An Euler Characteristic Curve Based Representation of 3D Shapes in Statistical Analysis

by

Zining Ma

Department of Statistical Science
Duke University

Date: _____

Approved:

_____

Sayan Mukherjee, Advisor

_____

David B. Dunson

_____

Scott C. Schmidler

A thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Department of Statistical Science
in the Graduate School of
Duke University

2021

ABSTRACT

An Euler Characteristic Curve Based Representation of 3D Shapes in Statistical Analysis

by

Zining Ma

Department of Statistical Science
Duke University

Date: _____

Approved:

_____

Sayan Mukherjee, Advisor

_____

David B. Dunson

_____

Scott C. Schmidler

An abstract of a thesis submitted in partial fulfillment of the
requirements for the degree of Master of Science
in the Department of Statistical Science
in the Graduate School of
Duke University

2021

Abstract

3D shape analysis is common in many fields such as medical science and biology. Analysis of original shape data is challenging and could be computationally heavy. A simplified representation of 3D shapes could help developing accessible shape analysis methods. In this paper we propose a method to generate a specific form of 3D shapes representation that could be applied in statistical analysis. We use Euler Characteristic curves to create the represention of shapes and utilize scaling function bases from diffusion wavelet to generate the representation. We discuss the details of our method and in the last we apply our method on a shape classification problem to test the performance of the representation.

Acknowledgements

Thanks greatly to my advisor Sayan Mukherjee for the research opportunity and his guidence. Thanks to David Dunson and Scott Schmidler attending my thesis committee. Thanks to Henry Kirveslahti for the discussions about possible approaches. Thanks for department of statistical science for providing the excellent courses in statistics, machine learning and information theory. I could never notice what I really want to do without the experience at Duke.

Thanks to my friends for their company during the two years at Duke. And lastly, great thanks to my parents for helping me getting over the depressing period before graduation.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

3D shape analysis is common in fields such as medical imaging and morphology [1, 2]. In this paper we introduce a practical method which obtains a representation of 3D shapes that could be utilized in shape analysis. Our method applies Euler Characteristic (EC) curves [1] to form representations and utilizes scaling function bases in diffusion wavelet [3] to construct an efficient search process for informative EC curves.

In this paper, the term 3D shape specifically refers to the surface of a shape restored in polygon mesh format, which is a common format of 3D shapes in 3D computer graphics. A polygon mesh consists of vertices, edges and faces of a shape. Further, we focus specifically on triangle mesh (all faces are triangles). A triangle mesh is stored as $M = (\mathcal{V}, \mathcal{F})$, where $\mathcal{V}$ is the list of vertices and $\mathcal{F}$ is the set of faces (each element of $\mathcal{F}$ is a list of 3 integers $[i, j, k]$ indicating a triangle with vertices $\mathcal{V}_i$, $\mathcal{V}_j$ and $\mathcal{V}_k \in \mathcal{V}$).

In the following sections of this chapter, we introduce two background works of our method: Euler Characteristic curves and scaling function bases in diffusion wavelet. In **Chapter** 2, we state our method and discuss the details. In **Chapter** 3, we visualize the outcomes of our method and give an application of the outcomes in shape classification. In **Chapter** 4, we conclude our work and discuss about possible improvements.

1.1 Euler Characteristic curves and shape statistic

In order to describe Euler Characteristic curves, we introduce Euler characteristic first. Given a 3D shape M, its Euler characteristic is a topological invariant calculated as

$$\chi(M) = \#V(M) - \#E(M) + \#F(M)$$

where $V(M), E(M), F(M)$ denote the sets of vertices, edges and faces of M, respectively.

