.17	27.59
	30	114,755	57.20	34.08

We determined for which value of N Algorithm 2 is as fast as Danielson's algorithm. On average, the algorithm requires $3.035 \cdot 10^{-4}$ seconds per grid point processed, whereas Danielson's method requires $0.684 \cdot 10^{-4}$ seconds per grid point processed. Thus, Algorithm 2 does 4-5 times as much work per grid point. In consequence, Algorithm 2 is better than Danielson's method for those offset distances at which the enclosed area is less than 22 % of the area of the entire grid.

Adding iteration and interpolation impacted performance significantly. The algorithm was run on a 30×30 grid using Newton iteration to get accurate distance at grid points and using linear interpolation to extract the offset. This required 0.65 sec. Visual inspection of the output showed that the offset so determined was roughly equal in quality than the offset determined by Algorithm 2 on a 300×300 grid, which required in contrast 8.74 sec.

6. CONCLUSIONS AND FUTURE WORK

In this thesis, the theory for the MAT of 2D solid has been investigated. We prove the uniqueness, divisibility, connectedness and reversibility properties for 2D solids. Algorithms are proposed to extract the MAT based on these properties. For example, the Newton and marching method can find all of the MAT from an MAT start point because of the connectedness property of the MAT. The Euclidean distance transform for 2D and 3D solids on the grid points is computed before extracting the MAT or offsets of the solids. Danielson's 8SED algorithm is used for the DT of the 2D solids because it is closed to the Euclidean distance transform. An similar 26SED algorithm extends the method to 3D and is used for the DT of 3D solids.

Several Methods have been proposed to extract the MAT for 2D and 3D solids. Two criteria, the maximal circle criterion and the equal distance criterion, are used to extract the MAT. Rosenfeld and Pfaltz's method, Danielson's method and interpolation/extrapolation method use the maximal circle criterion by testing each grid point locally to decide whether a specified grid point is an MA point. With this approach, the results of the algorithms are a set of grid points. Although¿ it is straightforward to extend the computation to 3D, such output is hard to visualize when the grid MAT points are projected to the 2D screen, especially for large numbers of grid points with lots of noisy points. Newton and marching and grid edge interpolation use the equal distance criterion. The results of these algorithms are a number of connected line segments. The algorithms can be extended to 3D, using more complex data structures, so that the result MA points are approximated by facets, line segments and vertices.

Detecting self-intersections in offsets is a problem that has both a mathematical and a combinatorial character. Some self-intersections can be detected based on local criteria applied to boundary elements, but others require evaluating the spatial

relationship between unknown parts of the base curve, and this is difficult for the traditional offset algorithms in the literature.

We have addressed the problem by discretizing ambient space, and by "posting" a geometric datum in this common space. Self-intersection is thereby reduced to the problem of deciding whether a particular array element has already been assigned. The approach poses an exacting trade-off between the accuracy that is achieved and the memory that is required. Greater accuracy demands denser grids, and so larger arrays are needed, especially in three dimensions. Algorithms 1 and 2 use this approach.

The memory requirements of Algorithms 1 and 2 are dominated by the matrix M. The matrix could be implemented differently. For example, by storing each element in a balanced search tree only $O(N)$ memory is needed. This is a big improvement for small offset distances, but costs a logarithmic factor in the access time. The impact on the overall performance can be judged from the speed differentials of Algorithms 1 and 2.

Hybrid schemes that combine the discrete algorithm with interpolation and iteration allow using coarser grids without sacrificing accuracy, and save both time and space. Algorithms 3 is an examples. Our experience with the implementation suggests exploring adaptive schemes in which an initial coarse grid is refined selectively only in critical areas.

Future work on the MAT will focus on the applications of the MAT of 3D solids. Many questions can be asked about the application, such as:

- Is MAT good to be used as shape representation in solid modeling?
- Is the point/solid classification problem for such a representation simplier than conventional representations?
- Are the union, difference and complement operations easier to implement than with the conventional representations?

- Is the MAT representation helpful for calculating the distance to the boundary of a solid?
- Is the MAT representation helpful for blending the surface?
- Can the MAT make the finite-element mesh generation problem in 3D easier?

These questions should be investigated in future work.

BIBLIOGRAPHY

BIBLIOGRAPHY

[1] Norman I Badler and Clayton Dane. The medial axis of a coarse binary image using boundary smoothing. *IEEE*, 2:286–291, 1979.

[2] Pierre Bézier. *The Mathematical Basis of the UNISURF CAD System.* Butterworths, 1986.

[3] H. Blum. A transformation for extracting new descriptors of shape. In W. Whaten-Dunn, editor, *Models for the Perception of Speech and Visual Form*, pages 362–380. MIT Press, Cambridge, MA, 1967.

[4] Harry Blum. Biological shape and visual science (part i). *Journal Theoretical Biology*, 38:205–287, 1973.

[5] Harry Blum and Roger N. Nagel. Shape descriptiion using weighted symmetric axis features. *Pattern Recognition*, 10:167–180, 1978.

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

[7] Gunilla Borgefors. Distance transformations in arbitrary dimensions. *Computer Vision, Graphics, and Image Processing*, 27:321–345, 1984.

[8] Gunilla Borgefors. Distance transformations in digital images. *Computer Vision, Graphics, and Image Processing*, 34:344–371, 1986.

[9] Christoph M Hoffmann Ching-Shoei Chiang and Robert E. Lynch. How to compute offsets without self-intersection. In *Proceedings SPIE Conference on Curves and Surfaces in Computer Vision and Graphics*, November 1991.

[10] Jung-Hong Chuang. *Surface Approximations in Geometric Modeling.* PhD thesis, Purdue University, Computer Science, August 1990.

[11] T.C Lee C.N. Chu, R.L.Kashyap and I.C. You. Castability evaluation via skeleton-based modeling, 1990-1991.

[12] Hilbert Cohn-Bossen. *Geometry and the Imagination.* Ghelsea Publishing Company, 1932.

[13] Conte and de Boor. *Elementary Numerical Analysis – An Algorithmic Approach.* McGraw-Hill, Inc., 1980.

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

[15] D.T.Lee and R.L. Drysdale. Generalization of voronoi diagrams in the plane. *SIAM J. Comput.*, 10(1):73–87, February 1981.

[16] Richard O. Duda and Peter E. Hart. *Pattern Classification and Scene Analysis.* Wiley-interscience Publication, 1973.

[17] Debasish Dutta and Christoph M. Hoffmann. A geometric investigation of the skeleton of csg objects. In *Proc. ASME Conf. Design Automation*, Chicago, 1990.

[18] G. Elber and E. Cohen. Error bounded variable distance offset operatorand surfaces. *International J. of Computational Geometry and Applications*, 1:67–78, 1991.

[19] R. T. Farouki. The approximation of nondegenerate offset surfaces. *Computer Aided Geometric Design*, 3:15–43, 1986.

[20] R. T. Farouki and C. A. Neff. Algebraic properties of plane offset curves. *Computer Aided Geometric Design*, 7:101–128, 1990.

[21] R. T. Farouki and C. A. Neff. Analytic properties of plane offset curves. *Computer Aided Geometric Design*, 7:83–100, 1990.

[22] R. T. Farouki and T. Sakkalis. Pythagorean hodographs. Technical Report RC-15223, IBM Yorktown Heights, 1990.

[23] Rida T. Farouki and Jean-Claude A. Chastang. Evolving wavefronts as algebraic curves. Technical Report Technical Report RC-16381, IBM Yorktown Heights, 1990.

[24] Ronald N. Goldman and James R. Miller. Detecting and calculating conic sections in the intersection of two natural uadric surfaces, part i: Detection.

[25] Ronald N. Goldman and James R. Miller. Detecting and calculating conic sections in the intersection of two natural uadric surfaces, part 2: Calculation.

[26] William C. Graustein. *Introduction to Higher Geometry.* The Macmillan company, 1937.

[27] Halit Nebi Gürsoy. Shape interrogation by medial axis transform for automated analysis. Ph.D. Thesis.

[28] C. Judith Hilditch. Linear skeletons from square cupboards. *Machine Intelligence*, 3:325–420, 1968.

[29] C. M. Hoffmann. *Geometric and Solid Modeling.* Morgan Kaufmann, San Mateo, Cal., 1989.

[30] C. M. Hoffmann. Algebraic and numerical techniques for offsets and blends. In S. Micchelli M. Gasca, W. Dahmen, editor, *Computations of Curves and Surfaces*, pages 499–528. Kluwer Academic, 1990.

[31] Christoph M. Hoffmann. A dimensionality paradigm for surface interrogations. *CAGD*, 7:517–532, 1990.

[32] Christoph M. Hoffmann. How to construct the skeleton of csg objects. In A. Bowyer and J. Davenport, editors, *The Mathematics of Surfaces IV.* Oxford University Press, 1990.

[33] Christoph M. Hoffmann. Computer vision, descriptive geometry and classical mechanics. In B. Falcidieno and I. Hermann, editors, *Computer Graphics and Mathematics.* Springer Verlag, 1992.

[34] Christoph M. Hoffmann and George Vaněček, Jr. Fundamental techniques for geometric and solid modelling. In C.T. Leondes, editor, *Advances in Control and Dynamics.* Academic Press, 1991.

[35] Christoph M. Hoffmann and Pamela J Vermeer. Eliminating extraneous solutions in curve and surface operations. *IJCGA*, 1, 1991.

[36] J. Hoschek. Offset curves in the plane. *Computer Aided Design*, 17:77–82, 1985.

[37] J. Hoschek. *Grundlagen der Geometrischen Datenverarbeitung.* Teubner Verlag, Stuttgart, 1989.

[38] J. Hoschek and N. Wissel. Optimal approximate conversion of splineapproximation of offset curves. *Computer Aided Design*, 20:475–483, 1988.

[39] E. N. Houstis, T. S. Papatheodorou, and J. R. Rice. Parallel ellpack: An expert system for parallel processingpartial differential equations. *Math. Comp. Simulation Journal*, 31:487–508, 1989.

[40] David G. Kirkpatrick. Efficient computation of continuous skeletons. *IEEE*, 2:18–27, 1979.

[41] D. E. Knuth. *The Art of Computer Programming, Vol. 1.* Addison-Wesley, Reading, Mass., 1968.

[42] Jan J. Koenderink. *Solid Shape.* The MIT Press, Cambridge, Massachusetts, London, England, 1990.

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

[44] Fritz Prinz Levent Gursoz and Atul Suchalkar. Continuous skeletons from discrete objects. Technical Report Technical Report CMU EDRC 24-87-92, Engineering Design Research Center, Camegie Mellon University, 1992.

[45] C. G. Armstrong M.A.PRice, T.K.H. Tam and R. M. Mckeag. Computing the branch points of the voronoi diagram of a planar object using a point delaunay triangulation algorithm, January 1990.

[46] U. Montanari. A method for obtaining skeletons using a quasi-euclidean distance. *Journal of the ACM*, 15(4):600–b24, 1968.

[47] U. Montanari. Continuous skeletons from digitized images. *Journal of the ACM*, 16(4):600–624, 1969.

[48] E. Müller and J. Krames. *Die Zyklographie.* Franz Deuticke, Leipzig und Wien, 1929.

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

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

[51] Z. Nadenik. Die ungleichungen fur die masszahlen der geschlossenen kanalflachen. *Czech. math J.*, 16, 1966.

[52] Z. Nadenik. Die ungleichungen fur die masszahlen der geschlossenen kanalkorper. *Czech. math J.*, 17, 1967.

[53] Z. Nadenik. Zur geometrie in grossen derkugelkongruenzen. *Czech. math J.*, 18, 1968.

[54] Joseph O'Rourke and Norman Badler. Decomposition of three-dimensional objects into spheres. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-1(3):295–305, 1979.

[55] Stanley Osher and James A. Sethian. Fronts propagating with curvature-dependent speed: Algorithms based on hamilton-jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[56] N.M.. Patrikalakis and H.N. Gürsoy. Shape interrogation by medial axis transform. Technical Report Design Lab. Memo. 90-2, Sea Grant College Program, MIT, 1990.

[57] Theodosios Pavlidis. A thining algorithm for discrete binary images. *Computer Graphics and Image Processing*, 13:142–157, 1980.

[58] Theodosios Pavlidis. An asynchronous thining algorithm. *Computer Graphics and Image Processing*, 20:133–157, 1982.

[59] John L. Pfaltz and Azriel Rosenfeld. Computer representation of planar regions by their skeleton. *Communications of the ACM*, 10(2):119–125, 1967.

[60] John L. Pfaltz and Azriel Rosenfeld. Distance functions on digital pictures. *Pattern Recognition*, 1:33–61, 1968.

[61] Otis Philbrick. Shape description with the medial axis transformation. In *Pictorial Pattern Recognition*, pages 395–407. Thompson Book Co., 1968.

[62] J. Plücker. On a new geometry of space. *Phil. Trans R. Soc.*, 155, 1865.

