國立交通大學

應用數學系

數學建模與科學計算碩士班

碩 士 論 文

對於物體的拓樸結構描述

The Description of topological structure for a shape

研　究　生：吳侑燊

指 導 教 授 ：林松山　教授

中 華 民 國 一百零一年 七 月

對於物體的拓樸結構描述

The Description of topological structure for a shape

研 究 生 ： 吳侑燊　　Student : Yu-Shen Wu

指導教授 ： 林松山 教授　　Advisor : Song-Sun Lin

國立交通大學

應用數學系

數學建模與科學計算碩士班

碩 士 論 文

A Thesis

Submitted to Institute of Mathematic Modeling and Scientific Computing
Department of Applied Mathematics
College of Science
National Chiao Tung University
in Partial Fulfillment of the Requirements
for the Degree of
Master
In
Mathematic Modeling and Scientific Computing
July 2012
Hsinchu, Taiwan, Republic of China

中 華 民 國 一百零一年 七 月

謝誌

本論文的完成，首先我要感謝的是我敬愛的指導老師—林松山教授。老師兩年來的對我的鞭策及教導，除了學業，告訴了我許多待人處事的道理。老師常說：「做人要正直。」此外，老師更給了我許多的機會讓我學習了許多課外的經驗，這些都是得來不易的。在此再次感謝老師的愛惜之心。

其次要感謝的是林文偉老師，不管是在北京清華的日子、在學校的日子，只要有問題找他，他總能給你一個好的方向去尋找答案。除了兩位老師外，何丹期學長的幫忙是我在寫論文時的最大助力，在喜獲麟兒之時，還要抽空與我共同檢視我的論文，辛苦實不在話下。胡文貴學長雖然與我研究方向不同，但是他以身教的方式告訴了我們許多師生之間的互敬互愛，是一位非常值得敬佩的學長。同時要感謝建模所及數學所的同學們，和你們一起讀書玩樂，實在是這兩年最開心的事情了，希望友誼長存。

最後，也是最重要的，是我的家人。是你們全力的支持與關愛，才能成就今天的我，獻上一輩子的感謝。希望未來，我能利用我所學，回饋社會，報答大家對我的愛護。

吳侑燊 2012.07.11

對於物體的拓樸結構描述

研究生：吳侑燊　　　　　　指導教授：林松山

國立交通大學

應用數學系

數學建模與科學計算碩士班

摘要

隨著科技的進步，想要得到高解析度及複雜的三維影像資料是不困難的。然而，我們每次都要處理這麼大量的資料其實是滿浪費資源的。因此，拓樸結構在分析物體上是不可或缺的。首先，我們會先介紹 Reeb graph 的方法，Reeb graph 是一種對函數的拓樸結構，如果能找到一個適當的函數來描述物體，那 Reeb graph 算是對物體的一個拓樸結構的表現。接著我們會介紹 skeleton 的方法，這是一種我們可以最直觀想像的提取方法。而最後我們會對於數學拓樸上找到一個好的基底來描述一個物體結構。利用基底的剪開來實現 Poincaré-Klein-Koebe Uniformization Theorem。

The Description of topological structure for a shape

Student: Yu-Shen Wu   Advisor : Song-Sun Lin

Institute of Mathematic Modeling and Scientific Computing

Department of Applied Mathematics

National Chiao Tung University

July, 2012

Abstract

With the recent advances in mesh acquisition device, polygonal mesh with high resolution and complex structure can be easily to get. Topological structure and is crucial for analyzing the shape of complex mesh model. We introduce Reeb graph first. Reeb graph is the topological structure of the function. The Reeb graph can be regarded as a topological structure if there exists a suitable function defined on mesh. Second, we will introduce the skeleton, which is the most intuitively for us to realize the topological structure of a shape. Finally we want to show the basis of the shape in topology, the homotopy basis and the homology basis. In the end, we compare this method and show our experiment.

Contents

Chapter 1

Introduction

With the developing of computer technology rapidly, the speed of the data processing is much faster than last century. In this improvement, people want to show the objects of the real world on the computer, that is the 3D image. We also want to play computer games in 3D vision, which can supply object rotating, zooming in and zooming out. In industry, we can check the process yield of the products, for example microchip and copies of historical relics. In biomedicine, we check a human bone, gene,and chromosome to know whether the human is heathy or not. In addition, with the conformal geometry developing, we can make 3D movies in incredible time. To reach these goals, we have developed two techniques. In the 3D scanner technology maturation phase, constructing 3D models becomes more convenient. The second one is that we can get and share amount of data very easy because of the high speed of the internet. Besides, in order to supporting the model process, there are many softwares appearing, Maya, Zbrush, AutoCAD, and so on. However, the whole data of three model is very huge that we may hard to storage, classify, and application in some computation complexity. We want to find some representations which can stand for complicated models which have topological structure without using amazing data.

There are many ways to extract the topological structure the the 3D model of the object. The **Reeb graph** is a compact shape descriptors, preserving topological information by level set of a function which is defined on the shape. Similarly, finding **Topological Skeleton** can also emphasize topological properties of the shape, especially the length, width, and connectivity. It also plays an important role to the role of animation. Those can only describe one of the **Homotopy Basis**, which is called **Tunnel Loop** we may give a clear definition later. Each genus may generate two different loops to be the element of a basis of the shape. We want to find a good basis of the shape to cut that can change from origin shape into a topological disk. In Chapter 2, we start to introduce how to find topological structure of the shape from reeb graph which is the topological description of the **Morse Function**. In Chapter 3, we want to show how to shrink a shape to a skeleton. In chapter 4, we explain why we need to introduce the basis of a shape, the we show how to find it. In the end we give a brief conclusion for whole paper.

Chapter 2

Reeb graph

The *Reeb graph* [11] is a 1D graph structure for representing the topology of a function. Once we can define a suitable function on the object surface, the reeg graph of the surface function can be regarded as the topological structure of the object. In this chapter, we review the definition of reeb graph, and discuss the methods to extract the topological structure from the 3D object using reeb graph.

The reeb graph is the topology of a function. By its definition, the function should satisfy the morse function [18]. The morse function comes from the Morse theory [18] which studies the relationship of the different shape in a space and the relation between the critical points, which is defined as follows:

Definition. *Let M be a 2-mainfold with or without boundary. A smooth map $f : M \mapsto R$ is called a morse function defined by following conditions:*

1. *All critical points of f are non-degenerate and lie in the interior of M.*
2. *All critical points of f restricted to the boundary of M are non-degenerate.*
3. *$f(x) \neq f(y)$ for all critical points $x \neq y$ of f and its restriction to the boundary.*

We now define the reeb graph as the topological structure of the morse function.

Definition (Reeb graph)**.** *Let $f : S \to \mathbb{R}$ be a real value function on a compact manifold S. The Reeb graph of f is the quotient space of f in $M \times \mathbb{R}$ by the equivalence relation,*

$$(X_1, f(X_1)) \sim (X_2, f(X_2)) \text{ if and only if } f(X_1) = f(X_2),$$

where X_1 and X_2 are the point of the shape. This means X_1 and X_2 are classified into the same node point if the satisfied the following two conditions: 1. X_1 and X_2 belong to the same interval under the classified of the function f, 2. X_1 and X_2 are connected each other in the corresponding interval.

To extract Reeb graph from a continuous surface, the function value for all the surface points should be evaluated. By classifying all the surface points with the same value of the function, we can get the quotient space from the origin space. The Reeb graph is then constructed by connecting the element of the quotient space related to the original shape.