Next, we define the filteration of a shape $M = (\mathcal{V}, \mathcal{F})$. Given a direction $\{\boldsymbol{v}\}$ in $\mathbb{S}^2$ (set of unit vectors in $\mathbb{R}^3$), a filteration, parametrized by height $r \in \mathbb{R}$, of M is defined as:

$$M(r)_{\boldsymbol{v}} = (\mathcal{V}(r)_{\boldsymbol{v}}, \mathcal{F}(r)_{\boldsymbol{v}})$$

where

$$\mathcal{V}(r)_{\boldsymbol{v}} = \{\boldsymbol{x} \in \mathcal{V} : \boldsymbol{x} \cdot \boldsymbol{v} \leq r\}$$

and

$$\mathcal{F}(r)_{\boldsymbol{v}} = \{(i, j, k) \in \mathcal{F} : \mathcal{V}_i, \mathcal{V}_j, \mathcal{V}_k \in \mathcal{V}(r)_{\boldsymbol{v}}\}$$

In words, on a fixed direction $\boldsymbol{v}$, $M(r)_{\boldsymbol{v}}$ is the "subshape" of M which consists of vertices and faces below height r. As r increases, $M(r)_{\boldsymbol{v}}$ grows as more vertices and faces are added and will finally recover the original shape M. Similar to $\chi(M)$, the Euler characteristic of $M(r)_{\boldsymbol{v}}$ could be computed:

$$\chi_M(r)_{\boldsymbol{v}} := \chi(M(r)_{\boldsymbol{v}})$$

With Euler characteristic and shape filteration, we can define the Euler characteristic curve (EC curve).

Definition 1.1. *Given the shape M and a direction $\boldsymbol{v} \in \mathbb{S}^2$, an EC curve is the function:*

$$\begin{aligned} f_{\boldsymbol{v}} : \quad & \mathbb{R} \to \mathbb{Z} \\ & r \mapsto \chi_M(r)_{\boldsymbol{v}} \end{aligned} \tag{1.1}$$

EC curves on all directions in $\mathbb{S}^2$ could be expressed in function form:

$$\begin{aligned} F_M: \quad & \mathbb{S}^2 \to \mathbb{Z}^n \\ & \boldsymbol{v} \mapsto f_{\boldsymbol{v}} \end{aligned} \tag{1.2}$$

[1] proved that F_M, which is called Euler Characteristic Transform in their paper, could serve as a sufficient statistic of shape M.

Clearly $f_{\boldsymbol{v}}$ is a function on the continuous set $\mathbb{R}$. In practice (e.g. the examples in [1]), a downsample is performed. Rather than $\mathbb{R}$, EC curves are computed on a finite discrete set $\{h_0, h_1, ..., h_{n-1}\}$ where $h_i = -R(M) + i * \delta$ and $\delta = 2R(M)/(n-1)$. $R(M) = max_{\boldsymbol{x} \in \mathcal{V}} ||\boldsymbol{x}||_2$ is the radius of M and n is the number of samples. Such downsample simplifies the storage and computation of EC curves. We call such a downsampled EC curve as practical EC curve.

Definition 1.2. *Given the shape M, number of samples n and a direction $\boldsymbol{v} \in \mathbb{S}^2$, a practical EC curve is a vector in space $\mathbb{Z}^n$:*

$$f_{\boldsymbol{v}} = (\chi_M(h_0)_{\boldsymbol{v}}, \chi_M(h_1)_{\boldsymbol{v}}, ..., \chi_M(h_{n-1})_{\boldsymbol{v}})^T \tag{1.3}$$

with $\{h_0, h_1, ..., h_{n-1}\}$ defined above.

In the rest of this paper, the term EC curve refers to practical EC curve rather than the one in **Definition** 1.1 if not specified.

Further more, F_M is computed on a finite set of directions $\boldsymbol{V} \subset \mathbb{S}^2$, and its EC curves are practical EC curves rather than the one in **Definition** 1.1. Thus a practical statistic of M is obtained as $\mathcal{T}_M(\boldsymbol{V}) = \{F_M(\boldsymbol{v}) : \boldsymbol{v} \in \boldsymbol{V}\}$.

[4] shows that a finite number of Euler Characteristic (EC) curves could convey all the information of a 3D shape and a bound of the quantity is given for a family of shapes. As a result, there exists a subset $\boldsymbol{V} \subset \mathbb{S}^2$ that $\mathcal{T}_M(\boldsymbol{V})$ could be a sufficient statistic of M and replace it in shape analysis.