[63] F.P. Preparata. The medial axis of a simple polygon. In *Proc. 6th Symp. Math. Foundations of Computer Science*, pages 443–450, September 1977.

[64] Ingemar Ragnemalm. *Generation of Euclidean Distance Maps.* PhD thesis, Linkoping Studies in Science and Technology, 1990.

[65] Azriel Rosenfeld. A characterization of parallel thinning algorithms. *Information and Control*, 29:286–291, 1975.

[66] Azriel Rosenfeld and John L. Pfaltz. Sequential operations in digital picture processing. *Journal of ACM*, 13(4):471–494, 1966.

[67] J. Rossignac and A. Requicha. Offsetting operations in solid modeling. *CAGD*, 3:129–148, 1984.

[68] D. M. Y. Sommerville. *Introduction to the Geometry on N Dimensions.* Dover Publications, Inc. New York, 1929.

[69] Peter V. De Souza and Philip Houghton. Computer location of medial axes. *Computers and Biomedical Research*, 10:333–343, 1977.

[70] Michael Spivak. *A Comprehensive Introduction to Differential Geometrh*, volume Two. Waltham, Mass., 1979.

[71] Sabine Stifter. The roider method: A method for static and dynamic collision detection. In C. Hoffmann, editor, *Issues in Robotics and Nonlinear Geometry.* JAI press, 1990.

[72] Sabine Stifter. An axiomatic approach to voronoi-diagrams in 3d, 1991.

[73] K. Strubecker. *Differentialgeometrie I.* Sammlung Göschen Bd. 1180/1180a, Walter de Gruyter, Berlin, Germany, 1969.

[74] Satoshi Suzuki and Keiichi Abe. Sequential thining of binary pictures using distance transformation. In *8th International Conference on Pattern Recognition*, pages 289–292, 1986.

[75] T.K.H Tam and C.G. Armstrong. 2d finite element mesh generation by medial axis subdivision, 1992.

[76] J. Tang V. Srinivasan, L. Nackman and S. Meshkat. Automatic mesh generation using the symmetric axis transformation of polygonal domains. Technical Report Technical Report RC 16132, IBM Yorktown Heights, 1990.

[77] Yun Xia. Skeletonization via the realization of the fire front's progagation and extinction in digital binary shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(10):1076–1086, October 1989.

[78] John A.. Goldak Xinhua Yu and Lingxian Dong. Constructing 3-d discrete medial axis. In *Proc ACM Symp Solid Modeling Found. and CAD/CAM Applic.*, pages 481–492, 1991.

[79] Hiromitsu Yamada. Complete euclidean distance transformation by parallel operation. In *7th International Conference on Pattern Recognition*, pages 69–71, 1984.

[80] Qin-Zhong Ye. The signed euclidean distance transform and its application. In *9th International Conference on Pattern Recognition*, volume 1, pages 495–499, 1988.

APPENDICES

Appendix A: The Plücker Coordinates and the nearest approach pairs between two primitive objects

A.1 Introduction

This appendix gives algorithms for finding closest approach points between two geometric shapes. Plücker coordinates are used in some of the cases considered, because they simplify the computation.

There are 6 sections in this appendix. The theory of Plücker coordinates is sketched without proof in section 2. Notations and definitions are given in section 3. With them, simpler computations can be given for finding the nearest approach pairs in some cases. Some theorems that are used in later sections are also stated. Sections 4, 5 and 6 propose some methods to find the shorest approach pairs between two objects. The object can be a CSG primitive surface, a curve generated by two CSG primitive surfaces, or an algebraic surface or curve. The shortest approach pairs can be a 0-surface such as two points, a 1-surface surface such as two lines, two circles, etc., or a 2-surface such as two spheres or two tori. Sections 4 and 5 consider cases where the shapes come from CSG primitive surfaces. Algebraic surfaces and curves are considered in section 6.

A.2 Plücker Coordinates

This section concerns the computational aspects of the geometry of lines in 3D using Plücker coordinates. Many properties of Plücker coordinates are listed without proof. Some of the proofs can be found in [26, 68].

A.2.1 Plücker ray coordinates

Let L be a line that passes through two points $x = (x_1, x_2, x_3, x_4)$ and $y = (y_1, y_2, y_3, y_4)$, where the point coordinates are homogeneous with x_1 and y_1 the homogeneous variable. Let $p_{ij} = x_i y_j - x_j y_i$, where $i \neq j$. The six-tuple

$$(p_{12}, p_{13}, p_{14}, p_{34}, p_{42}, p_{23})$$

gives the Plücker *ray coordinates* of the line. We write $L_{ray}(x, y) = (\mathrm{P}, \hat{\mathrm{P}})$ where $\mathrm{P} = (p_{12}, p_{13}, p_{14})$ and $\hat{\mathrm{P}} = (p_{34}, p_{42}, p_{23})$. For simplicity, we usually write $L_{ray} = (\mathrm{P}, \hat{\mathrm{P}})$.

A.2.2 Plücker axis coordinates

Let $a = [a_1, a_2, a_3, a_4]$ represent the plane $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = 0$ and let $b = [b_1, b_2, b_3, b_4]$ represent the plane $b_1x_1 + b_2x_2 + b_3x_3 + b_4x_4 = 0$. Assume that these two planes intersect in the line L. The six-tuple

$$(q_{12}, q_{13}, q_{14}, q_{34}, q_{42}, q_{23})$$

where $q_{ij} = a_i b_j - a_j b_i$ are the Plücker *axis coordinates* of L. We write $L_{axis} = (\mathrm{Q}, \hat{\mathrm{Q}})$ where $\mathrm{Q}=(q_{12}, q_{13}, q_{14})$ and $\hat{\mathrm{Q}} = (q_{34}, q_{42}, q_{23})$. Note that Plücker axis coordinates have many of the properities of the Plücker ray coordinates. The connection between these two coordinate systems is given by:

Theorem A.31 The ray coordinates $(\mathrm{P}, \hat{\mathrm{P}})$ and axis coordinates $(\mathrm{Q}, \hat{\mathrm{Q}})$ of a line are connected by the equations

$$(p_{12} : p_{13} : p_{14} : p_{34} : p_{42} : p_{23}) = r(q_{34} : q_{42} : q_{23} : q_{12} : q_{13} : q_{14})$$

That is, there exists an $r \neq 0$, such that $(\mathrm{P}, \hat{\mathrm{P}})=\mathrm{r}(\hat{\mathrm{Q}}, \mathrm{Q})$ as vectors.

A.2.3 Properties

Before further describing the properties of Plücker coordinates, we define some notations for operations on points and vectors. Let us define the polymorphic operators "+", and "−", denoting sum and difference of two points, two vectors, or

a point and a vector. The symbol "$*$" denotes the product of two scalars, a scalar and a point, or a scalar and a vector. The symbols "$\cdot$" and "$\times$" denote the inner product and the cross product of two vectors. All of the operators are defined for 3D Euclidean coordinates and the corresponding homogeneous coordinates.

Plücker coordinates have the following elementary properties:

1. $p_{ij} = -p_{ji}$

2. $\mathrm{P} \cdot \hat{\mathrm{P}} = 0$, where $\cdot$ denotes the inner product of vectors.

3. The points on the line $L_{ray}(\mathrm{P},\hat{\mathrm{P}})$ differ by multiples of P. Moreover, the plane which contains the line $L_{ray}(\mathrm{P},\hat{\mathrm{P}})$ and the origin has normal $\hat{\mathrm{P}}$.

4. Let L_{ray} be a line whose Plücker ray coordinates (p_{ij}) have been computed from the two points x and y. Let x' and y' be any two points on the line, i.e., let $x' = k_1 x + l_1 y$ and $y' = k_2 x + l_2 y$. Then the Plücker coordinates derived from x' and y' are (p'_{ij}) where $(p'_{ij}) = (k_1 l_2 - k_2 l_1) p_{ij}$

5. Let L_{ray} be a line with Plücker ray coordinates (p_{ij}). If $p_{12} \neq 0$, then L_{ray} contains $(0, p_{12}, p_{13}, p_{14})$ and $(p_{12}, 0, -p_{23}, p_{42})$. If $p_{13} \neq 0$, then L_{ray} contains $(0, p_{12}, p_{13}, p_{14})$ and $(p_{13}, p_{23}, 0, -p_{34})$. If $p_{14} \neq 0$, then L_{ray} contains $(0, p_{12}, p_{13}, p_{14})$ and $(p_{14}, -p_{42}, p_{34}, 0)$. If $p_{12} = p_{13} = p_{14} = 0$, the line is at infinity. Notice that every finite line, which is a line with one of p_{12}, p_{13}, p_{14} not equal to zero, always contains a finite point and a point at infinity.

6. Let L_{axis} be a line with Plücker axis coordinates (q_{ij}). If $q_{12} \neq 0$, then L_{axis} is on the planes $[0, q_{12}, q_{13}, q_{14}]$ and $[q_{12}, 0, -q_{23}, q_{42}]$. If $q_{13} \neq 0$, then L_{axis} is on the planes $[0, q_{12}, q_{13}, q_{14}]$ and $[q_{13}, q_{23}, 0, -q_{34}]$. If $q_{14} \neq 0$, then L_{axis} is on the planes $[0, q_{12}, q_{13}, q_{14}]$ and $[q_{14}, -q_{42}, q_{34}, 0]$. Notice that the plane $[0, q_{12}, q_{13}, q_{14}]$ contains the origin. If $q_{12} \neq 0$, the plane $[q_{12}, 0, -q_{23}, q_{42}]$ is parallel to but does not contain the x_2-axis.

7. Let $x = (x_1, x_2, x_3, x_4)$ be any point. Then x is on the line $L_{ray}=(\mathrm{P},\hat{\mathrm{P}})$ if and only if $(x_2, x_3, x_4) \times \mathrm{P} = x_1 \hat{\mathrm{P}}$ for $x_1 \neq 0$ and $(x_2, x_3, x_4) = \lambda \mathrm{P}$ for $x_1 = 0$. Note

that the condition $x_1 \neq 0$ is equivalent to requiring that the point is not at infinity.

A.2.4 More Properties

In the following, points are always denoted by the symbols r and r'. (P, $\hat{\text{P}}$), (Q, $\hat{\text{Q}}$) and (W, $\hat{\text{W}}$) are always lines. Planes are denoted a and a'. All of the Plücker coordinates are ray coordinates from now on.

Theorem A.32 The Plücker ray coordinates for the line which contains (r_1, r_2, r_3, r_4), $r_1 \neq 0$, with the direction (d_2, d_3, d_4) are (P, $\hat{\text{P}}$) where $\text{P} = (d_2, d_3, d_4)$ and $r_1\hat{\text{P}} = (r_2, r_3, r_4) \times (d_2, d_3, d_4)$. This is equivalent to $L_{ray}((r_1, r_2, r_3, r_4)(0, d_2, d_3, d_4))$.

Theorem A.33 The parametrization for the line with ray coordinates (P, $\hat{\text{P}}$) is:

$$(x_1, x_2, x_3, x_4) = (0, p_{12}, p_{13}, p_{14})t + (p_{12}, 0, -p_{23}, p_{42}) \text{ if } p_{12} \neq 0$$

$$(x_1, x_2, x_3, x_4) = (0, p_{12}, p_{13}, p_{14})t + (p_{13}, p_{23}, 0, -p_{34}) \text{ if } p_{13} \neq 0$$

$$(x_1, x_2, x_3, x_4) = (0, p_{12}, p_{13}, p_{14})t + (p_{14}, -p_{42}, p_{34}, 0) \text{ if } p_{12} \neq 0$$

Note that all of these lines pass through a finite point when $t = 0$ and through the point $(0, p_{12}, p_{13}, p_{14})$ at infinity.

Theorem A.34 Let $L^1 = (\text{P}, \hat{\text{P}})$ and $L^2 = (\text{Q}, \hat{\text{Q}})$:

1. If P=λQ and $\hat{\text{P}}$=$\lambda\hat{\text{Q}}$ for some $\lambda \neq 0$, then $L^1 = L^2$.
2. If P=λQ and $\hat{\text{P}} \neq \lambda\hat{\text{Q}}$ for some $\lambda \neq 0$, then L^1 is parallel to L^2.
3. If $\hat{\text{P}}$=$\lambda\hat{\text{Q}}$ for some $\lambda \neq 0$, then L^1 and L^2 are on the same plane. Also, the plane will have the normal $\hat{\text{P}}$ and contain the origin.
4. P$\cdot\hat{\text{Q}}$+$\hat{\text{P}}\cdot$Q=0 if and only if L^1 and L^2 intersect.
5. Assume P$\cdot\hat{\text{Q}}$+$\hat{\text{P}}\cdot$Q=0. Then P$\cdot$Q=$\|P\|\|Q\|\cos\theta$ if and only if L^1 and L^2 intersect with the angle θ. In the special case P$\cdot$Q=0, the lines are normal to each other.