Since there exist different morse functions for the same surface, the Reeb graphs extracted from the functions may not be the same. In this way we have to find a good morse function to classified the shape. Shinagawa and Kunii used the height function on the object surface to extract the Reeb graph [12]. The height function $h(x, y, z) = z$ may be intuitively to express a shape. A level set is the primage of each height. We want to observe the level set changed when the image value of the height function h, especially we encounter the critical points. There may be four possible results occur at a critical point:

1. The minimum point which all the point around it is higher than it.
2. The first kind of saddle point which will separate the level set into two part after the constant value increasing.
3. The second kind of saddle point which will combine two part into one part after the constant value increasing.
4. The maximum point which all the point around it is lower than it.

After we connect these critical points, we can describe the shape. Moreover, the number of loops in the reeb graph is equal to the surface genus. Now we describ how to construct the reeb graph. The construction of Reeb graph in the work of Shinagawa and Kunii [12] is based on the classification of iso-contours of the function. The process starts by locating the surface point having the global minimum of function, which is denoted as the critical point. Then, trace the iso-contour by increasing the iso-value of the function. Once the

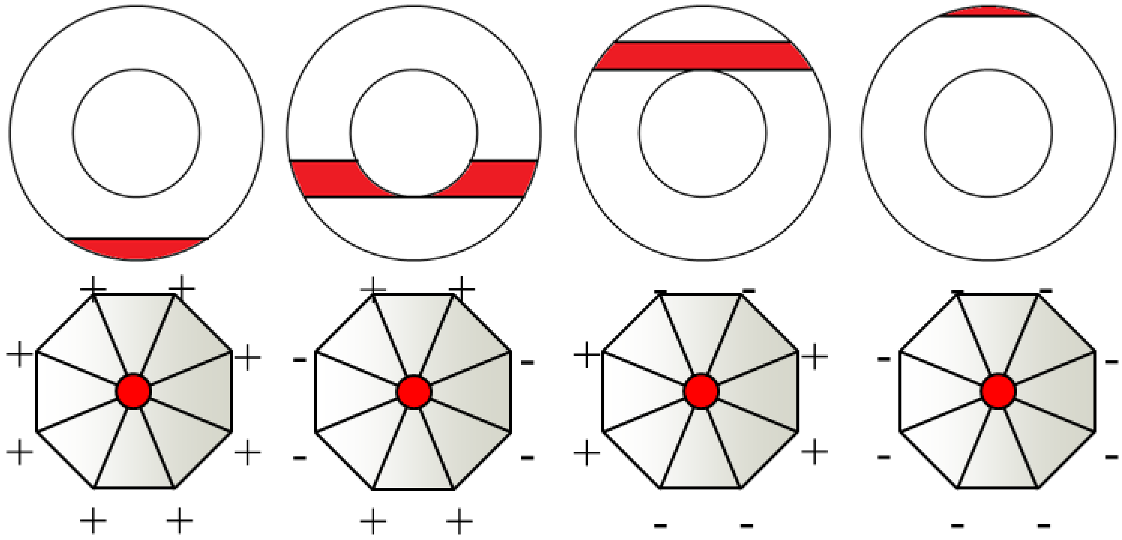


Figure 2.1: Four possible result for critical point

topology of the iso-contour is changed, a critical point is inserted at the position and decide the type of critical point it is. This process is performed until the global maximum of the function is reached. Fig. 2.2 illustrates the process of finding the reeb graph of the donus surface by dunking it in the coffee. We begin from no donut in the coffee. The first critical point is the minimum point at the bottom. Before we pass the second critical point, we found that the topology of part which have dunk in the coffee from a point expands to a topological disk. After passing the second one, we found that portion from a topology disk changing into a cylinder. We also observe that during from a critical point to next critical point, the topology of the object may not change. After passing the third critical point, the portion changed from a cylinder into a punctured torus, which is a 2-mainfold with boundary of genus 2. If we dunked all the donut, that is passing the maximum point, the topology will be a torus in the end.

The height function may be useful when modeling a 3D shape from cross section like CT image. However, the height function is not suitable to identify a shape because it may change the result when transforming such as rotation. In order to handle these problems, Lazarus[5] proposed geodesic distance from a source point for a function, where the geodesic distance is the shortest distance on the surface from point to point. Geodesic

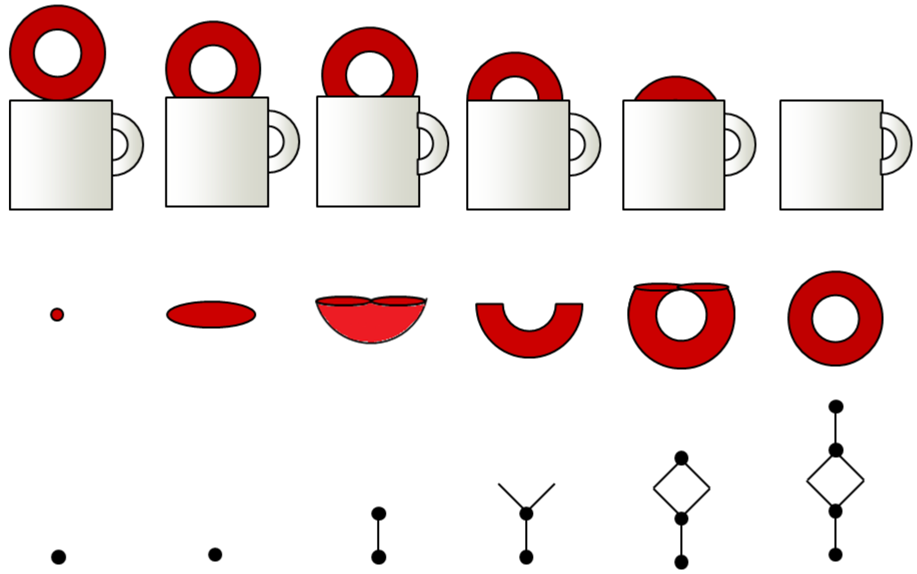


Figure 2.2: Process of construct the reeb graph of torus.

distance may avoid result changing by rotation and some small perturbation. However, to identify a shape, the source point need to be determined automatically and stable. A small change of the shape may not determined the same source point, this is a difficult problem but need to solved which we wish to construct a stable Reeb graph. **Hilaga et al.** [8] extended the geodesic distance function from a source point to the average geodesic distance function. In the average geodesic distance function, the function value for a surface point is defined by the average of geodesic distances from this point to all the surface points as in Eq. 2.1.

$$\mu(v) = \int_{p \in s} g(v,p) dS, \tag{2.1}$$

where $g(v,p)$ is the geodesic distance on surface S.

Because of this function has no source point, it will be more stable. But the function $\mu(v)$ is not invariant to scale of the object. So the function changed into a normalize version,

$$\mu_n(v) = \frac{\mu(v) - \min_{p \in S} \mu(p)}{\max_{p \in S} \mu(p)}$$

This function is very useful for as object deformation, because of the geodesic distance is not change a lot on the surface. Besides, we observed that the value of $\mu_n(v)$ is small when the point is close to the central of the object, otherwise, when the point is far from the center, $\mu_n(v)$ is large. So this function, the normalized integral of geodesic distance continuous function is suitable for topological matching.