1.2 Scaling function bases in diffusion wavelet

Scaling function bases are outcomes of diffusion wavelet proposed in [3]. [3] introduces a method which computes diffusion wavelet for multiscale analysis. Scaling function bases are interim outcomes of the method and could be applied in clustering data points in a metric space at multiple scales.

The method firstly constructs a graph from the data set and then inspects the diffusion operator T on the graph to generate scaling function bases of T, T^2, T^4, ... as grouping indicators at various scales. Specifically, given a set of observations $\{x_i\}$, the data set is transformed into a graph where each vertex stands for a data point and the weights of connections are determined by a pairwise metric, for example, $W_{ij} = exp(-\kappa d(x_i, x_j)^2)$ for $i \neq j$ and $W_{ii} = 0$ conventionally.

The diffusion operator T is defined as $I - L$ where $L = I - D^{-\frac{1}{2}} W D^{-\frac{1}{2}}$ (D is degree matrix) is the symmetric normalized Laplacian. Physically the diffusion operator could illustrate a process of heat diffusion on the data cloud. T^i illustrates the diffusion procedure between i time units, thus T, T^2, T^4, ... could illustrate the procedure at different time scales.

At each resolution level j corresponding to T^{2^j}, the scaling function bases, which are the main components of the vector space spanned by T^{2^j}, are obtained by applying sparse QR decomposition on T^{2^j}. These scaling function bases have form $\{\boldsymbol{\phi}_k^j\}$ where $j = 1, 2, ..., J$ corresponds to resolution levels and for each j, $k = 1, 2, ..., K_j$ corresponds to groups at level j. Each $\boldsymbol{\phi}_k^j$ is a vector with length equal to the quantity of data points, indicating the intensity of each point in kth group at resolution j. Large j indicates coarse resolution level.

Chapter 2

Method

2.1 Overview

The method we introduce in this paper focuses on the question that how to find the EC curves that could represent a shape efficiently in practical problems. First, we discuss two facts that lead to our method.

In practical problems, complete information of a shape is not necessary. A great proportion of information of a shape recovered by a statistic could be sufficient in subsequent analysis. A general representation could be formed by a few EC curves and the contribution of the rest becomes less and less. As a result, a practical representation could be obtained from EC curves less than the upper bound given in [4].

Further more, $F_M(\boldsymbol{v})$ is continuous in $\boldsymbol{v}$ [5, 6, 1]. Thus for every $\boldsymbol{v} \in \mathbb{S}^2$, $F_M(\boldsymbol{v})$ could be an estimation of other EC curves in the neighbourhood $\{\boldsymbol{x} \in \mathbb{S}^2 : \langle \boldsymbol{x}, \boldsymbol{v} \rangle \leq \theta\}$ for some small θ.

Considering these two facts, a representation could be constructed by two components.

Definition 2.1. *A representation of a shape M is composed of two parts:*

- *a finite set of subsets $\mathcal{S} = \{A_1, A_2, ..., A_N \subset \mathbb{S}^2\}$ that $\forall A_i \in \mathcal{S}$ is a connected set, $\cup_i A_i = \mathbb{S}^2$ and $A_i \cap A_j = \emptyset$ for $i \neq j$*
- *a finite set of EC curves $\mathcal{C} = \{F_1, F_2, ..., F_N\}$ where $F_i = F_M(\boldsymbol{v}_i)$ for one and only one $\boldsymbol{v}_i \in A_i$. F_i is called the representative curve of A_i.*

In words, a representation consists of a partition of $\mathbb{S}^2$ and one EC curve sampled from each set in the partition. Our method is designed to generate such a representation. In the method, we specifically utilize triangulation to produce $\mathcal{S}$ (i.e. $\forall A \in \mathcal{S}$ is a triangle region)

and on each triangle A, its representative curve is sampled on its center. In addition to the representation, our method further performs clustering of regions in $\mathcal{S}$ by applying scaling function bases introduced in **section** 1.2. The grouping outcome is exploited in updating the representation and providing simplified representations in shape analysis.

Now we state the ourline of our method:

1. An initial triangulation is provided to produce the initial representation.
2. Clustering of triangles is performed by applying scaling function bases and then the grouping outcome is obtained.
3. Based on the grouping outcome, certain triangles are subdivided into smaller triangles. As a result, the triangulation and consequently the representation are updated.
4. By repeating steps 2 and 3, the representation is updated iteratively until certain stop condition is satisfied.