Theorem A.35 The Plücker coordinates for the normal of the surface $F(x_1, x_2, x_3, x_4) = 0$ at the point (x_1, x_2, x_3, x_4) are $L_{ray}((x_1, x_2, x_3, x_4), (0, F_{x_2}, F_{x_3}, F_{x_4}))$, where F_{x_i} is the partial differential of F with respect to x_i.

A.3 Definitions and theorems

The definitions and theorems of in this section simplify the problem of finding the nearest approach pairs in some cases.

Definition 6 Let $p = (x_1, x_2, x_3, x_4)$ be a point or a vector in 3D with x_1 the homogeneous variable. If $x_1 = 0$, then p represents not only a point at infinity, but also a 3D vector (x_2, x_3, x_4).

Vector operations in homogeneous coordinates, such as inner product, cross product, $\cdots$, etc., can be derived directly from the affine definition. The operators for points or point/vector are:

- Addition of two finite points. ($x_1 \neq 0$ and $y_1 \neq 0$).

$$(x_1, x_2, x_3, x_4) + (y_1, y_2, y_3, y_4) = (x_1y_1, x_1y_2 + x_2y_1, x_1y_3 + x_3y_1, x_1y_4 + x_4y_1)$$

- Addition for a point and a vector. ($x_1 \neq 0$)

$$(x_1, x_2, x_3, x_4) + (0, y_2, y_3, y_4) = (x_1, x_2 + y_2, x_3 + y_3, x_4 + y_4)$$

- Difference of two points.

$$(x_1, x_2, x_3, x_4) - (y_1, y_2, y_3, y_4) = (0, \frac{x_1y_2 - x_2y_1}{x_1y_1}, \frac{x_1y_3 - x_3y_1}{x_1y_1}, \frac{x_1y_4 + x_4y_1}{x_1y_1})$$

  Notice the difference of two points is a vector from one point to the other.

- Scalar product for a point or a vector.

$$a(x_1, x_2, x_3, x_4) = (x_1, ax_2, ax_3, ax_4)$$

- The parametric equation for a line passing through a point $p = (x_1, x_2, x_3, x_4)$ with the direction $n = (0, n_2, n_3, n_4)$ is $p + t * n$, where $t \in \mathcal{R}$.

Definition 7 We define the following notations for CSG primitives as shown in Figure A.1:

plane $[a_1, a_2, a_3, a_4]$ represents the plane $a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 = 0$, where x_1 is the homogeneous variable. The normal vector $[0, a_2, a_3, a_4]$ is always normalized, that is $a_2 + a_3 + a_4 = 1$.

sphere $S(p, r)$ represents the sphere centered at p with radius r.

cylinder $Cy(p, axis, r)$ represents the cylinder which has the axis $p + t * axis$, $t \in \mathcal{R}$ with radius r. The axis is always normalized.

cone $Co(p, axis, len, wid)$ represents the cone which has the axis $p + t * axis, t \in \mathcal{R}$ and the angle $\alpha = \tan^{-1} \frac{wid}{len}$

Torus $T(p, axis, major, minor)$ represents the torus which has the axis $p+t*axis, t \in \mathcal{R}$ with major radius "major" and minor radius "minor". Consider the plane that contains the point p with normal $axis$. Then the circle centered at p with radius $major$ on this plane is the circle axis of the torus.

We define the point p for the primitives $S(p, r), CY(p, axis, r), Co(p, axis, len, wid)$ and $T(p, axis, major, minor)$ the reference point of the primitive surface.

Definition 8 Let q be a point on the CSG primitive surface f. The normal projection reference point q' of q is defined as follows:

- If $f = [a_1, a_2, a_3, a_4]$, then $q' = (0, a_2, a_3, a_4)$.
- If $f = S(p, r)$, then $q' = p$.
- If $f = Cy(p, axis, r)$ or $f = Co(p, axis, len, wid)$, then q' is the intersection point of the normal to f at q and the axis of f.
- If $f = T(p, axis, major, minor)$, q' is the intersection point of the normal to f at q on f and the circle axis.

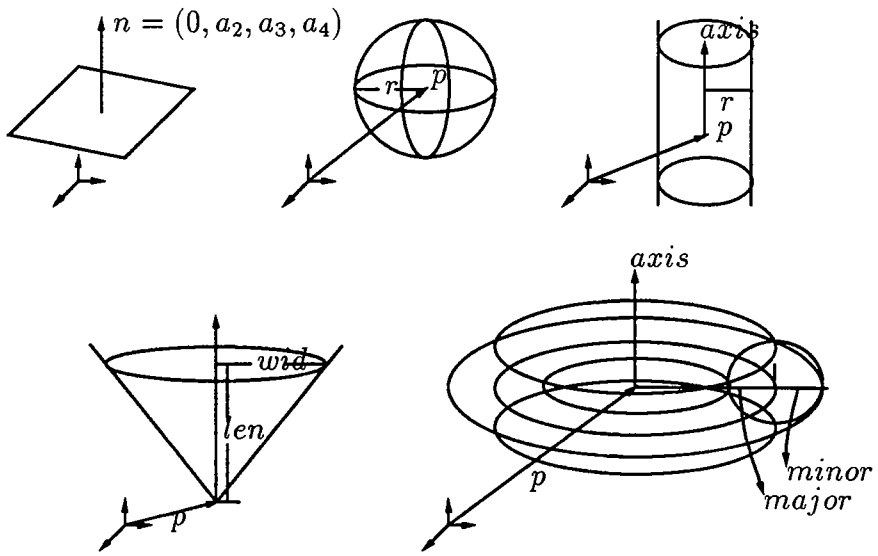


Figure A.1 Primitive surface for CSG

We write $q' = NP(f, q)$. Sometimes we write $q' = NP(f)$ if f is a plane or sphere.

Theorem A.36 If $f = Cy(p, axis, r)$ and q is on f, then

$$NP(f, q) = p + (axis \cdot (q - p)) * axis$$

If $f = Co(p, axis, len, wid)$ and q is on f, then

$$NP(f, q) = p + \frac{\|q - p\|^2}{(q - p) \cdot axis} * axis$$

If $f = T(p, axis, major, minor)$ and q is on f, then

$$NP(f, q) = p + major * (axis \times (\frac{q - p}{\|q - p\|} \times axis))$$

After expanding the equation, we can find each component of the normal projection point. For example, if $f = CY(p, axis, r)$, where $p = (1, p_x, p_y, p_z)$, $axis = (0, a_x, a_y, a_z)$ and $q = (1, q_x, q_y, q_z)$ is on f, Then,

$$NP(f, q) = (1, p_x + l * a_x, p_y + l * a_y, p_z + l * a_z)$$

where $l = a_x * (q_x - p_x) + a_y * (q_y - p_y) + a_z * (q_z - p_z)$.

Similarly, if $f = Co(p, axis, len, wen)$ where $p = (1, p_x, p_y, p_z)$, $axis = (0, a_x, a_y, a_z)$ and $q = (1, q_x, q_y, q_z)$ is on f, Then,

$$NP(f, q) = (l_1, p_x * l_1 + a_x * l_2, p_y * l_1 + a_y * l_2, p_z * l_1 + a_z * l_2)$$

where $l_1 = a_x * (q_x - p_x) + a_y * (q_y - p_y) + a_z * (q_z - p_z)$ and $l_2 = (q_x - p_x)^2 + (q_y - p_y)^2 + (q_z - p_z)^2$.

The $NP(f, q)$ for the torus case can be obtained similarly.

If we want to test whether a point q is on the surface f, we can evaluate the implicit equation of f. When f is in general position, then f could be obtained by transforming the implicit equation of f, in standard position, with help of a coordinate transformation. Numerically, this is not attractive, because floating-point errors subtly change coefficients in the resulting equation so that, for example, a circular cylinder would not be exactly circular, but probably somewhat elliptic. A method

that is numerically more stable is to transform the geometric characteristics of f and to construct from them the transformed implicit equation. This is also advocated by Goldman and Miller [24, 25]. In particular, we proceed as follows.

Let q be a point on the surface, $q' = NF(q, f)$ the normal projection reference point of q. Then

- If $f = [a_1, a_2, a_3, a_4]$, we need to check whether $\sum_{i=1}^{4} a_i x_i = 0$.
- If $f = S(p, r)$, we need to check whether $\|q - p\| = r$.
- If $f = Cy(p, axis, r)$, we need to check whether $\|q - NP(f, q)\| = r$.
- If $f = Co(p, axis, len, wid)$, we need to check whether $axis \cdot (q - p) = \|q - p\| * \frac{len}{\sqrt{len^2 + wid^2}}$.
- If $f = T(p, axis, major, minor)$, we need to check whether $\|q - NP(f, q)\| = minor$.

Sometimes we need the implicit equation for all primitive surfaces, especially when we generate a system of equation. The implicit surface is easy to generate by treating the point q as unknown. For example, let $p = (1, p_x, p_y, p_z)$ and $q = (1, x, y, z)$. The implicit equation for the sphere S(p,r) is $(q - p) \cdot (q - p) - r^2 = 0$. More precisely, the implicit equation is:

$$(x - p_x)^2 + (y - p_y)^2 + (z - p_z)^2 - r^2 = 0$$

Similar techniques can be used for cylinder, cone, and torus. Assume the cylinder $Cy(p, axis, r)$ has $p = (1, p_x, p_y, p_z)$ and $axis = (0, a_x, a_y, az)$, then the implicit equation for the cylinder is:

$$(x - p_x - l * a_x)^2 + (y - p_y - l * a_r)^2 + (z - p_z - l * a_z)^2 - r^2 = 0$$

where $l = a_x * (x - p_x) + a_y * (y - p_y) + a_z * (z - p_z)$ The implicit equations for cone and torus can be derived similarly. The implicit equations so derived have degree 2,2,2,4 for sphere, cylinder, cone and torus respectively.

Let $\Pi(n,p)$ be the plane through the point $p=(x_1,x_2,x_3,x_4)$, $x_1 \neq 0$ with normal $n=(n_1,n_2,n_3,n_4)$. Then $\Pi(n,p)=[-(n_2x_2+n_3x_3+n_4x_4),x_1n_2,x_1n_3,x_1n_4]$. The tangent plane for the surface $f(x_1,x_2,x_3,x_4)=0$ is $[-x_1(x_2f_{x_2}+x_3f_{x_3}+x_4f_{x_4}),\ f_{x_2},\ f_{x_3},\ f_{x_4}]$, where f_{x_i} is the partial differential of f with respect to x_i.

Let $(l,\hat{l})=L_{ray}(p_1,p_2)=L_{axis}(\Pi_1,\Pi_2)$ be a line in 3D. This line contains the points p_1, p_2 and is the intersections of the planes Π_1 and Π_2. Notice that the direction of the line $(l,\hat{l})$ will be p_2-p_1 or $normal(\Pi_1)\times normal(\Pi_2)$.

Let F and G be primitive surfaces, or a curve that is the intersection of primitive surfaces, or a vertex. The nearest approach pairs are the set of points on F and on G, call them F' and G' such that for every point of F', we can find a point in G' so that the distance between these two points is equal to the minimum distance between F and G. If F and G are two coplanar and concentric circles of different radii, then $F'=F$ and $G'=G$. So, in this case the nearest approach pairs are two circles. If F' and G' are single points, then the line through them is the nearest approach line.

Theorem A.37 Let F and G be primitive surfaces or a curve that is the intersection of two primitive surfaces. Assume the nearest approach line $(l,\hat{l})$ of F and G intersects F at q_1 and G at q_2. Then:

1. If F is a primitive surface, then $(l,\hat{l})$ is perpendicular to F and passes through $NP(F,q_1)$.
2. If F is the intersection of the primitive surfaces f_1 and f_2, let Π_1 be the plane which contains the points $NP(f_1,q_1), NP(f_2,q_1)$ and q_1. Then the line $(l,\hat{l})$ is on the plane Π_1.

The proof for the first case is trivial. It is easy to see the second case is true, because the three vectors, which are l, the normal of f_1 at q_1, and the normal of f_2 at q_1, are perpendicular to the tangent of the curve, and the normals of f_1 at q_1 and of f_2 at q_1 pass through the points $NP(f_1,q_1)$ and $NP(f_2,q_1)$, respectively.

Notice that if F and G are the curves generated by the intersection of two primitive surfaces, say f_1,f_2 and g_1,g_2 respectively, and if Π_1 and Π_2 are two planes that contain

three points, $q_1, NP(f_1, q_1), NP(f_2, q_1)$ and $q_2, NP(g_1, q_2), NP(g_2, q_2)$, respectively, then, $(l, \hat{l}) = L_{axis}(\Pi_1, \Pi_2)$.

Lemma A.38 Let Intersect$(obj1, obj2)$ denote the intersection of $obj1$ and $obj2$.