However, even we defined a good function to description, it is not useful if we can not use it on mesh. Valerio[16] proposed a new algorithm which can compute the reeb graph of even non-manifold surface. Let a simplicial mesh $\mathbb{K}$ with a piecewise linear function $\mathbf{F}$. First, we do not give an order to input the mesh except, the two vertices of edge must appear before the edge, all edges and vertices of corresponding triangle must appear before the triangle. Now we begin to construct the Reeb graph by adding the mesh. Valerio give the following data structure. As the following figure we see, we construct the node on Reeb graph by corresponding vertex as we see Figure 2.3(a). We also construct the arc in Reeb graph when we add edge into the mesh, each arc stores a list of pointer to edges which are intersect between the interval in the corresponding arc. For each edge we store the highest arc which has stored this edge. To construct the reeb graph easily we give an extra information. As we see in Figure 2.3(a) e_4 not only stores the highest arc a_0 but also stores an extra number which presents link times of e_4 from a_0. The algorithm begin from the empty Reeb Graph. First we add a new vertex, edge, or triangle. For each vertex we use **CreatNode** to generate a new node in the Reeb Graph. Then we call **CreatArc** to generate the arc in Reeb graph. Storing edges which is intersection with the corresponding interval. Then After all three vertices and edges are added, calling **MergePath** to connected the new contour and update all the arcs and edges as we see from Figure 2.3(c) to (d). Finally we removed the edge which need two links from it beginning to the end, as we shown in Figure 2.3(e), e_4 is removed.

In this algorithm we don't need to find the critical point at first. Besides it also has a concept of the persistent homology we may explain later. However, to construct a Reeb graph on the surface we need to give a morse function which is not easy to find, and we may not to realize how does it works for most function intuitively. So, the next chapter we want to introduce another topological structure expression that is usually to be used, a the graph of medial axis, we also called a skeleton in usual.

Algorithm 1 Reeb Greph

Input: a Mesh $\mathbb{K}$

```
for all element in K do
    if element is a vertex v then
        CreatNode(v)
    else if element is a edge then
        CreatArc(v_i,v_j)
    else
        MergePaths(e_i,e_j,e_k)
    end if
    Update the list of arcs and edges.
    if the number of link in e_i is up to two then
        RemoveEdge(e_i)
    end if
end for
```

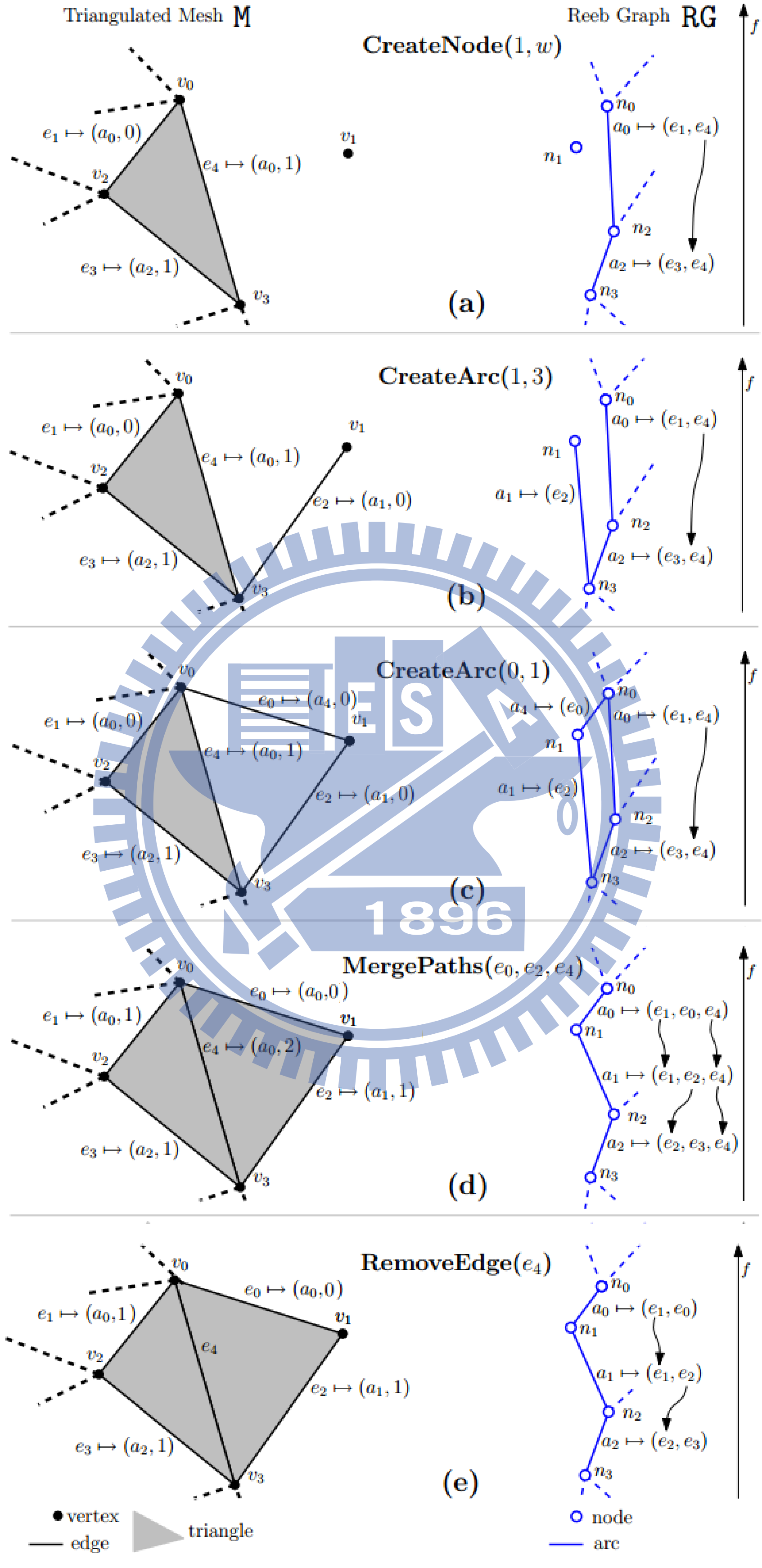


Figure 2.3: Data structure of Reeb graph. [16]

Chapter 3

Skeleton

The most intuitive way to describe the topological structure of an object is by looking at its skeleton. The skeleton of the object is an 1D structure that represents the object's shape and the topological characteristics. In this chapter, we review the skeletonization methods that extract the 1D curved-skeleton from the 3D closed surface. The curve-skeleton is ill-defined, therefore there exists lots of algorithms in the literature proposed with different definitions. It is hard to decide which method we want to choose for our requirement and which algorithm support properties. For example, virtual navigation for medical is strict to require on the center. Curve-skeletons are closely related to the medial axis. The medial axis of the shape is a set of curves which all point on these curves gave more than one closet point on the boundary of the shape. There is another concept of medial axis, Harry Blum[3] defined a medial axis when we defined the shape as the dry grass, set on fire at all points on the boundary until the fire front meet and quench each other. The set of quench points are called medial axis. The skeleton is defined as the locus of the points of maximal inscribed ball,

Definition (skeleton)**.** *Let $S \subset R^3$ be a 3D shape. $B(r, x)$ is a open ball consists of the radius r and the center $x \in S$, which is $B(r, x) = y \in R^3, d(x, y) < r$, where $d(x, y)$ is the distance between x and y.*

However, Even though medial axis and skeleton are closely related, they are not exactly the same[1]. The difference between the medial axis and the skeleton arise on the limited case, the end of the ellipse in [1] for example, the end points only belong to skeleton.

In this paper we categorize many of the existing methods into two classes: volumetric and geometric.