In the following sections, we discuss the details of these steps. In **section** 2.2 we present the details of triangulation. In **section** 2.3 we specific the computation of scaling function bases and the criterion of clustering in step 2. In **section** 2.4 we explain the strategy of selecting triangles for further subdivision in step 3. In **section** 2.5 we show the stop condition in step 4. In **section** 2.6 we introduce the simplified representations obtained from the grouping outcome.

2.2 Triangulation

Firstly, we define the triangles on $\mathbb{S}^2$. Given 3 directions $\{\boldsymbol{v}_i\}_{i=1,2,3} \subset \mathbb{S}^2$, a triangle is defined as

$$tri = \{\boldsymbol{v}/||\boldsymbol{v}||_2 : \boldsymbol{v} = (\boldsymbol{v}_1, \boldsymbol{v}_2, \boldsymbol{v}_3) \cdot \boldsymbol{\lambda}, \boldsymbol{\lambda} \in \mathbb{R}^3, ||\boldsymbol{\lambda}||_1 = 1, \lambda_i \geq 0 \text{ for } i = 1, 2, 3\} \tag{2.1}$$

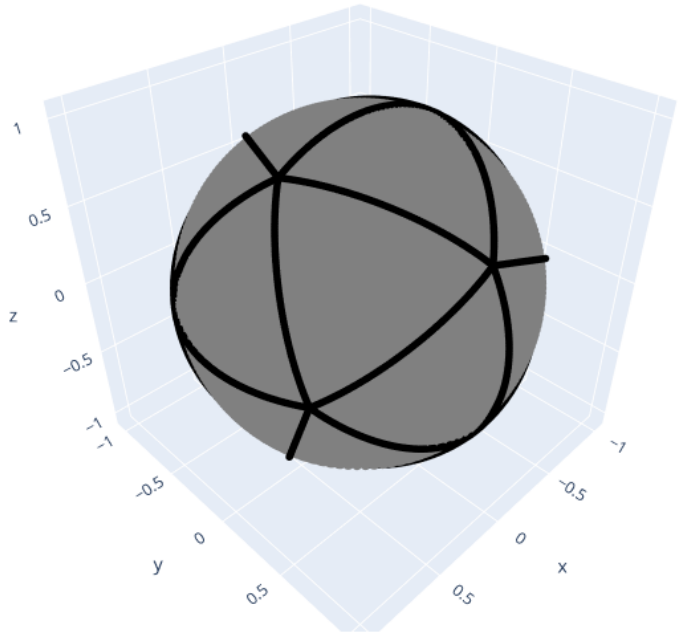


Figure 2.1: Spherical icosahedron.

The term triangle actually stands for spherical triangle, in which case the 3 edges of the triangle are geodesics and the center c of tri is the unit vector on the direction of $(\boldsymbol{v}_1 + \boldsymbol{v}_2 + \boldsymbol{v}_3)/3$.

The purpose of triangulation is to subdivide $\mathbb{S}^2$ into triangles and use the EC curve on each triangle center to represent a triangle area. In this case, small difference between the EC curves in a triangle area and the one on the center, i.e. small $max_{\boldsymbol{v}\in tri}\,||F_M(\boldsymbol{v}) - F_M(\boldsymbol{c})||_2$, is expected. When no information is known about shape M, the best strategy is the minimization of $max_{\boldsymbol{v}\in tri}\langle \boldsymbol{v}, \boldsymbol{c}\rangle$. Thus among triangles with same area, the equilateral triangle is the best choice. As a consequence, spherical icosahedron, which is composed of 20 spherical equilateral triangles, is appropriate for initial subdivision. **Figure** 2.1 shows a spherical icosahedron.