1. If $obj1 = \Pi_1$ and $obj2 = \Pi_2$ are planes, then Intersect$(\Pi_1, \Pi_2) = L_{axis}(\Pi_1, \Pi_2)$.
2. If $obj1 = \Pi$ is a plane and $obj2 = (l, \hat{l})$ is a line, the intersection point is easy to find after parametrizing the line.
3. Let $obj1 = (l_1, \hat{l_1})$ and $obj2 = (l_2, \hat{l_2})$. If these two lines intersect, then the equation $l_1 \cdot \hat{l_2} + l_2 \cdot \hat{l_1} = 0$ is true. In this case, we can find the intersection point of the line $(l_2, \hat{l_2})$ and an arbitrary plane which contains $(l_1, \hat{l_1})$ but not $(l_2, \hat{l_2})$. This arbitrary plane can easily found, because it is easy to transfer the Plücker ray coordinates of $(l_1, \hat{l_1})$ to Plücker axis coordinates and from the Plücker axis coordinate, we can easily find two planes which intersect at $(l_1, \hat{l_1})$. We select either one of the planes whose normal is not perpendicular to l_2.

Intersect$(obj1, obj2)$ also works for finding the intersection point for a line and a CSG primitive surface. After parametrizing the line, one, two or four intersection points will be found depending on the surface.

Definition 9 The operator Move(obj, n, d) moves the object "obj" along the direction n by the distance d.

Definition 10 The operator Rotate$(obj, p, axis, alpha)$ rotates the object "obj" around the line passing through the point p and $p + axis$ clockwise, looking from p to the point $p + axis$, by an angle of alpha degrees.

Definition 11 Circle$(p, axis, r)$ represents a circle in 3D. The circle is centered at p with radius r. The circle is the intersection of the plane $\Pi(axis, p)$ and the cylinder $Cy(p, axis, r)$.

A.4 Closest approach pairs between CSG primitive surfaces

In this section, we assume that the two surfaces f and g do not intersect. If they intersect, the closest approach pairs will be the intersection points of the two surfaces. Points and lines are considered as spheres and cylinders with radius zero.

A.4.1 Solving closest approach pair between two primitive surface

The nearest approach pairs for two CSG surfaces can be two points, two half lines, two lines, two circles, or even two planes or two tori. We will find the nearest approach pair algebraically, that is, by solving a system of equations, when the nearest approach pairs are finitely many pairs of points. Otherwise, the solution will be found by geometric constructions.

A.4.1.1 Sphere vs. sphere or sphere vs. plane

It is trivial to find closest approach pairs for this case.

A.4.1.2 Sphere vs. cylinder

Assume $f = S(p_1, r_1)$ is a sphere and $g = Cy(p_2, axis, r_2)$ is a cylinder. Let the nearest approach line $(l, \hat{l})$ pass through the point $q' = p_2 + t * axis$ on the axis of the cylinder. From the fact that $(p_1 - q')$ must be perpendicular to $axis$, we derive $t = (p_1 - p_2) \cdot axis$. After we find the line passing though p_1 and q', the nearest approach pairs are easy to find.

A.4.1.3 Sphere vs. cone

Assume $f = S(p_1, r)$ and $g = Co(p_2, axis, len, wid)$. Assume also that $\alpha = \tan^{-1}(wid/len)$. There are three cases for this problem:

1. p_1 is on $p_2 + t * axis, t > 0$. In this case, the nearest approach pairs are two circles.

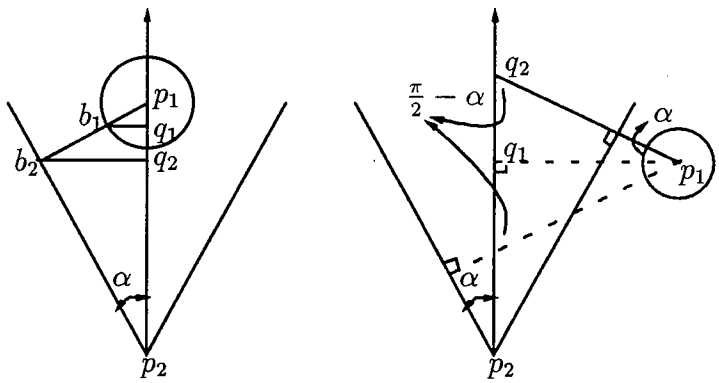


Figure A.2 Sphere vs. cone

2. p_1 is inside or on the cone $Co(p_2, -axis, wid, len)$. In this case, the nearest approach pairs are two points. One of the points is the apex of the cone.

3. p_1 is not on the axis of the cone and not in the interior or on the cone $Co(p_2, -axis, wid, len)$. In this case, the nearest approach pairs are two points. None of the points is the apex of the cone.

The closest approach pairs for the first case are two circles. Consider a plane that contains the axis of the cone. The plane intersects the cone in two lines, as shown in figure A.2(a). Then

$$\sin\alpha = \frac{\|p_1 - q_1\|}{r} = \frac{\|p_1 - b_2\|}{\|p_1 - p_2\|} = \frac{\|b_2 - q_2\|}{\|p_2 - b_2\|}$$

$$\cos\alpha = \frac{\|b_1 - q_1\|}{r} = \frac{\|p_2 - b_2\|}{\|p_1 - p_2\|} = \frac{\|p_2 - q_2\|}{\|p_2 - b_2\|}$$

we derive

$$\|p_1 - q_1\| = r\sin\alpha$$

$$\|q_1 - b_1\| = r\cos\alpha$$

$$\|p_2 - q_2\| = \|b_2 - p_2\|\cos\alpha = \|p_2 - p_1\|\cos^2\alpha$$

$$\|b_2 - q_2\| = \|b_2 - p_2\|\sin\alpha = \|p_2 - p_1\|\sin\alpha\cos\alpha$$

Let

$$n = \frac{(p_1 - p_2)}{\|p_1 - p_2\|}$$

$$q_1 = p_1 - \|p_1 - q_1\| * n$$

$$q_2 = p_2 + \|p_2 - q_2\| * n$$

$$r_1 = \|b_1 - q_1\|$$

$$r_2 = \|b_2 - q_2\|$$

The circles $Circle(q_1, axis, r_1)$ and $Circle(q_2, axis, r_2)$ are closest approach pairs for the sphere f and the cone g in the first case.

The closest approach pairs for the second case are the same as "sphere vs. sphere", where one of the spheres is the apex of the cone.

For the third case, let $q_2 = p_2 + t * axis$ be the point on the nearest approach line of f and g, and on the axis of the cone. From the fact that the nearest approach line will have the angle $\frac{\pi}{2} - \alpha$ with the axis of the cone, we have $(q_2 - p_1) \cdot axis = \|q - p_1\| \cos(\frac{\pi}{2} - \alpha)$. We can find t by solving this equation. Notice that the equation has degree 2, so we have two solutions in this case, as shown in Figure A.2(b). We are only interest in the larger t in this case.

There is also an easy geometric solution. Consider the figure A.2(b). Note that the projection point q_1 of p_1 onto the line $p_2 + t * axis$, can be found easily. From the fact

$$\tan\alpha = \frac{len}{wid} = \frac{\|q_1 - q_2\|}{\|p_1 - q_1\|}$$

$\|q_1 - q_2\|$ can also be found, and then the point q_2 can be calculated easily by $q_2 = q_1 + \|q_1 - q_2\| * axis$.

A.4.1.4 Sphere vs. torus

Assume $f = S(p_1, r)$ and $g = T(p_2, axis, major, minor)$. There are five cases for this problem:

1. The center of the sphere p_1 is on the axis of the torus. That is, the closest approach pairs will be two circles. Notice that a point can also be represented by a circle with radius zero.

2. The center of the sphere is on the circle axis of the torus. In this case, the nearest approach pairs are two circles.

3. The center of the sphere is not on the circle axis of the torus and is on the cylinder $Cy(p_2, axis, major)$.

4. The center of the sphere is not on the axis of the torus and is in the interior of the cylinder $Cy(p_2, axis, major)$.

5. The center of the sphere is not on the axis of the torus and is in the exterior of the cylinder $Cy(p_2, axis, major)$.

Consider Figure A.3(a), and assume that the angle between $(q - p_1)$ and $p_2 - p_1$ is α. Then $\alpha = \tan^{-1} \frac{major}{\|p_1 - p_2\|}$. From the fact that

$$\sin\alpha = \frac{\|q_1 - b_1\|}{r} = \frac{\|q_2 - b_2\|}{\|p_1 - b_2\|} = \frac{major}{\|p_1 - q\|}$$

$$\cos\alpha = \frac{\|p_1 - q_1\|}{r} = \frac{\|p_1 - q_2\|}{\|p_1 - b_2\|} = \frac{\|p_1 - p_2\|}{\|p_1 - q\|} = \frac{\|p_2 - q_2}{minor}$$

it can be derived

$$\|p_1 - q_1\| = r * \cos\alpha$$

$$\|q_1 - b_1\| = r * \sin\alpha$$

$$\|p_2 - q_2\| = minor * \cos\alpha$$

$$\|q_2 - b_2\| = (\|p_1 - p_2\| - \|q_2 - p_2\|)\tan\alpha = (\|p_1 - p_2\| - \|q_2 - p_2\|) * \frac{major}{\|p_1 - p_2\|}$$

Let

$$n = \frac{(p_2 - p_1)}{\|p_2 - p_1\|}$$

$$r_1 = \|q_1 - b_1\|$$

$$r_2 = \|q_2 - p_2\|$$

$$q_1 = p_1 + \|p_1 - q_1\| * n$$

$$q_2 = p_2 - \|q_2 - p_2\| * n$$

Then, these two circles $Circle(q_1, axis, r_1)$ and $Circle(q_2, axis, r_2)$ are the nearest approach pairs for this case. Notice this case does not cover the case that $p_1 = p_2$.

The case the $p_1 = p_2$ is trivial. If $r > major + minor$, then the nearest approach pairs are $Circle(p_1, axis, r)$ and $Circle(p_1, axis, major + minor)$. If $r < major - minor$, then the nearest approach pairs are $Circle(p_1, axis, r)$ and $Circle(p_1, axis, major - minor)$. If $major - minor \leq r \leq major + minor$, then f and g intersect each other.

For the second case, the nearest approach pairs are two circles, namely $Circle(p_1, n, r)$ and $Circle(p_1, n, minor)$ where $n = axis \times (p_1 - p_2)$.

For the third case, the nearest approach line is the line passing through p_1 with direction $axis$. We can solve this case geometrically by find the line $(l, \hat{l}) = L_{ray}(p_1, p_1 + axis)$. Notice that this line passes through p_1, the circle axis of the torus, and intersects the axis of the torus at infinity.

For the last two cases, let l_x and l_y be the distance from p_1 to the axis of torus and to the plane $\Pi(axis, p_2)$ respectively. Assume the nearest approach line passes a point q' on the axis of the torus and the distance from q' to p_2 is l. If p_1 is on the plane $\Pi(axis, p_2)$, we know that the nearest approach line will passing through p_1 and p_2. Without loss of the generality, we assume the point p_1 is on the positive side of the plane $\Pi(axis, p_2)$, that is, $axis \cdot (p_1 - p_2) > 0$. With the picture in A.3(b)(c) and the similar triangular properties, the value l can easily be found for the last two cases. They are $l = major * \frac{l_y}{l_x}$ for the fourth case and $l = \frac{major * l_y}{l_x - major}$ for the last case.

The nearest approach line passes the point $p_2 + l * axis$ for the fourth case and $p_2 - l * axis$ for the last case. Notice that the nearest approach line for these two cases passes through p_1, the "circle axis" and the axis of the torus.