Most existing curve-skeleton methods come from a volumetric discrete representation in 3D space, which can be either the voxelized representation or a distance field function. For voxelized representation, thinning methods attempt to produce a curve-skeleton by iteratively remove voxels from boundary until the thinness is obtain. All thinning algorithms rely on the concept of sample point which is a shape point that can be removed without change the topology of the shape. The advantage of thinning methods is they can only check their neighbor to decided whether the sample point can be removed or not. This may make the algorithm be much more efficient. Distance field methods is defined the smallest distance from each interior point of 3D shape to the boundary of the shape,

Definition (Distance field). *$D(x) = min(d(x, y))$, where d is some distance metric between x in shape and y in boundary of shape.*

The ridges of the field correspond to voxel are locally centered in the shape. Connecting this candidate for constructing an approximate medial surface. There are another field methods to use, for example repulsive forces[9], for every point in the shape do $F(p) = \sum_{b \in \text{boundary}} F_{bp}$, where $F_{bp} = \frac{\overrightarrow{bp}}{\mathrm{d}(bp)}$, d is distance between b and p and $\overrightarrow{bp}$ is the vector from b to p. Thus, every point in the shape has a corresponding vector, as the Figure 3.1. This method determined the potential value at interior points by considering the whole boundary and then is less than sensitive to noise.

However, these previous methods require the volumetric mesh, but most of data of shapes provide only the surface data. So, we need to construct the volume data, and that may be unstable. A disadvantage of the medial axis is that it is sensitive to small change as Figure 3.2 Besides, most of methods need to compare relationship between whole boundary and the interior points that may cost expensive. In order to most shapes only provide the surface mesh, geometry methods can work on only polygon mesh or point

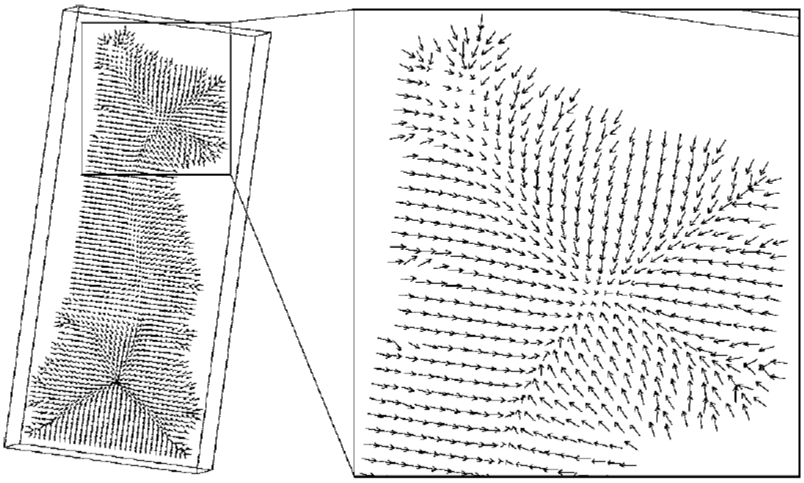


Figure 3.1: Repulsive force field of a 3D chess piece.[9]

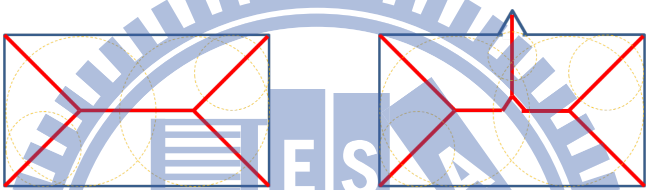


Figure 3.2: Sensitive of the medial axis

set directly. A common approach methods come from the key concept of the **Voronoï diagram** [15]. Reeb graph can also use for constructing the skeleton for 3D model with many different surface function for example geodesic function as we introduced before. Harmonic function can also use for reeb graph [2], proposed by Aujay, embedded the skeleton into the geometry with appropriate refinement, solving the Laplacian equation[2].

There are some methods extracting skeletons directly on surface. Au[10] proposed a effect method instead of using volume data. Given a mesh $G = (V, E)$, where $V = [v_1^T, v_2^T, v_3^T, \cdots, v_n^T]$ are the vertex position. Our goal is to find $S = (N, B)$, where $N = [N_1^T, N_2^T, N_3^T, \cdots, N_m^T]$ are the node position and edges B. Au approached with geometry contraction process that iterative smoothly and collapse the mesh geometry without any voxelization process. The main idea is to iterative smoothly and contract the surface of the shape into an zero-volume mesh without change topology structure of the shape. This problem is like a simple energy minimization problem with constrained Laplacian smoothing. In this way, the energy function consists of two terms, the first term is to contract geometry and the second one is to attract energy to preserve the origin topological structure of the shape. To convert the zero-volume mesh into a curve-skeleton, we perform a connectivity surgery process. We collapse the edge without alters topology. This method has following advantages, maintain the topological structure because of the process without disconnected the origin mesh, works directly on the surface making the method efficient, and because of using iterative smoothly operator, the method can handle the noise naturally.

Next we want to show the detail of the method. We move the vertices along their approximate curvature normal direction by applying Laplacian smoothing iteratively. The formulation is to solve a sparse linear system. Let L be the n-by n curvature-flow Laplace operator with elements,

$$L_{ij} = \begin{cases} \omega_{ij} = \cot \alpha_{ij} + \cot \beta ij, & \text{if } (i,j) \in E \\ \sum_{(i,k)\in E} -\omega_{ik} & \text{if } i = j \\ 0 & \text{others} \end{cases} \tag{3.1}$$

and let V' be smoothly contracted along the normal directions by solving $LV' = 0$. Because we want to preserve the geometry structure, we constrain all the vertices to theirs current position with the different weight. To balance the contract constrain and attraction constrain of geometry we may design the different weight. Therefore, we solve the following system for the vertex position,

$$\begin{bmatrix} \mathbf{W}_L \mathbf{L} \\ \mathbf{W}_H \end{bmatrix} V' = \begin{bmatrix} 0 \\ \mathbf{W}_H V \end{bmatrix} \tag{3.2}$$

where $\mathbf{W}_L$ and $\mathbf{W}_H$ are the diagonal weighting metrics of contraction and attraction constraints. Solving system Eq. 3.2 ones does not make the model into a 1D shape. We need to iterate several times with proper weight. However, using the same weight is too slow to collapsing. Therefore, to increasing the collapsing speed we gain the contraction weight $\mathbf{W}_L$ for every iteration and change the attraction weight $\mathbf{W}_{H,i}$ by the degree of local area change for each vertex i. The iterative process as follow,

1. Solve $\begin{bmatrix} \mathbf{W}_L^t \mathbf{L} \\ \mathbf{W}_H^t \end{bmatrix} V^{t+1} = \begin{bmatrix} 0 \\ \mathbf{W}_H^t V^t \end{bmatrix}$ to update V

2. Update $\mathbf{W}_L^{t+1} = s_L \mathbf{W}_L^t$ and $\mathbf{W}_{H,i}^{t+1} = \mathbf{W}_{H,i}^0 \sqrt{A_i^0 / A_i^t}$ where A_i^t is the one-ring area. and s_L is a power to gain the collapse speed.

3. Construct the new Laplacian operator L^{t+1} by the new vertex position V^{t+1}.

Au.[10] use $\mathbf{W}_L^0 = 10^{-3}\sqrt{A}$ where A is the average area of the surface and $\mathbf{W}_H^0 = 1.0$ for the initial setting. they experiment that when $s_L = 2.0$ the models are usually iterated less than 10 times with the threshold.

We now contract mesh with zero-volume which looks like a 1D skeleton. However, it is still the original mesh. So next we need to simplify the contracted mesh into a 1D graph to get a curve-skeleton. To achieve this goal, we remove collapsed face from the degenerate mesh by applying a series of connectivity surgery operation. We remove each face without change the topology of the model.