The next step is subdivision. **Figure** 2.2 is an example of subdivision. Triangle ABC, the original triangle, is subdivided into triangles ADF, BDE, CFE and DEF where D,E,F are middle points of arcs AB, BC and CA respectively. It's worth mention that under the case

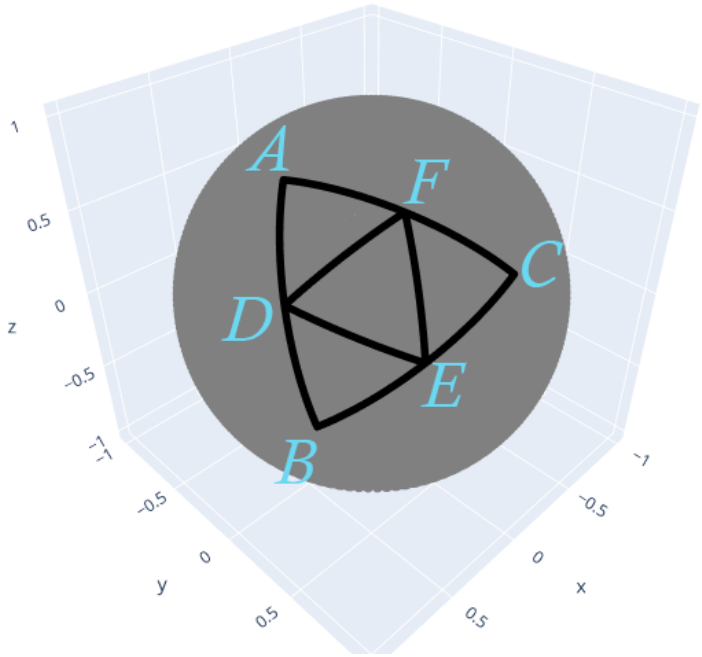
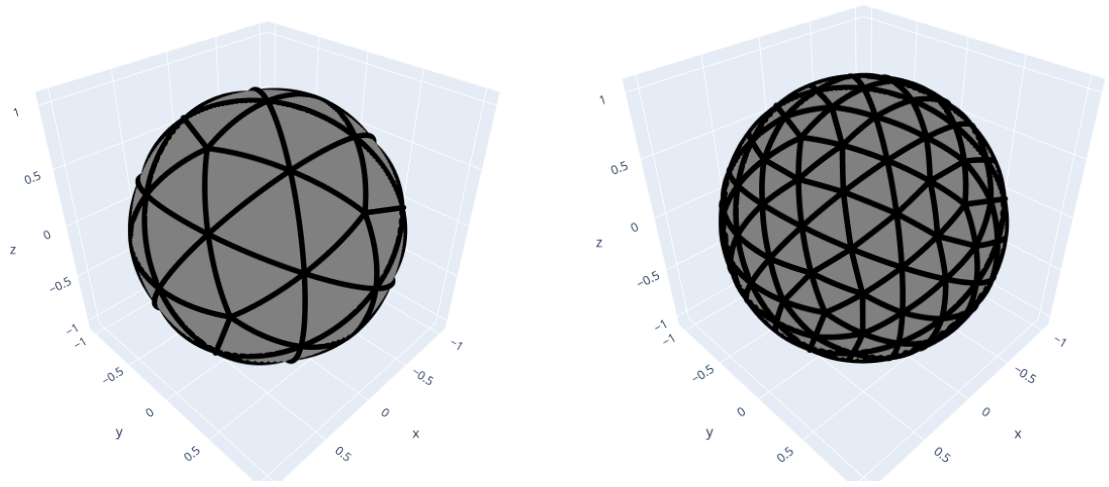


Figure 2.2: Subdivision Example.

triangle ABC is equilateral. ADE, BDF and CFE are no longer equilateral triangles because $\mathbb{S}^2$ is not Euclidean. However the subdivided triangles will always be acute triangles, which will not result in large $max_{\boldsymbol{v}\in tri}\langle \boldsymbol{v}, \boldsymbol{c}\rangle$ relative to triangle area.

As stated in **section** 2.1, the method acts iteratively and the precision of initial iteration affects the precision of the final outcome. An icosahedron initial subdivision could be insufficient for a complex shape. Thus a simple technich is applied which subdivides all triangles several times before the first iteration. Rather than a 20-triangles triangulation, the first iteration could start from an 80-triangles triangulation or a 320-triangles triangulation which are generated from subdividing all triangles once or twice respectively. **Figure** 2.3 shows the resulted triangulation of these 2 cases.