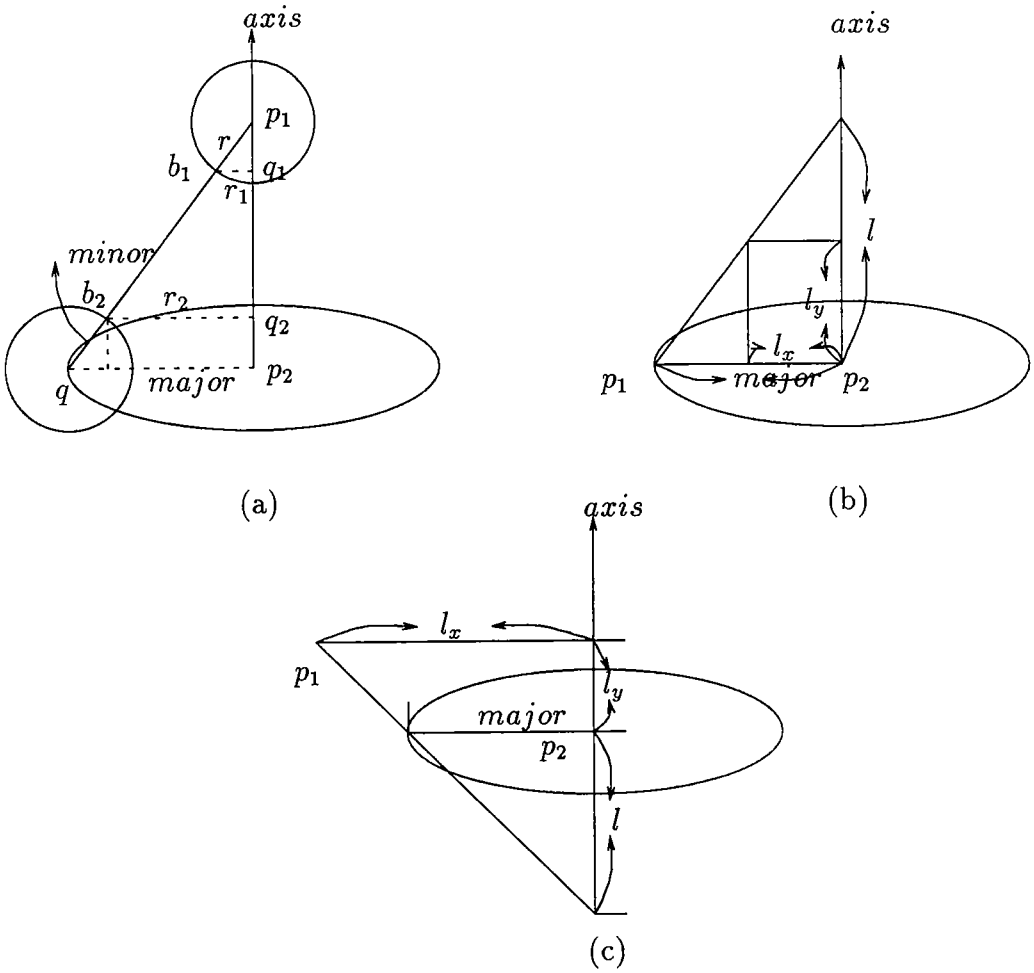


Figure A.3 Sphere vs. Torus

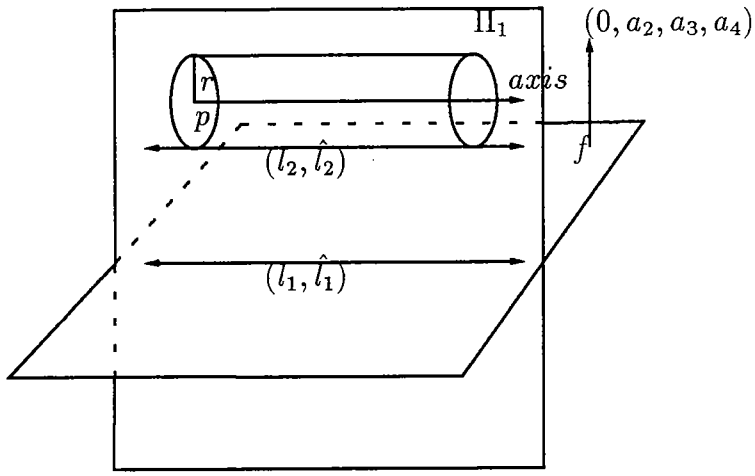


Figure A.4 Plane vs. cylinder

A.4.1.5 Plane vs. plane

Assume that $f = [a_1, a_2, a_3, a_4]$ and $g = [a_1', a_2, a_3, a_4]$, the closest approach pairs are f and g itself. The middle plane for f and g is $[\frac{a_1+a_1'}{2}, a_2, a_3, a_4]$.

Notice that the normals of planes are always normalized. Notice also that f and g do not intersect if and only if they have the same normal.

A.4.1.6 Plane vs. cylinder

Assume $f = [a_1, a_2, a_3, a_4]$ and $g = Cy(p, axis, r)$. The only case that f and g do not intersect is when the axis of the cylinder is perpendicular to the normal of the plane. The nearest approach pairs for this case are two lines. We can find these two lines by(See figure A.4):

1. Find the plane that contains the axis of the cylinder and is perpendicular to the plane f by $\Pi_1 = \Pi((0, a_2, a_3, a_4) \times axis, p)$.

2. Assume the normal of f points toward the axis of the cylinder. Let $(l_1, \hat{l}_1) = L_{axis}(\Pi_1, f)$ and $(l_2, \hat{l}_2) = Move(axis, -(0, a_2, a_3, a_4), r)$. Then, $(l_1, \hat{l}_1)$ and $(l_2, \hat{l}_2)$ are the nearest approach pairs of this case.

A.4.1.7 Plane vs. cone

Assume that $f = [a_1, a_2, a_3, a_4]$ and $g = Co(p, axis, len, wid)$, let $\alpha = \tan^{-1} \frac{wid}{len}$. There are two cases for this problem:

1. The closest approach pairs are two points. In this case, the closest approach line will contain the apex of the cone.

2. The closest approach pairs are two half lines. In this case, the angle between the axis of the cone and the normal of the plane is $\frac{\pi}{2} - \alpha$.

The first case is the same as the "sphere vs. plane" case. The nearest approach pairs for the second case can be found as follows (See figure A.5):

1. Find the plane that contains the axis of the cone and is perpendicular to the plane f by $\Pi_1 = \Pi(axis \times (0, a_2, a_3, a_4), p)$.

2. Find the intersection line for the plane f and Π_1 by $(l_1, \hat{l_1}) = L_{axis}(f, \Pi_1)$.

3. Find the plane parallel to f and tangent to g by $\Pi_2 = \Pi(n \times l_1, p)$, where $n = axis \times (0, a_2, a_3, a_4)$.

4. Find corresponding line for $(l_1, \hat{l_1})$ by $(l_2, \hat{l_2}) = L_{axis}(\Pi_1, \Pi_2)$.

The closest approach pairs are contained in these two lines $(l_1, \hat{l_1})$ and $(l_2, \hat{l_2})$. The end point of the line $(l_2, \hat{l_2})$ is p, the apex of the cone. The end point on the line $(l_1, \hat{l_1})$ is the projection point of p on the plane f. Note that the plane normals of the points on the half line of $(l_1, \hat{l_1})$ intersect the half line of $(l_2, \hat{l_2})$ and the cone axis.

A.4.1.8 Plane vs. torus

Assume $f = [a_1, a_2, a_3, a_4]$ and $g = T(p_2, axis, major, minor)$. There are two cases for this problem:

1. The normal of f is parallel to the axis of the torus. That is, the closest approach pairs will be two circles.

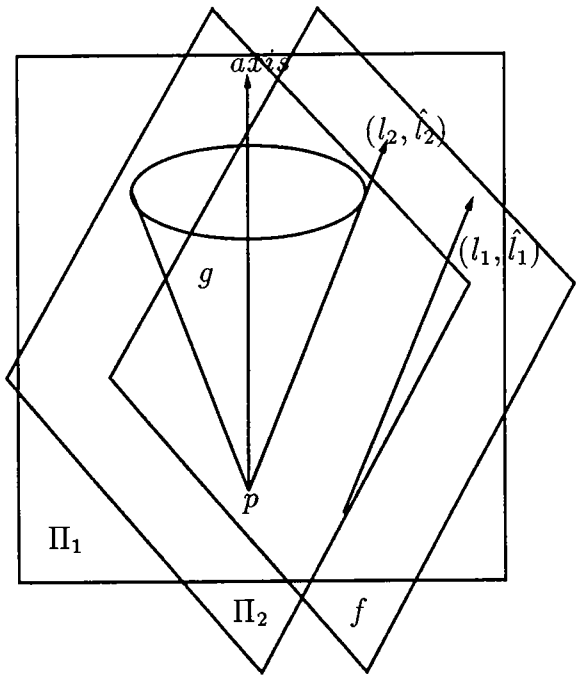


Figure A.5 Plane vs. cone

2. The normal of f is not parallel to the axis of the torus.

The closest approach pair for the first case can be found by:

1. Finding the distance l from p to f.

2. Decide whether the plane is above or below the torus. If the axis of the torus is parametrized by $p_2 + t * axis$, and the intersection point of the axis of the torus with the plane is a point with positive parameter, then the plane is above the torus. Otherwise the plane is below the torus. Let $sign = 1$ if f is above the torus. Or, $sign = -1$ if f is below the torus.

3. The nearest approach pairs are two circles. They are $Circle(axis, p + sign * minor * axis, major)$ and $Circle(axis, p + sign * l * axis, major)$.

For the second case. Let $n = (a_2, a_3, a_4)$ be the normal of the plane. Without loss of generality, we assume the point p_2 is on the positive side of the plane. That is, the angle between $p_2 - p$, where p is any point on the plane, and the plane is less than $\frac{\pi}{2}$. Let the nearest approach line pass the point q, which is a point on the axis of the torus, and let l be the distance from p to p_2, as shown in Figure A.6(a)(b). From the properties that $axis \cdot n = \cos\alpha$ and $\cot\alpha = \frac{l}{major}$, we can find $l = major * \frac{\sqrt{1-|axis \cdot n|}}{|axis \cdot n|}$. If $axis \cdot n > 0$, then $q = p_2 + l * axis$. If $axis \cdot n = 0$, then $q = p_2$. If $axis \cdot n < 0$, then $q = p_2 - l * axis$.

The six remaining cases consider surface pairs (f, g) where each surface has an axis of symmetry. For point-pair solutions, the closest approach line must interest both axis. In the following, we assume that the intersection points are

$$q_1 = p_1 + s * axis_1$$

where $axis_1$ is the axis of f and p_1 is the reference point of f, and

$$q_2 = p_2 + t * axis_2$$

where $axis_2$ is the axis of g and p_2 is the reference point of g. We also assume

$$v = q_2 - q_1 = (p_2 - p_1) + (t * axis_2 - s * axis_1)$$

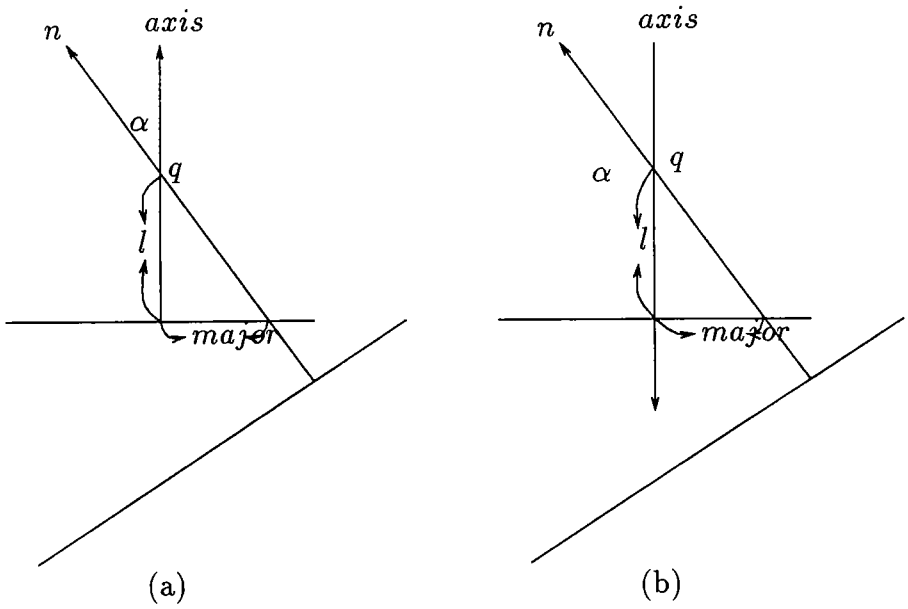


Figure A.6 Plane vs. torus

A.4.1.9 Cylinder vs. cylinder

Assume $f = Cy(p_1, axis_1, r_1)$ and $g = Cy(p_2, axis_2, r_2)$, there are three cases for this problem:

1. f and g have the same axis. In this case, the nearest approach pairs are f and g. See the figure A.7(a).

2. f and g have parallel axes. In this case, the nearest approach pairs for f and g will be two lines. See the figure A.7(b)(c).

3. f and g have skew axes. In this case, the nearest approach pairs for f and g will be two points.

The first case is simple. The skeleton for these two cylinders will be $Cy(p_1, axis_1, \frac{r_1+r_2}{2})$

The closest approach pair for the second case will be two lines. They can be found by: (Notice $axis_1 = axis_2$ in this case)

1. Find the plane that contains the axis of f and the axis of g by $\Pi_1 = \Pi(n, p_1)$, where $n = (p_2 - p_1) \times axis_1$.

2. Find a plane $\Pi_2 = \Pi(axis, p_1)$ which has the normal $axis_1$.

3. Let $(l, \hat{l}) = P_{axis}(\Pi_1, \Pi_2)$. Parametrize the line $(l, \hat{l})$, and find the two intersection points of $(l, \hat{l})$ and f, and the two intersection points of $(l, \hat{l})$ and g.

4. Let the nearest approach points for f and g on the line $(l, \hat{l})$ be q_1 and q_2 respectively, then the two lines $(l_1, \hat{l_1}) = L_{ray}(q_1, q_1 + axis_1)$ and $(l_2, \hat{l_2}) = L_{ray}(q_2, q_2 + axis_2)$ are the closest approach pairs of f and g in this case.