The curve-skeleton is useful and popular to research, especially the geometry part. It can show how the object change, that is very useful to animation. However, some researchers want to find the circles directly on the surface of a shape that the circles can not be shrink in to a point in topology, which is called non trivial. Curve-skeleton and reeb graph can only find one kind of the circles. The next chapter we will find another.

Chapter 4

Homotopy basis

In the previous two chapters, we introduced how to description a topological structure of the shape. But many applications of 3D models need to cut the model into a topological disk. Especially studying **Poincaré-Klein-Koebe Uniformization Theorem**. However, Reeb graph and skeleton can only provide one part of cuts, which is called tunnel loop. To find the others, handle loops, we need to find homotopy basis. A homotopy basis includes information about the shape of topological space intuitively. In this paper, we only need to find the first dimension homotopy basis, handle loops and tunnel loops which can generate the shape.

First, we need to give some definition. A **loop** is a continuous map at a base point x $L : [0,1] \to M$ such that $L(0) = L(1) = x$. In other word, a loop on a surface means a curve on a surface and the start point and the end point are the same one. And we say two loops are homotopy equivalent if one loop can be elastically changing to the others continuously. We give a precise definition for homotopy on mathematic:

Definition (Homotopy)**.** *Two loops L and L' through the same base point are homotopic if there exist a continuous map $h : [0,1] \times [0,1] \to M$ such that $h(0,t) = L(t),\ h(1,t) = L'(t)$ and $h(s,0) = h(s,1) = x \quad \forall s,t \in [0,1]$.*

There are three kinds of loop on the surface. The first one may separate the surface into two part after we cut this loop. The second kind the contractible loop. We say a loop is **contractible** if it can be homotopic to a constant loop, in other word the loop may shrink into a point. We are interested in the third kind of loops, non-contractible and non-separate loop. The fundamental group is a group which may quotient all the contractible loop as trivial.

Definition (Fundamental group $\pi_1(M,x)$)**.** *The set of homotopy equivalence classes of loops at x forms a group and the identity of this group is the contractible loops. We form $\pi_1(M,x)$ under concatention:*

1. *Let L and L' be two loops, $L \cdot L' := L(2t)$ if $t < \frac{1}{2}$ else $L'(2t-1)$*
2. *$L^{-1}(t) = L(1-t)$*
3. *Identity:$I(t) = x$*

Definition (Homotopy basis)**.** *There are $2g$ loops generate $\pi_1(M,x)$ called a homotopy basis.*

As the following figure we see, a system of loops is a homotopy basis but not vice versa. What we want is a system of loops, which the surface can be changed into a topological disk by removed it.

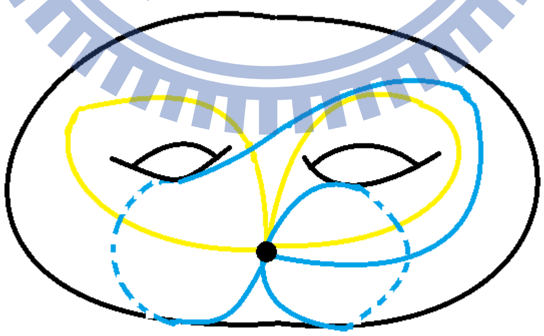


Figure 4.1: A homotopy basis but not a system of loops

Definition (greedy homotopy basis[7])**.** *We define $l_1, l_2, \cdots, l_{2g}$ be the greedy homotopy basis if $M \backslash (l_1 \cup l_2 \cup \cdots l_i)$ is connected for each i*

Our goal is to find cuts that may change our object into the topology disk. A dual graph of a given planar graph G is a graph G' which a vertex in G' is corresponding to a face in G. A edge between 2 face in G' is corresponding to each edge in G. A face in G' is correspond to the vertex in G. The "Dual" means this property is symmetric, in other word if G' is a Gual mesh of G, so is G. We collect some edges in a dual graph into a set with no loop and all connected, we find the corresponding set in G is the face which is connected. In topology, this set in G is just a topology disk. Our strategy is to find connected faces as big as possible on the surface. In the way, we may use minimum spanning tree. A spanning tree of a graph is connected all the vertex together without generating any loop. A spanning tree of a graph may not be unique. So, we can also assign each edge a weight for the collected order. A minimum spanning tree is a spanning tree with the weight choosing that the whole weight is less than any other spanning tree. We also call it minimum weight spanning tree.

We combine these two knowledge and give a algorithm to find homotopy basis, first we find the spanning tree T on graph G which is embedding in the surface for shortest paths from a base point we choose. Then we find a maximum spanning tree T' on the dual mesh G'. Collecting all the rest edges in T' into a set $C*$. The set C which dual edge of the edges in C is in C'. We get the greedy homotopy basis from generating the loops when we add the edge in C into the tree T. We give a brief algorithm to show how to do it,

Algorithm 2 Computing homotopy basis

Input: a graph G

Compute a minimum spanning tree T of the graph G

Find the dual graph G' of G

Compute a maximim spanning tree T' of the dual graph G'

$C = e \mid e' \notin T'$

For each $e_{ij} = [v_i v_j]$ we find a unique path γ_i from base point to v_i and a unique path γ_j from base point to v_j on T

We get a loop $l_{ij} = \gamma_i [v_i v_j] \gamma_j$

Find all loops until we finish using the whole edges in C.

All loops are independent and form a homotopy basis of G.

Erickson and Har-Peled [6]have shown that compute the actual shortest cut graph is np-hard. So most of posterity who is also study the cut gragh may research how to make the algorithm fast with the same basic idea. Until 2008, Tamal K. Dey proposed a novel concept to find a homology basis instead of homotopy basis, which may not use a base point so it can find some global small loops.

4.1 Persistent Homology

In the final of this paragraph we want to introduce a method which is a different application to simplicial complex from persistent topology. This method provides mathematical guarantee on finding handle and tunnel loops.

Definition (Simplex). *A k-simplex is a convex hull with $k+1$ vertice.*

A 0-simplex is a vertex, a 1-simplex is a line, a 2-simplex is a triangle, and a 3-simplex is a tetrahedron.

Definition (Simplicial complex). *A simplicial complex is a topological space which is a union of simplices, in other word we glued some simplices together form a simplicial complex.*

Definition (Chain). *In a simplcial complex $\mathbb{K}$, a p-Chain means the sum of some p-simplices in $\mathbb{K}$, denote$\sum_i a_i\sigma_i$, where σ_i is a p-simplex in $\mathbb{K}$ and a_i belongs to $\mathbb{Z}/2\mathbb{Z}$ in our paper.*

Definition (boundary operator). *A boundary operator is a homomorphism, ∂_p:$C_p \to C_{p-1}$ such that*

$$\partial_p[v_0, v_1, v_2, \cdots, v_p] = \sum_i [v_0, \cdots, \dot{v}_i, \cdots, v_p].$$

where $\dot{v}_i$ means v_i is dropped.

Property. $\partial_{p-1}\partial_p C_p = 0$

For example, as Figure 1 shows, $f_1 = [v_1, v_2, v_3]$ is a triangle. We do the boundary operator on it,

$$\partial f_1 = [v_2, v_3] + [v_1, v_3] + [v_1, v_2] = e_1 + e_2 + e_3.$$

The boundary of a triangle is the sum of three line segments. We just do boundary operator again,

$$\partial\partial f_1 = \partial(e_1 + e_2 + e_3) = \partial[v_2, v_3] + [v_1, v_3] + [v_1, v_2] = v_2 + v_3 + v_1 + v_3 + v_1 + v_2 = 0$$

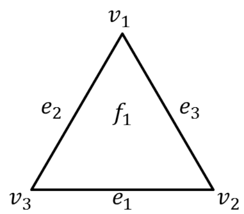


Figure 4.2: Example for boundary operator

Definition (Cycle)**.** *A p-Chain C_p is called a p-Cycle when* $\partial C_p = 0$

∂f_1 is a cycle for the above example.