(a) 80-triangles triangulation.

(b) 320-triangles triangulation.

Figure 2.3: Two cases of initial triangulation.

2.3 Grouping via scaling function bases

In this section we present the details of constructing the diffusion graph and the computation of scaling function bases. Then we specify the application of scaling function bases in clustering the triangles.

2.3.1 Construction of graph

Denote a triangulation of $\mathbb{S}^2$ with $\mathcal{S} = \{tri_i\}_{i=1,\dots,N}$ and EC curves on their centers as

$$\{F_i = F_M(\boldsymbol{c}_i)\}_{i=1,\dots,N} \tag{2.2}$$

A complete weighted graph (V, W) is constructed where $V = \{v_i\}_{i=1,\dots,N}$ with each v_i standing for a triangle tri_i and $W = \{W_{ij}\}_{i,j=1,\dots,N}$ with each W_{ij} standing for the similarity between tri_i and tri_j.

Following the examples in [3], we tried two formulations of W:

The first one reflects the similarity between EC curves:

$$W_{ij} = \frac{1}{n}\sum_{k=1}^{n} I((F_i - F_j)\cdot \boldsymbol{e}_k = 0) \tag{2.3}$$

where $I(\cdot)$ is indicator function. It's easy to prove that $d_{ij} = 1 - W_{ij}$ serves as a metric in $\mathbb{Z}^n$, thus the larger W_{ij} is, the closer two curves are in $\mathbb{Z}^n$. However, **Equation** (2.3) is not appropriate because it violates the requirement in [3] that the diffusion operator T should be a sparse matrix. And in fact the experiments we tried showed that **Equation** (2.3) failed in grouping the EC curves at meticulous resolution level.

The second one, imitating the heat kernal in exponential kernal family [7], has the form

$$W_{ij} = exp(-\kappa * ||F_i - F_j||_2^2/n) \tag{2.4}$$

Equation (2.4) not only reflects the distance between curves but also generates a sparse operator T. In experimental results it could group the EC curves at a meticulous resolution level in comparison to **Equation** (2.3). **Equation** (2.4) is utilized in our method.

2.3.2 Diffusion matrix

As stated in **section** 1.2. A basic formulation of diffusion matrix is

$$T = D^{-\frac{1}{2}} W D^{-\frac{1}{2}} \tag{2.5}$$

where $D_{jj} = \sum_{i:i\neq j} W_{ij}$ is the degree matrix. **Equation** (2.5) works well for an impartial triangulation, for example the triangulations shown in **Figure** 2.3. However, as the triangulation is updated iteratively, some triangles are subdivided more times than others and as a result triangles of various sizes are produced. In such case, **Equation** (2.5) could be problematic.

The problem raises because scaling function bases tend to classify together the data points that lie closely to each other. The regions where data points cumulate are likely to be group centers. In other words, the grouping outcome depends highly on the density of

data points. Suppose an EC curve F_i, representing triangle tri_i, lies at the marginal area of its group. If tri_i is subdivided during next iteration, 3 additional EC curves will cumulate around F_i and the density around F_i quadruples. Then the region around F_i could shift from the margin to the center of a group. As a result of such shifting, the grouping alters dramatically and a stable outcome does not exist.

As an adaption to the problem, we propose a modification to **Equation** (2.5). Denoting area of each triangle tri_i as s_i, first we define the flow matrix E as

$$E_{ij} = s_i W_{ij} \tag{2.6}$$

Then we compute the modified T as

$$T = ED^{-1} \tag{2.7}$$

where D is the diagnal degree matrix with $D_{jj} = \sum_{i:i\neq j} E_{ij}$.

The modification reflects a heat transfer process among triangles (or vertices in graph (V, W)). The flow matrix E illustrates the heat flow between triangle pairs. Specifically, the jth column of E dipicts the heat transfer from tri_j to other triangles, where the heat flux from tri_j to tri_i is W_{ij} and the rate of heat flow is $s_i * W_{ij}$. Adding an multiplier s_i to heat flow is equivalent to adding a weight s_i to tri_i so that it's contribution to data density is modified. As a result, even if tri_i is subdivided during following iterations, it's influence in grouping remains approximately the same. The normalized flow matrix T in **Equation** (2.7) then illustrates a similar heat transfer process which is energy conserved. In the following sections and our method, T is computed using **Equation** (2.7).