The closest approach pairs for the third case will be two points. From the assumption before and the fact that $axis_1 \cdot v = 0$ and $axis_2 \cdot v = 0$, we derive:

$$(axis_1 \cdot axis_2)s - t = axis_1 \cdot (p_1 - p_2)$$

$$s - (axis_1 \cdot axis_2)t = axis_2 \cdot (p_1 - p_2)$$

After solving these two equations, we can find the nearest approach pair easily.

It is easy to solve this problem geometrically by (See the figure A.7(d)):

1. Find a vector perpendicular to $axis_1$ and $axis_2$ by $n = axis_1 \times axis_2$.
2. Find the plane that contains p_1 and is spanned by the vectors $axis_1$ and n, so $\Pi_1 = \Pi(n \times axis_1, p_1)$
3. Find the plane that contains p_2 and is spanned by the two vectors $axis_2$ and n, so $\Pi_2 = \Pi(n \times axis_2, p_2)$
4. The line $(l, \hat{l}) = L_{axis}(\Pi_1, \Pi_2)$ is now the closest approach line.

A.4.1.10 Cylinder vs. cone

Assume $f = Cy(p_1, axis_1, r)$ is a cylinder and $g = Co(p_2, axis_2, len, wid)$ is a cone. There are three cases for this problem:

1. The closest approach pairs are two half lines. In this situation, the axes of f and g intersect with an angle $\alpha = \tan^{-1} \frac{wid}{len}$
2. The closest approach pairs are two points. One of the point is the apex of the cone.
3. The closest approach pairs are two points. Neither point is the apex of the cone. In this situation, the axes for the cylinder and the cone are skew.

The lines that contain the two half lines for the first case can be found by (See figure A.8):

1. Find the line $(l_1, \hat{l_1})$ passing through p_2 parallel to $axis_1$.
   $(l_1, \hat{l_1}) = L_{ray}(p_2, p_2 + axis_1)$.
2. Find the line $(l_2, \hat{l_2})$ on the cylinder closest to the cone by:
   $(l_2, \hat{l_2}) = Move(L_{ray}(p_1, p_1 + axis_1), n, r)$ where $n = a_1 \times ((p_2 - p_1) \times a_1)$.

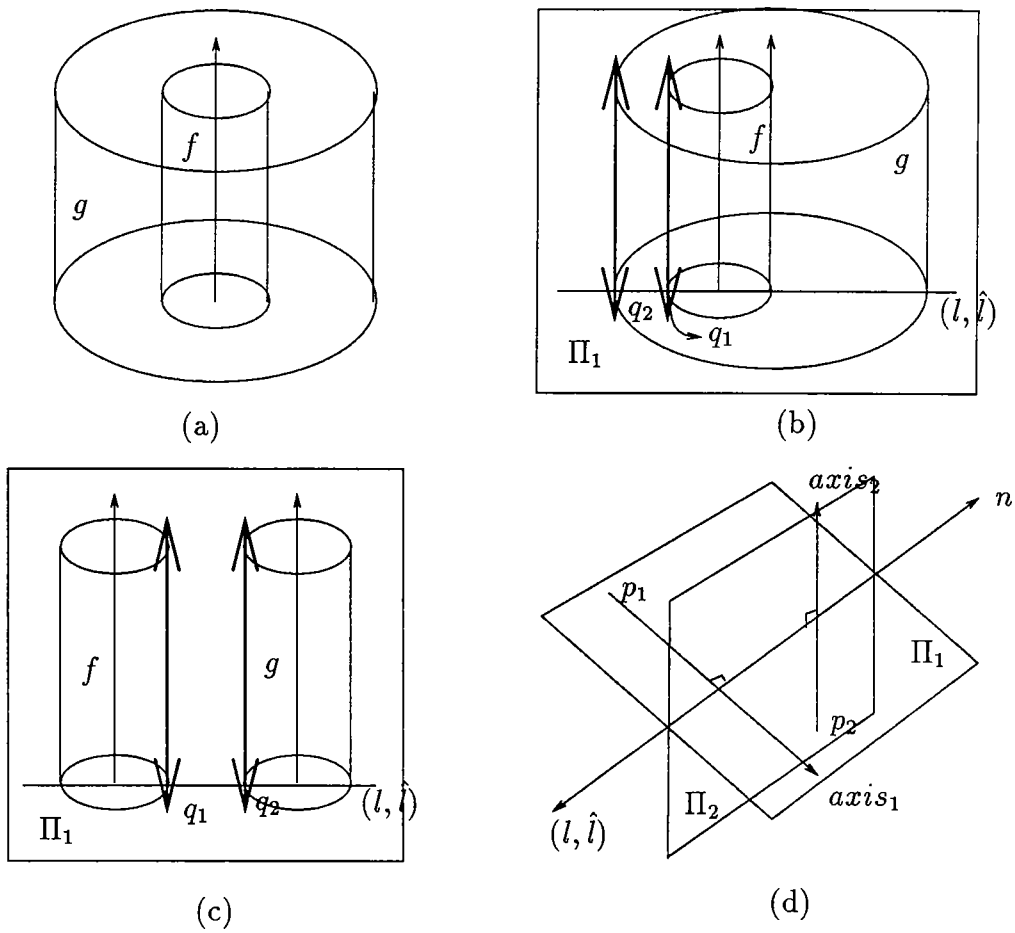


Figure A.7 Cylinder vs. cylinder

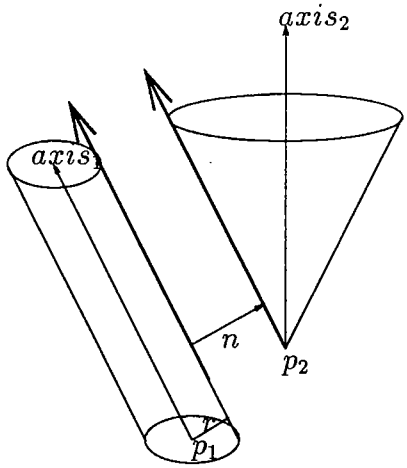


Figure A.8 Cylinder vs. cone

3. The two lines $(l_1, \hat{l_1})$ and $(l_2, \hat{l_2})$ contain those two half lines we want to find. The end points of the half lines are easy to find.

The second case is the same as "sphere vs. cone".

For the third case, from the assumptions we state before, and the fact that

$$axis_1 \cdot v = 0$$

$$axis_2 \cdot v = \|v\| \cos(\frac{\pi}{2} - \alpha)$$

we need to solve two equations, one with degree one and the other with degree two, for this problem.

A.4.1.11 Cylinder vs. torus

Assume that $f = Cy(p_1, axis_1, r)$ is a cylinder and $g = T(p_2, axis_2, major, minor)$ is a torus. There are three cases for this problem.

1. The two axes coincide. In this case, the nearest approach pair is two circles. See figure A.9(a)(b).

2. The two axes are perpendicular to each other, and the minimum distance d between the two axes d is less than or equal to $major$. In this case, the nearest

approach pair are two points if $d = major$, or two pairs of points if $d < major$. See Figure A.9(c).

3. The case not included above will have one or two pairs of points as its nearest approach pairs. Moreover, the nearest approach line passes through the axes of f and g, and the circle axis of the torus g.

The first case is simple. If $r > major + minor$, then $Circle(p_2, axis, r)$ and $Circle(p_2, axis, major + minor)$ are the nearest approach pair in this case. If $r < major - minor$, then $Circle(p_2, axis, r)$ and $Circle(p_2, axis, major - minor)$ are the nearest approach pair in this case. Notice that if $major - minor \leq r \leq major + minor$, f and g will intersect, which is excluded here.

The second case can be solved by (See figure A.9(c)):

1. Find the plane containing the circle axis of the torus.
   $\Pi_1 = \Pi(axis_2, p_2)$.

2. Find the projection line of the axis of the cylinder onto Π_1.
   $(l, \hat{l}) = P_{axis}(\Pi_1, \Pi_2)$ where $\Pi_2 = \Pi(axis_1 \times axis_2, p_1)$

3. After parametrizing the line $(l, \hat{l})$, it is easy to find the intersection points of $(l, \hat{l})$ and the circle axis of the torus. Let these two intersection points be q_1 and q_2. Notice that if $d = major$, then $q_1 = q_2$.

4. The nearest approach line will be

$$(l_1, \hat{l_1}) = P_{ray}(q_1, q_1 + axis_2)$$

$$(l_2, \hat{l_2}) = P_{ray}(q_2, q_2 + axis_2)$$

Notice that the nearest approach line intersects the circle axis but not the axis of the torus in this case.

For the third case, we solve two equations with two variables s and t. These two equations are degree 1 and 4 as shown below.

$$\begin{cases} axis_1 \cdot v = 0 \\ axis_2 \cdot v = \frac{t}{\sqrt{major^2+t^2}} * \|v\| \end{cases}$$

where $v = (p_2 - p_1) + (t * axis_2 - s * axis_1)$.

A.4.1.12 Cone vs. cone

Let $f = Co(p_1, axis_1, len_1, wid_1)$ and $g = Co(p_2, axis_2, len_2, wid_2)$. Assume $\alpha_1 = \tan^{-1}(wid_1/len_1)$ and $\alpha_2 = \tan^{-1}(wid_2/len_2)$. Without loss of generality, we further assume $\alpha_2 \geq \alpha_1$. There are six cases in this problem:

1. The nearest approach line passes through at least one of the vertices of these two cones.

2. The two axes coincide and the angle of the two cones are equal. In this case, the nearest approach pairs are a cone and a trimmed cone, as shown in figure A.10(a).

3. The two axes are in the same plane and the angle between $axis_1$ and $axis_2$ is equal to $\alpha_1 + \alpha_2$, as shown in figure A.10(b). In this case, the nearest approach pairs are two half lines.

4. The two axes are in the same plane and the angle between $axis_1$ and $axis_2$ is equal to $\alpha_2 - \alpha_1$. Furthermore, if $\alpha_1 \neq \alpha_2$, the two axes intersect at the exterior of the two cones, as shown in Figure A.10(c).

5. The two axes are in the same plane and the angle between $axis_1$ and $axis_2$ is equal to $\pi - (\alpha_2 - \alpha_1)$, as shown in figure A.10(d). Furthermore, the footpoint of p_1 on g is not the apex of g. And, if $\alpha_1 \neq \alpha_2$, then the two axes intersect in the interior of the cone f. In this case, the nearest approach pairs are two line segments.

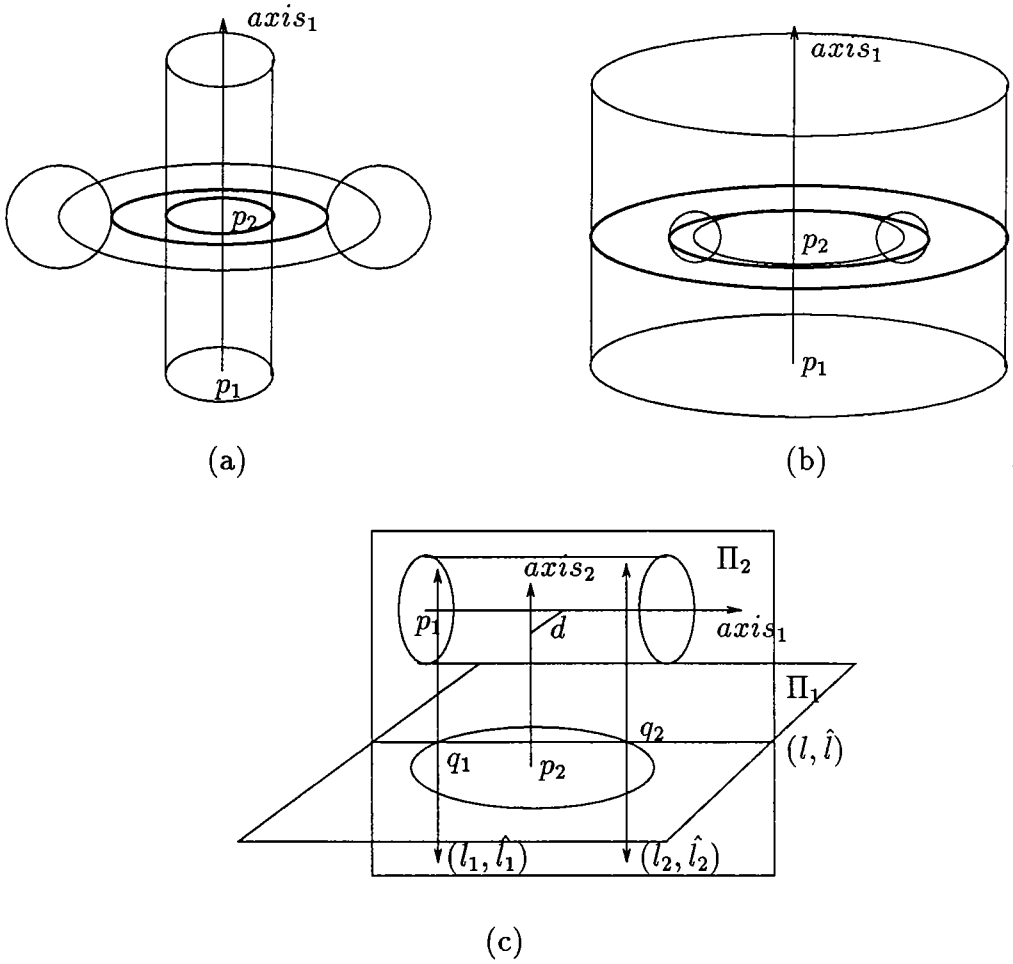


Figure A.9 Cylinder vs. torus

6. The nearest approach pairs are two points. None of the points are the vertices of either cone.

The first case is same as the sphere/sphere or the sphere/cone case.