Definition (Boundary)**.** *A p-chain C_p is called a p-Boundary if there exist a $(p+1)$-chain D_{p+1} such that $\partial D_{p+1} = C_p$.*

It is easy to see that all the boundaries are cycles from the property we mentioned before. We collect all the p-cycles into a set, denoted Z_p, is the kernel of the p-boundary operator, that is $Z_p = \mathrm{Ker}\partial_p$. We also collect all the p-cycles into a set, denoted B_p, is the image of the $p+1$-boundary operator, that is $B_p = \mathrm{Img}\partial_{p+1}$. Therefore $B_p \subseteq Z_p$.

Definition (Homology Group)**.** *The p-dimension homology group is defined as the quotient group,*

$$H_p = \frac{Z_p}{B_p}$$

In this section, H_p is a vector space with a basis. The dimension of H_p is the number of elements in a basis, and it is also the pth Betti's number β_p. We are interested in the first homology group H_1, and a 1-cycle we may called intuitively a loop.

Now we can give a definition for handle loop and tunnel loop. Let $\mathbb{K}$ be a closed surface in $\mathbb{R}^3$. $\mathbb{M}$ separate $\mathbb{R}^3$ in to two parts: (I) for inside and $\mathbb{O}$ for outside, all of them include $\mathbb{M}$.

Definition (Tunnel loop)**.** *A loop which is a boundary in $\mathbb{O}$ but not a boundary in $\mathbb{I}$ called a Tunnel loop. In other words, Tunnel loop is trivial in $H_1(\mathbb{O})$ and nontrivial in $H_1(\mathbb{I})$*

Definition (Handle loop)**.** *A loop which is a boundary in $\mathbb{I}$ but not a boundary in $\mathbb{O}$ called a Handle loop. In other words, Handle loop is trivial in $H_1(\mathbb{I})$ and nontrivial in $H_1(\mathbb{O})$*

By definition, we found that no loop can be both a handle loop and a tunnel loop. However a loop may neither be handle loop nor tunnel loop. For example, the loop shown as figure. Cutting this loop as a boundary may not expand to a disk. The result will be as a band rotating twice and connecting two sides together. This is not desirable in our goal. This method is based on a pairing of simplices which is a key concept in persistent

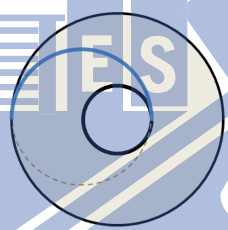


Figure 4.3: The loop is neither handle nor tunnel.

homology[13]. We construct simplicial complex $\mathbb{K}$ by a nested sequence of complex called filtration,

$$\phi \subset \mathbb{K}_1 \subset \mathbb{K}_2 \subset \cdots \subset \mathbb{K}_n = \mathbb{K}.$$

We give a inclusion map $f : \mathbb{K}_{i-1} \hookrightarrow \mathbb{K}_i$ defined by $f(x) = x$. Then a map between the homology groups induced by inclusion map is $f_* : H_p(\mathbb{K}_{i-1}) \to H_p(\mathbb{K}_i)$. The persistent homology studies how homology groups change over the following filtration,

$$0 \to H_p(\mathbb{K}_1) \to H_p(\mathbb{K}_2) \cdots \to H_p(\mathbb{K}_n) = H_p(\mathbb{K}).$$

Now we must give every simplex a signal, which is either a positive simplex or a negative simplex. We assume $\mathbb{K}_i - \mathbb{K}_{i-1} = \sigma_i$, where σ_i is a simplex. This means we add one simplex each time in the filtration. A simplex is called a positive p-simplex by creating a non-boundary p-cycle after adding it. The other is killing of an existing $(p-1)$-cycle called a negative p-simplex.

4.2 Implementation

We give an example to elaborate the above classification. The simplicial complex will be constructed by following order: when adding a new p-simplex, all simplices belongs to the $(p-1)$-chain which is the boundary of this p-simplex being added in the past. As the Figure 4.4 you can see, we add all vertices from $\mathbb{K}_1$ to $\mathbb{K}_4$. Because a vertex is a non-boundary cycle obviously by definition, so all the vertices are positive simplices. Next step, we add e_1 to $\mathbb{K}_4$. Because $\partial e_1 = v_1 + v_4$, adding e_1 will change two cycles into one cycle, in other word it kills a non-boundary cycle. We choose the youngest one cycle v_4, the latest creating one, to be killed. Thus e_1 is a negative simplex. We add the edges as a negative simplex until adding e_4, adding will create a cycle $e_1 + e_2 + e_4$, and it is also a non-boundary cycle. So e_4 is a positive simplex. We follow these rules to separate every simplex. As we can see, adding a positive simplex will increase the dimension of $H(\mathbb{K})$, otherwise adding a negative we get the opposite effect.

By the persistent homology theory, a negative p-simplex is paired with a positive $(p-1)$-simplex. For example e_1 is paired with v_4 as we mentioned. This is the key concept in persistent homology, each pair tells the time of a non-boundary cycle birth and what time it was killed in the filtration. This concept is also used in Reeb graph in the past section. We construct the reeb graph by paired the critical point which will express the start and end.

Now we want to introduce how we pair the simplicial complex in detail. First we need to decision the simplex is positive or negative. If the simplex is negative, then we need to pair a positive simplex with it. By the persistence homology theory we pair the youngest one,which is appear the latest in the same homology group and unpaired. Let *sigma* is a new p-simplex we add in the filtration. We do the boundary operation on it and get a chain $c = \partial\sigma$. Assume d is the youngest simplex in the chain, check whether it is paired

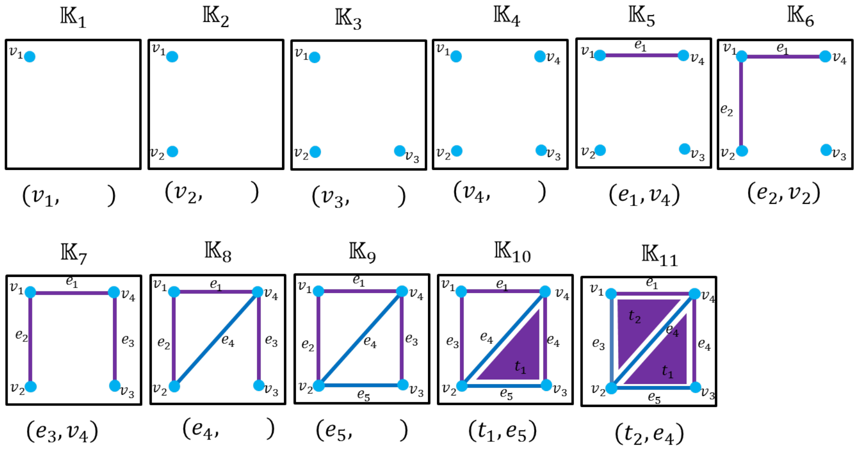


Figure 4.4: Example for classification

or not. If d is paired withσ', update $c = c + \partial\sigma'$. This step will cancel some simplices because we module 2 in this paper. Updating the youngest simplex d and checking again and again until the chain c is empty or the simplex d is unpaired. This while-loop is the most important procedure during we are finding the loops we want. After breaking while-loop, if c is empty that means there exist a d simplex that is unpaired, the σ is negative simplex paired with d. Otherwise, σ is a positive simplex when c is empty. We may also observe that the paired edge is the youngest edge in the corresponding loop which is killed.. During the while-loop, we find a face that every time we update the chain c is also a cycle which is expand from the first one, $\partial\sigma$. These cycles are homology equivalent. This is also helpful for us to find a good loop in geometrically.