For the second case, assume f is contained in the interior of g, as shown in Figure A.10(a). From the picture, we know

$$\tan\alpha = \frac{wid}{len} = \frac{\|q - p_1\|}{r} = \frac{r}{\|q - p_2\|}$$

From the fact that

$$\|p_1 - p_2\| = \|q - p_1\| + \|q - p_2\|$$

we can find r. Once we find r, the point q is easy to find. The cone f, and the cone g above and on the plane $\Pi(axis_1, q)$ are the nearest approach pairs in this case.

The third case can be solved by:

1. Find the plane that contains the two axes, call it Π_1. Let Π_1 has normal n_1. Then
   $n_1 = axis_1 \times axis_2$.
   $\Pi_1 = \Pi(n_1, p_1)$.

2. Rotate the $axis_1$ clockwise about n_1 by α_1 degree.
   $n_2 = Rotate(axis_1, p_1, n_1, \alpha_1)$.

3. Rotate the $axis_2$ counterclockwise about n_1 by α_2 degree.
   $n_3 = Rotate(axis_2, p_2, n_1, \alpha_2)$.

4. These two lines $(l_1, \hat{l_1}) = L_{ray}(p_1, p_1 + n_2)$ and $(l_2, \hat{l_2}) = L_{ray}(p_2, p_2 + n_3)$ contain the nearest approach pairs in this case. The end points of the half lines are easy to find.

The fourth case is similar to the third case. Notice that if $\alpha_1 \neq \alpha_2$ and the two axes intersect at the exterior of the two cones, as shown in Figure A.10(e), the situation is the same as in sphere/cone case.

The fifth case is also similar to the third case. Notice that if the footpoint of p_1 on g is the apex of g, this situation is the same as the sphere/sphere case. If $\alpha_1 \neq \alpha_2$ and the two axes intersect in the interior of the cone f, as shown in Figure A.10(f), this situation is the same as the sphere/cone case.

The last case requires solving two equations of degree 2 and 1.

$$\begin{cases} axis_1 \cdot v = \|v\| * \sin\alpha_1 \\ (axis_1 \cdot v) * \sin\alpha_2 = (axis_2 \cdot v) * \sin\alpha_1 \end{cases}$$

where $v = (p_2 - p_1) + (t * axis_2 - s * axis_1)$.

A.4.1.13 Cone vs. torus

Assume $f = Co(p_1, axis_1, len, wid)$ where $\alpha = \tan^{-1}(wid/len)$ and g=$T(p_2$,$axis_2$,$major$,$minor)$. There are three cases for this problem:

1. The two axes of f and g coincide. In this case, the nearest approach pairs are two circles. It is possible that one of the circles has radius zero.

2. The nearest approach pairs are two points. One of the points is the apex of the cone.

3. The nearest approach pairs are two points. Neither point is the apex of cone.

The first case is trivial and is illustrated in figure A.11(a)(b)(c). Using techniques analogous to the sphere/torus case, where the center of the sphere is on the axis of the torus, we can find q_1, q_2, r_1 and r_2. Then, the two circles $Circle(q_1, axis_1, r_1)$ and $Circle(q_2, axis_2, r_2)$ are the nearest approach pair in this case.

The second case is the same as sphere/torus case.

The third case requires solving two equations of degree 2 and 4 because:

$$\begin{cases} axis_1 \cdot v = \|v\| * \sin\alpha \\ axis_2 \cdot v = \|v\| * \frac{t}{\sqrt{major^2 + t^2}} \end{cases}$$

where $v = (p_2 - p_1) + (t * axis_2 - s * axis_1)$.

The details are routine.

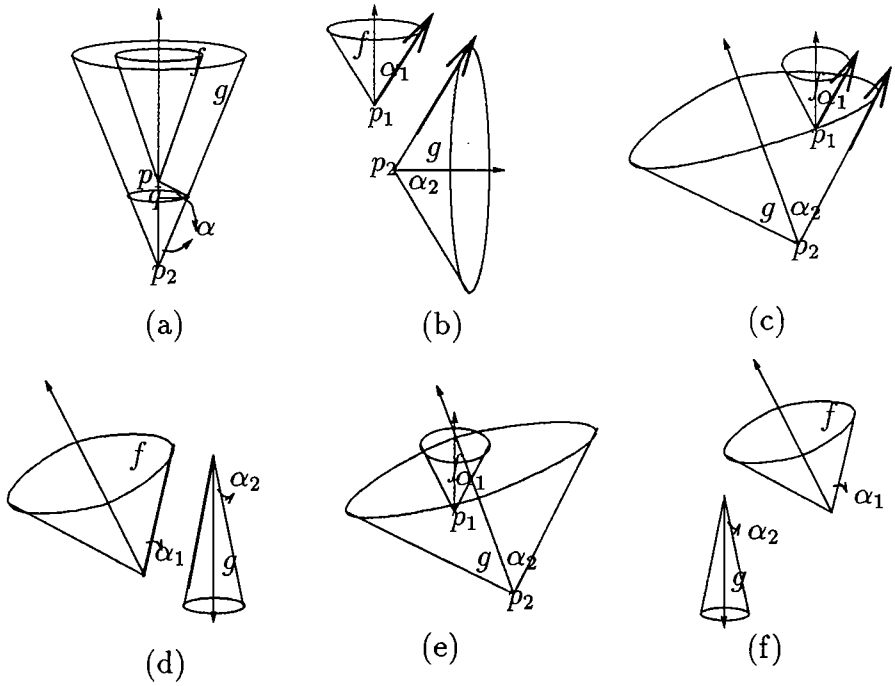


Figure A.10 Cone vs. cone

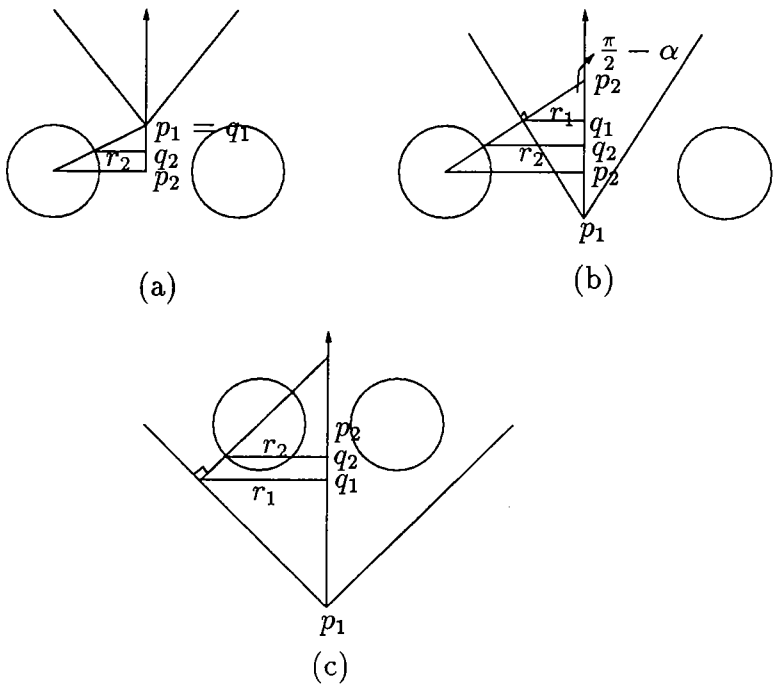


Figure A.11 Cone vs. torus

A.4.1.14 Torus vs. torus

Assume $f = T(p_1, axis_1, major_1, minor_1)$ and $g = T(p_2, axis_2, major_2, minor_2)$. There are eight cases for this problem:

1. The two axes coincide, $p_1 = p_2$, and $major_1 = major_2$. In this case, the two tori are the nearest approach pairs.

2. The two axes are perpendicular to each other, $p_2 - p_1$ is perpendicular to $axis_1$ and $axis_2$, and $major_1 = \|p_1 - p_2\| > major_2$ or $major_2 = \|p_1 - p_2\| > major_1$. In this case, two circles are the nearest approach pairs as shown in figure A.12(a).

3. The two axes are perpendicular to each other, $p_2 - p_1$ is perpendicular to $axis_1$ and $axis_2$, and $major_1 = major_2 = \|p_1 - p_2\|$. In this case, two pairs of circles are the nearest approach pairs for this problem as shown in figure A.12(b).

4. The two axes coincide, $p_1 = p_2$, and $major_1 \neq major_2$. In this case, the nearest approach is a pair of circles as shown in figure A.12(c)(d). The two circles are in the plane $\Pi(axis_1, p_1)$

5. The two axes coincide, $p_1 \neq p_2$ and $major_1 = major_2$. In this case, the nearest approach pair is a pair of circles as shown in figure A.12(e). The two circles are on the cylinder $Cy(a_1, axis_1, major_1)$.

6. The two axes coincide, $p_1 \neq p_2$ and $major_1 \neq major_2$. In this case, the nearest approach pair is a pair of circles as shown in figure A.12(f). The two circles are on a cone whose angle is $\alpha = \frac{major_1 - major_2}{\|p_1 - p_2\|}$.

7. In all other cases, nearest approach pairs will consist of pairs of points.

The first case is trivial.

For the second case, without loss of generality, we assume $major_1 > major_2$. Let $n = \frac{p_1 - p_2}{\|p_1 - p_2\|}$. Then the two circles $Circle(p_1, axis_1, minor_2)$ and $Circle(p_1,\ axis_1,\ major_1 - minor_1)$ are the nearest approach pairs in this case. The circle axes of the tori and the nearest approach pairs are shown in figure A.12(a).

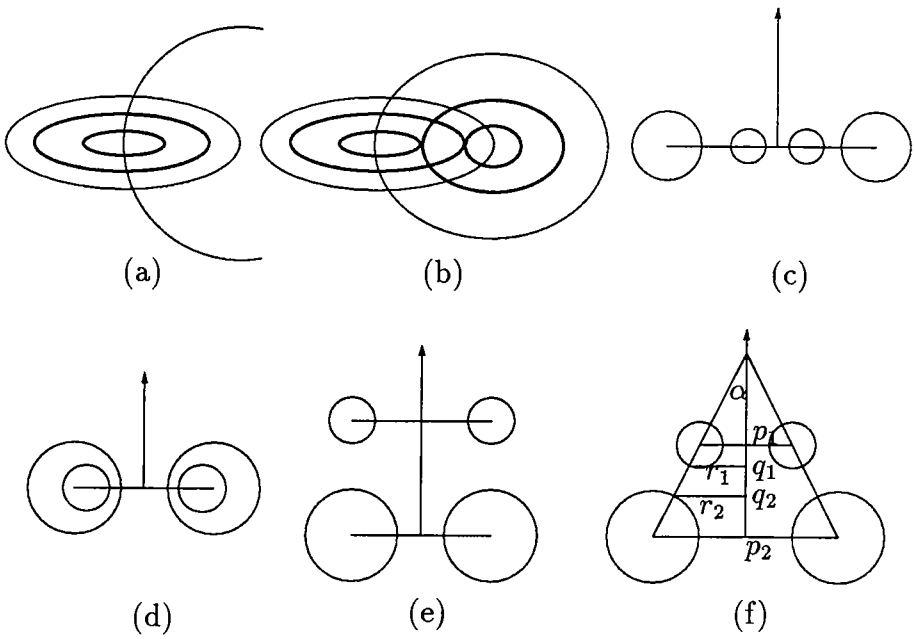


Figure A.12 Torus vs. torus

The nearest approach circles, in the third case, can be found by applying the second case twice. The circle axes of the tori and the nearest approach pairs in this case are drawn in the figure A.12(b).

The nearest approach pairs for the fourth, fifth, and sixth cases are pairs of circles with the same axis. We need to find the centers and the radii of these circles. We only draw the intersection of the tori with a plane that contains the axes of the tori. The nearest approach pairs for the fourth case, shown in Figure A.12(c)(d), can be obtained straightforwardly. The centers of these circles are p_1 and the radii of these circles depend on the relation between $major_1, major_2, minor_1$ and $minor_2$.

For the fifth case, assume that $axis_1$ and $axis_2$ point in the same direction and f is above g. Then, $Circle(p_2 + minor_2, axis_2, major_2)$ and $Circle(p_1 - minor_1, axis_1, major_1)$ are the nearest approach pairs for this case, as shown in figure A.12(e).