Algorithm 3 Pair(σ)

Input: a p-simplex σ

A chain $c = \partial\sigma$

d is the youngest positive $(p-1)$-simplex in c

while d is paired or c is non-empty **do**

Let c' be the cycle which is killed and paired with d

$c = c + c'$ module 2

Update the youngest positive $(p-1)$-simplex d

end while

if c is non-empty **then**

σ is a negative p-simplex paired with d

else

σ is a positive p-simplex

end if

We have introduced the pair algorithm. Now we describe an algorithm for finding handle loop and tunnel loop on an arbitrary closed surface. First step we have to describe how to find them topological guarantee. We assume the input surface is presented with a simplicial complex $\mathbb{K}$. This means we can have the whole simplicial complex to represent both inside space $\mathbb{I}$ and outside space $\mathbb{O}$. What we need in those is only up to 2-simplex of the space. Assume $\mathbb{I}$ and $\mathbb{O}$ are given, we add all simplex on surface $\mathbb{M}$ from vertices to faces into the filtration. Then doing pair on this filtration. Since the dimension of $H_1(M)$ is $2g$, the algorithm will generate $2g$ unpaired positive edges. Here we know if the youngest edge belong to each loop on $\mathbb{M}$ is not one of unpaired edges. Then it is a boundary cycle. In other words, If we have a loop which youngest edges is unpaired edge on $\mathbb{M}$, it means the loop is not a boundary cycle in $\mathbb{M}$. A non-boundary cycle in M may also not a non-boundary cycle in $\mathbb{I}$ or in $\mathbb{O}$, or neither. The next thing we need to do is determining which loop is a boundary in $\mathbb{I}$ and which is in $\mathbb{O}$. After finishing pair on $\mathbb{M}$, we add inside space $\mathbb{I}$ into the filtration up to 2-simplex. Then continue to pair with filtration until the unpaired edges on $\mathbb{M}$ is paired. Since rank of $H_1(\mathbb{I})$ is g, we will pair g

unpaired edges in the filtration. That means we killed g loops according to each unpaired edge. Those loops is a boundary in $\mathbb{I}$, but not boundary in $\mathbb{O}$. By definition these are called Handle loop. Now we add outside space $\mathbb{O}$ into the filtration and do the same way to find the other g loops. Those are non-boundary in $\mathbb{I}$, but boundary in $\mathbb{O}$, which are called Tunnel loop. Because we observe the unpaired edge is generated from $\mathbb{M}$, then it is the youngest edge in the corresponding loop when we paired the inside space and outside space. So we can guarantee that all the edges in each loop are lying on $\mathbb{M}$. Collecting these loops to be a basis of $H_1(\mathbb{M})$.

Algorithm 4 Handle and Tunnel loops

Input: $\mathbb{M}$ is a complex in $\mathbb{K}$

Output: Handle loops and Tunnel loops

For each simplex σ in $\mathbb{M}$, Pair(σ)

UE is the set of unpair edges.

For each simplex σ in $\mathbb{I}$, Pair(σ)

if d is the unpaired edge in UE and paired with the σ **then**

Output the first loop consists with all edges lying on the $\mathbb{M}$ during the while-loop as the handle loop.

Delete d from UE

end if

For each simplex σ in $\mathbb{O}$, Pair(σ)

if d is the unpaired edge in UE and paired with the σ **then**

Output the first loop consists with all edges lying on the $\mathbb{M}$ during the while-loop as the tunnel loop.

Delete d from UE

end if

Now we can find the handle loops and tunnel loops and guarantee them to be topological correct. But we can not guarantee them to be a good geometrically. Out target is to find them as small as possible. Some methods may use the concept of universal covering space to find the shortest way from a point to corresponding point. This is the same homotopy type of loops. However we need to find the shortest loops with homology equivalent. In our algorithm, we have known that all loops are homology equivalent during while-loop. We also known that the loops is expanding from one of the triangles. In this way, we know the loop is longer than before during every time of the while-loop. So we need to get the first time all edges in loop paired unpaired edge are on $\mathbb{M}$. That will be more close to the origin face, instead of the latest loop.

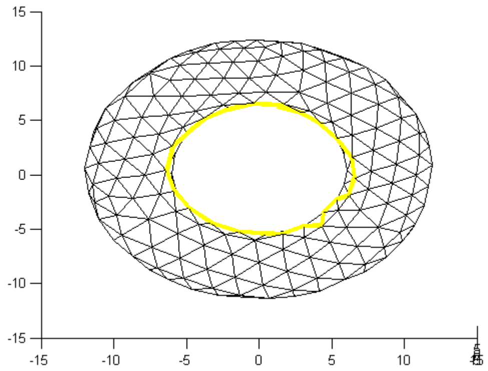
In 2009, Tamal[14] revisited this method, adding some rules to make the algorithm faster. In the past we do not give an order for adding triangles on surface. Now we give a new rule that is when we add a new triangle, it should be adjacent to the boundary of the mesh we added from filtration so far. And another rule is the mesh must remains a topological disk all the time. In our reading, this is just a concept of spanning tree on a dual graph of the mesh, which we have introduced before.

To use this algorithm, there is a problem we encounter that is how can we construct the outside mesh. Because in our knowledge, the outside may not bounded which we can not construct the mesh infinity. We only can be easy to find handle loops. Therefore, we add a concept that we may turn over the surface from inside to outside. Then, the outside turn into inside, and inside become outside. Under this concept, we can find tunnel loops as handle loops after we turn over the surface. However, we can not really to turn over the surface in mesh on the computer. Because after we turned over the surface, the mesh maintain the same one. The difference between the earlier and the later is the normal of the triangle in mesh is opposite. Therefore the normal of the triangles turn from point to inside into point to outside. And there is no bounded for normal direction and that why we can not construct the outside space. To solve this problem we only need to construct a box that can bounded the surface. Then the inside space after we add the box is changed where it is between the surface and the box. Under this space we may find handle loops which is the tunnel loops of the origin surface. The following figures are the samples.

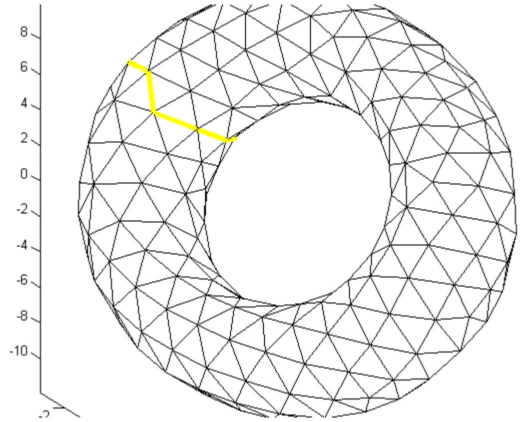
Besides, we can have a tip which may let algorithm faster. We observe that Let $t(\sigma)$ denote the time stamp of the simplex adding into the filtration. Assume d is the youngest edge in loop c, i.e. $t(d)$ is the largest time stamp among all loop in c. Let u be the oldest unpaired edge. We observe that if $t(d) < t(u)$ at while-loop, then the volume triangle is positive. This may help us to save much time in add the volume triangles.