Consider the figure A.12(f) for the sixth case. Assume the axes of f and g point in the same direction and $major_1 < major_2$. From

$$\tan\alpha = \frac{major_1}{\|p_1 - q\|} = \frac{major_2}{\|p_2 - q\|} = \frac{major_2 - major_1}{\|p_1 - p_2\|}$$

$$\sin\alpha = \frac{r_1 - major_1}{minor_1} = \frac{major_2 - r_2}{minor_2}$$

$$\cos\alpha = \frac{\|p_1 - q_1\|}{minor_1} = \frac{\|p_2 - q_2\|}{minor_2}$$

We can find

$$r_1 = major_1 + minor_1 * \sin\alpha$$

$$r_2 = major_2 - minor_2 * \sin\alpha$$

$$\|p_1 - q_1\| = minor_1 \cos\alpha$$

$$\|p_2 - q_2\| = minor_2 \cos\alpha$$

where $\tan\alpha = \frac{major_2 - major_1}{\|p_1 - p_2}$.

Then, $Circle(p_1 - \|p_1 - q_1\| * axis_1, axis_1, r_1)$ and $Circle(p_2 + \|p_2 - q_2 * axis_2, axis_2, r_2)$ are the nearest approach pairs in this case.

Table A.1 The number and degree of the equations we have to solve

	sphere	plane	cylinder	cone	torus
sphere	$\perp$	$\perp$	(1)	(2)	(2)
plane		$\perp$	(1)	(2)	(2)
cylinder			(1,1)	(1,2)	(1,4)
cone				(1,2)	(2,4)
torus					(4,4)

We have to solve two degree 4 equations for the seventh cases because

$$\begin{cases} axis_1 \cdot v = \|v\| * \frac{s}{\sqrt{major_1^2+s^2}} \\ axis_2 \cdot v = \|v\| * \frac{t}{\sqrt{major_2^2+t^2}} \end{cases}$$

where $v = (p_2 - p_1) + (t * axis_2 - s * axis_1)$.

Note that there are 16 solutions for the equations. We have to test each one of them so that the nearest approach pairs can be founded.

The following table summarizes the number of equations we need to solve in each case if the nearest approach pairs are finitely many point pairs. The notation $\perp$ means that no equation needs to be solved. One equation with degree i will be represented by (i) and two equations with degree i and j will be represented by (i, j).

A.5 Closest approach pairs involving curves generated by two CSG primitive surfaces

We do not enumerate the cases in which the nearest approach pairs consist of infinitely many point pairs. For example, the nearest approach for a sphere and a circle, or for two circles could be two circles. We only describe the generic case. Before the description of this section, we would like to introduce some abbreviations.

Let f be a primitive surface. Then its implicit equation $f(x_1, x_2, x_3, x_4) = 0$ can be obtained from the feature of the surface, as described in the beginning of this appendix. For testing whether the point $q = (1, q_2, q_3, q_4)$ is on the surface, we write "$q \in f$ test". For example, if $g = Cy(p, axis, r)$ is a cylinder, the associated implicit equation $g(x_1, x_2, x_3, x_4) = 0$ for the cylinder can be produced. Then, the "$q \in g$ test" is the equation $g(1, q_2, q_3, q_4) = 0$ where $q = (1, q_2, q_3, q_4)$.

Four points p_1, p_2, p_3 and p_4, where $p_i = (x_{i,1}, x_{i,2}, x_{i,3}, x_{i,4})$, are on the same plane if and only if the the following derterminant vanishes:

$$\begin{vmatrix} x_{1,1} & x_{1,2} & x_{1,3} & x_{1,4} \\ x_{2,1} & x_{2,2} & x_{2,3} & x_{2,4} \\ x_{3,1} & x_{3,2} & x_{3,3} & x_{3,4} \\ x_{4,1} & x_{4,2} & x_{4,3} & x_{4,4} \end{vmatrix} = 0$$

We write

$$\begin{vmatrix} p_1 \\ p_2 \\ p_3 \\ p_4 \end{vmatrix} = 0$$

as its abbreviation.

A.5.1 Curve vs. surface

Let f, g and h be CSG primitives. We want to find the nearest approach pairs between the surface f and the curve $g \bigcap h$

A.5.1.1 The surface is a plane or sphere

Assume that f is a plane or a sphere and the nearest approach line contains the point $q \in g \bigcap h$. We have to solve 3 equations with 3 variables as listed below (Note that we can always force the homogeneous variable to be equal to 1 if we determine

a finite point):

$$\begin{cases} q \in g \text{ test} \\ q \in h \text{ test} \\ \begin{vmatrix} NP(f) \\ q \\ NP(g,q) \\ NP(h,q) \end{vmatrix} = 0 \end{cases}$$

We give an example for this system of equations. To simplify the example, we set the homogeneous variable to 1. Let $f = S((4,0,0),1)$, $g = Cy((0,0,0),(0,0,1),2)$, $h = Co((0,4,0),(0,0,1),1,1)$, and $q = (1,x,y,z)$. The system of equations we have to solve is:

$$\begin{cases} q \in g \text{ test} & : & (x-2)^2 + (y-2)^2 - 4 = 0 \\ q \in h \text{ test} & : & x^2 + (y-4)^2 - z^2 = 0 \\ \begin{vmatrix} NP(f) \\ q \\ NP(g,q) \\ NP(h,q) \end{vmatrix} & : & \begin{vmatrix} 1 & 4 & 0 & 0 \\ 1 & x & y & z \\ 1 & 0 & 0 & z \\ z & 0 & 0 & x^2+y^2+z^2+z \end{vmatrix} = 0 \end{cases}$$

The third equation is derived by the fact that $NP(f), q, NP(g,q)$ and $NP(h,q)$ are on the same plane. Note also that when g is a cone or a torus, $NP(g,q)$ has degree 2. When g and h are tori, we need to solve 3 equations whose degrees are 4, 4 and 5 respectively.

A.5.1.2 The surface is a cylinder

Assume f is a cylinder whose axis has direction $axis$. We need to solve 4 equations with 3 variables. The equations are listed below:

$$\begin{cases} q \in g \text{ test} \\ q \in h \text{ test} \\ axis \cdot (NP(h,q) - q) = 0 \\ axis \cdot (NP(g,q) - q) = 0 \end{cases}$$

Notice that the last two equations are at most degree 2, even if the surface g is a torus.

A.5.1.3 The surface is a cone

Assume $f = Co(p, axis, len, wid)$ and the nearest approach line contains $q_1 = p + t * axis$ and $q_2 \in g \bigcap h$. We have 4 equations with 4 variables.

$$\begin{cases} q_2 \in g \text{ test} \\ q_2 \in h \text{ test} \\ axis \cdot (q_2 - q_1) = \|q_2 - q_1\| \cdot \frac{wid}{\sqrt{len^2 + wid^2}} \\ \begin{vmatrix} q_1 \\ q_2 \\ NP(g, q_2) \\ NP(h, q_2) \end{vmatrix} = 0 \end{cases}$$

The third equation is a quadratic equation and the fourth equation has degree 6 in the worst case.

A.5.1.4 The surface is a torus

Assume $f = T(p, axis, major, minor)$ is a torus and the nearest approach line contains $q_1 = p + t * axis$ and $q_2 \in g \bigcap h$. We need to solve 4 equations with 4 variables in order to find the closest approach line.

$$\begin{cases} q_2 \in g \text{ test} \\ q_2 \in h \text{ test} \\ axis \cdot (q_2 - q_1) = \|q_2 - q_1\| \frac{t}{\sqrt{major^2 + t^2}} \\ \begin{vmatrix} q_1 \\ q_2 \\ NP(g, q_2) \\ NP(h, q_2) \end{vmatrix} = 0 \end{cases}$$

The third equation has degree 4 and the fourth equation has degree 6 in the worst case.

A.5.2 Curve vs. curve

We want to find the nearest approach line between the curves $g_1 \bigcap h_1$ and $g_2 \bigcap h_2$. Assume the nearest approach line contains $q_1 \in h_1 \bigcap g_1$ and $q_2 \in g_2 \bigcap h_2$. We solve 6 equations with 6 variables.

$$\begin{cases} q_1 \in h_1 \text{ test} \\ q_1 \in g_1 \text{ test} \\ q_2 \in h_2 \text{ test} \\ q_2 \in g_2 \text{ test} \\ \begin{vmatrix} q_1 \\ q_2 \\ NP(g_1, q_1) \\ NP(h_1, q_1) \end{vmatrix} = 0 \\ \\ \begin{vmatrix} q_1 \\ q_2 \\ NP(g_2, q_2) \\ NP(h_2, q_2) \end{vmatrix} = 0 \end{cases}$$

The fifth and sixth equation are based on the fact that $q_1, q_2, NP(g_1, q_1)$, $NP(h_1, q_1)$ and q_1, q_2, $NP(g_2, q_2)$, $NP(h_2, q_2)$ are in the same plane. The degree of these two equations depends on what h_1, h_2, g_1 and g_2 are. For example, if g_2 is a plane or a sphere, then $NP(g_2, q_2)$ is a constant and the degree of the sixth equation will be reduced because one row of the determinant is constant. In the worst case, when h_1, g_1, h_2 and g_2 are all tori, we need to solve 6 equations with degree 4, 4, 4, 4, 6 and 6 respectively.

A.6 The nearest approach pairs between algebraic surfaces and curves

Consider two algebraic surface $f(x_1, x_2, x_3, x_4) = 0$ and $g(x_1, x_2, x_3, x_4) = 0$. Assume $(l_1, \hat{l}_1)$ is a normal line on f and $(l_2, \hat{l}_2)$ is a normal line on g. We also assume the degree of f is m and g is n. We know the degree for $l_1, \hat{l}_1, l_2, \hat{l}_2$ is $m-1, n-1, m, n$ respectively. If $(l_1, \hat{l}_1)$ and $(l_2, \hat{l}_2)$ are the same line, then $(l_1, \hat{l}_1) = r(l_2, \hat{l}_2)$. From it, we find 6 equations with degree $max(m, n) + 1$. With the other two equations $f(1, u_1, v_1, w_1) = 0$ and $g(1, u_2, v_2, w_2) = 0$, we have 8 equations and 7 variables. This is worse than the method proposed by Hoffmann [32]. In his paper, 6 equations with degree $max(m, n)$ and 6 variables need to be solved. So, we conclude that the Plücker coordinate system is not suggested for finding the nearest approach line in this case. A similar argument applies to finding nearest approach lines for algebraic space curves.

Appendix B: Constant-Time Array Initialization

We solve the following problem: Given a matrix M of size $m \times n$, whose entries are random,

1. initialize every element of M to *unassigned* in constant time.
2. For any valid index (i, j), decide whether $M[i, j]$ is *unassigned*, in constant time.
3. Retrieve or store data in M in constant time.

The solution to this problem is standard material in the theory of algorithms.

We use the matrix M itself, with each element a pointer into a stack S. An entry of the stack S is a pair consisting of a pointer to $M[i, j]$ and the value of $M[i, j]$. There is a variable T that records the top of the stack S. The pointers are implemented as integers, with $M[i, j]$ referred to by the value $(i-1)n + j$. The pointer part of $S[j]$ is referred to as $S[j].p$, and the value part as $S[j].v$.

1. To initialize M, assign zero to T.
2. To test $M[i, j]$, retrieve its value k. If k is not in the range $1...r$, where r is the value of T, then $M[i, j]$ is unassigned. Otherwise, if $S[k].p \neq (i-1)n + j$, then $M[i, j]$ is unassigned. Otherwise, $M[i, j]$ has the value $S[k].v$.
3. To retrieve the value of $M[i, j]$, do Step 2 above. To assign u to $M[i, j]$ we proceed as follows: If $M[i, j]$ is not unassigned, then assign u to $S[k].v$, where k is the value of $M[i, j]$. Otherwise, let k be the value of T; assign $k + 1$ to $M[i, j]$; assign $(i-1)n + j$ to $S[k+1].p$; assign u to $S[k+1].v$; increment T by 1.

It is easy to see that this correctly implements all operations, and it is clear that each operation requires constant time, independent of the size of the array or the values of i and j. Moreover, it generalizes to arrays of any size and dimension.

VITA

VITA

Ching-Shoei Chiang was born in Keelung, Taiwan, R.O.C., on January 5, 1961, the son of GoangLong Chiang and Shiow-Mei Hwang Chiang. After completing his work at Keelung High School in 1979, he entered Soochow University and received a Bachelor of Science degree in June of 1983. After graduation he worked at the Computer Center of Soochow University as an operator for a year. In Sepember, 1984, Mr. Chiang continued his education at the Graduate School of The University of Texas at El Paso where he received his M.S. degree majoring in Computer Science in 1986. Mr. Chiang pursued his Ph.D. degree in Purdue University from August 1986, and received his M.S. and Ph.D. degrees in Computer Science in 1990 and 1992, respectively.

His research interests include geometric and solid modeling,, computer-aided geometric design and computer graphics. While at Purdue, he was supported as a research assistant under the supervision of Professor Christoph M. Hoffmann.