4.3 Result

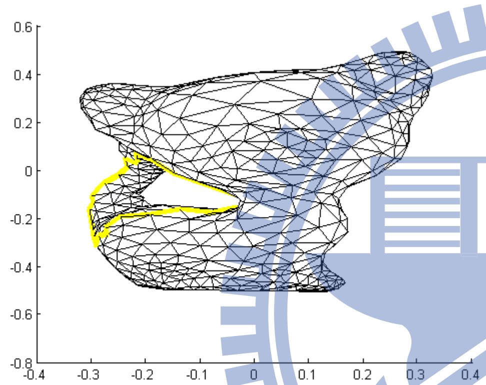
We use Matlab2010 with I5 CPU and 4G RAM. We construct the volume mesh by using Netgen software. The first one is torus which has 1608 vertices and 3212 faces. It costs 26 seconds to find handle and tunnel loops. The second one is kitten which is also one hole and has 610 vertices and 1220 faces and costs 17 seconds. The last one is three tori which has 4524 vertices and 9506 faces and costs 120 seconds. We can easily to know that homology basis is not restrict to a base point. So, it can find small loops as homology group in global. However, the algorithm of persistence homology can only find the surface which is closed and bound without boundary. On the other hand the algorithm of greedy homotopy basis can be used on the surface with or without boundary even though we know that find a shortest path of homotopy basis is np-hard[6]. In 2007, Xiaotain[17] used the universal covering space to find the shortest path for homotopy basis which can find the shortest point in local point. But finding a fastrst algorithm is still a challenge for us. And let persistent homology work on the surface with boundary is another problem.



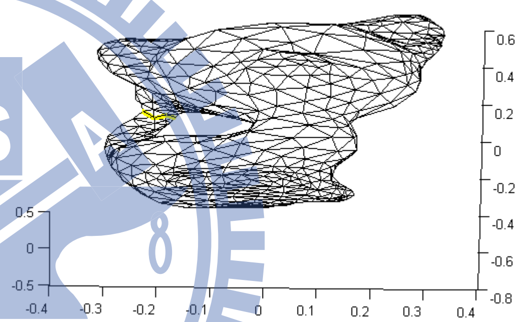
(a) Tunnel loop of torus



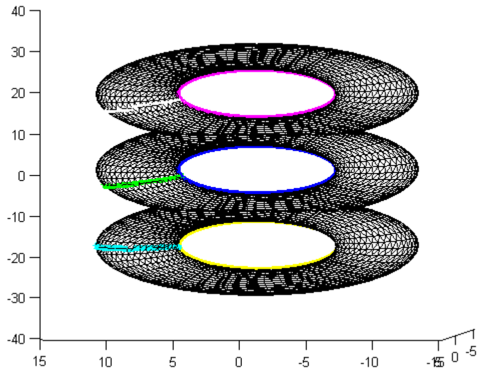
(b) Handle loop of torus



(c) Tunnel loop of kitten



(d) Handle loop of kitten



(e) Handle and Tunnel loops of 3 tori

Figure 4.5: Tunnel loop and Handle loop

Chapter 5

Conclusion

We have introduced the Reeb graph, skeleton, homotopy basis , and homology basis. First two description usually be used to check the equivalence of topological structure and topological matching. Reeb graph can be representation the topology of the function which may express the object. However, it is not easy to construct the function to describe a object. Skeleton is intuitively to show the activity of the object. But it is hard to maintain the topology correct. The other two is useful for conformal mapping which need to cut a surface into a topological disk. The homotopy basis can not only work on compact manifold, also works on the manifold with boundary. However, the shortest loop of homotopy basis is still np-hard. homology basis can only work on compact manifold and so far the volume construct is required. But we can find a global shorter loops we want. But even though they are not the same type, all of them can be used on the action of the movie making, animation, and kinds of object moving.

Bibliography

[1] A. Lieutier, Any Open Bounded Subset of R^n Has the Same Homotopy Type than Its Medial Axis, Proc. ACM Shape Modeling Int"l Conf., 2003

[2] AUJAY, G., HÉ TROY, F., LAZARUS, F., AND DEPRAZ, C. 2007. Harmonic skeleton for realistic character animation. In Sympo- sium on Computer Animation, 151V160. .

[3] Blum, Harry (1967), "A Transformation for Extracting New Descriptors of Shape", in Wathen-Dunn, W., Models for the Perception of Speech and Visual Form, Cambridge, MA: MIT Press, pp. 362V380.

[4] CORNEA, N. D., MIN, P., AND SILVER, D. 2007. Curve-skeleton properties, applications, and algorithms. IEEE Transactions on Visualization and Computer Graphics 13, 3, 530V548.

[5] F. Lazarus and A. Verroust. Level Set Diagrams of Polyhedral Objects. Proc. Symp. Solid Modeling, pp.130-140, 1999.

[6] J. Erickson and S. Har-Peled. Optimally cutting a surface into a disk. Discrete Comput. Geom. 31:37V59, 2004.

[7] Jeff Erickson , Kim Whittlesey, Greedy optimal homotopy and homology generators, Proceedings of the sixteenth annual ACM-SIAM symposium on Discrete algorithms, January 23-25, 2005, Vancouver, British Columbia

[8] Masaki Hilaga , Yoshihisa Shinagawa , Taku Kohmura , Tosiyasu L. Kunii, Topology matching for fully automatic similarity estimation of 3D shapes, Proceedings of the

28th annual conference on Computer graphics and interactive techniques, p.203-212, August 2001

[9] N. Cornea, D. Silver, X. and R. Balasubramanian, Computing Hierarchical Curve-Skeletons of 3D Objects, The Visual Computer, vol. 21, no. 11, pp. 945-955, 2005.

[10] Oscar Kin-Chung Au, Chiew-Lan Tai, Hung-Kuo Chu, Daniel Cohen-Or, and Tong-Yee Lee, Skeleton Extraction by Mesh Contraction, ACM Transactions on Graphics (SIGGRAPH 2008 issue), Vol. 27, No. 3, August 2008, pp. 44:1-44:10.

[11] REEB, G. 1946. Sur les points singuliers d'une forme de pfaff completement intergrable ou d'une fonction numerique [on the singular points of a complete integral pfaff form or of a numerical function]. Comptes Rendus Acad.Science Paris 222, 847V849.

[12] SHINAGAWA, Y., AND KUNII, T. 1991. Constructing a Reeb graph automatically from cross sections. IEEE Computer Graphics and Applications 11, 44V51

[13] Tamal K. Dey , Kuiyu Li , Jian Sun , David Cohen-Steiner, Computing geometry-aware handle and tunnel loops in 3D models, ACM Transactions on Graphics (TOG), v.27 n.3, August 2008

[14] Tamal K. Dey, Kuiyu Li: Persistence-based handle and tunnel loops computation revisited for speed up. Computers and Graphics 33(3):351-358 (2009)

[15] Voronoï, Georgy (1908). "Nouvelles applications des paramètres continus à la théorie des formes quadratiques". Journal für die Reine und Angewandte Mathematik 133: 97V178

[16] Valerio Pascucci , Giorgio Scorzelli , Peer-Timo Bremer , Ajith Mascarenhas, Robust on-line computation of Reeb graphs: simplicity and speed, ACM Transactions on Graphics (TOG), v.26 n.3, July 2007

[17] Xiaotian Yin, Miao Jin, Xianfeng Gu: Computing shortest cycles using universal covering space. The Visual Computer 23(12): 999-1004 (2007)

[18] S. YOSHIHISA, T. L. KUNII AND Y. L. KERGOSIAN. Surface coding based on Morse theory. IEEE Comput. Graphics Appl. 11 (1991), 66V78.