2.3.3 Scaling function bases and grouping

Given the diffusion operator T, an algorithm which computes the scaling function bases is introduced in [3]. Here we use a modified algorithm shown in **Algorithm** 1. The modification is necessary because [3] assumes T to be self-adjoint, however the modified T in **Equation** (2.7) is not self-adjoint thus the original algorithm is not applicable.

Algorithm 1 Diffusion bases

procedure DIFFUSIONBASES(T,J,SpQR,ϵ)

// Input:

// T: Diffusion matrix

// J: number of resolution levels to compute

// SpQR: function of sparse QR decomposition [3]

// ϵ: precision of sparse QR decomposition

// Output:

// scaling function bases $\{\Phi^j\}_{j=1}^{J}$

$\Phi^{-1} \leftarrow I$

for $j = 0$ to $J - 1$ **do**

$Q, R \leftarrow \mathrm{SpQR}(T,\epsilon)$

$\Phi^j \leftarrow \Phi^{j-1}Q$

$T \leftarrow RTQ$

end for

return $\{\Phi^j\}_{j=0}^{J-1}$

end procedure

For a fixed resolution level j, columns of Φ^j are written in $\{\phi_k^j\}_{k=1,2,\ldots,K_j}$. These vectors consitute the bases of diffusion operator T^{2^j}. As a consequence of applying sparse QR decomposition which returns the approximate QR decomposition of input matrix, K_j decreases as j increases [3]. In other words, the number of groups decreases as the resolution level becomes coarser.

In each vector ϕ_k^j, its ith component $(\phi_k^j)_i$ represents the intensity of tri_i in group k. Thus at each resolution level j, the group assignment of tri_i could be determined by

$$G_i^j = \underset{k}{argmax}(\phi_k^j)_i \tag{2.8}$$

In words, each triangle is assigned to the group where its intensity maximizes.

In practice, Φ^0 is discarded because it fails in representing groups. **Equation** (2.6) shows that the heat flow from a triangle to itself is 0 during a diffusion process of 1 time unit. As a result, no matter how close it is to its neighbours, they will not be assigned to a same group. However, after a diffusion process of 2 time units (corresponding to T^2) or more, the heat flow originated from tri_i will return partially to itself and as a result tri_i could be assigned to the same group of its neighbours.

2.4 Determination of further subdivision

Given a triangulation $\mathcal{S} = \{tri_i\}_{i=1,\ldots,N}$, grouping is performed following the procedure in **section** 2.3.2 and group assignments $\{G_i^j\}$ are obtained. The next step is to find the triangles whose representative curves actually can not represent themselves and, consequently, need further subdivision. Intuitively, the selection could be determined by checking whether EC curves on the vertices of a triangle belong to the same group as the triangle center. However, in **section** 2.3, group assignments are not computed for vertices and as a result the strategy is not applicable.

Exploiting the idea of heat diffusion, we propose a new strategy in selecting such tri-

angles. Denote the member of a group $g \in \{1, 2, ..., K_j\}$ at level j as

$$m_g^j = \{i : G_i^j = g\} \tag{2.9}$$

Then for each group $g \in \{1, 2, ..., K_1\}$ at level $j = 1$, a lower bound of heat flow from the group to itself is computed:

$$l_g = \min_{i \in m_g^1} \sum_{k \in m_g^1, k \neq i} T_{ki} \tag{2.10}$$

For each vertex $\boldsymbol{v}$, a heat transfer process from $\boldsymbol{v}$ to all triangles is established where the heat flow from $\boldsymbol{v}$ to tri_i is computed by:

$$E_{\boldsymbol{v}}(tri_i) = s_i * exp(-\kappa ||F_M(\boldsymbol{v}) - F_i||_2^2/n)/Z_{\boldsymbol{v}}$$

where $Z_{\boldsymbol{v}}$ is a normalizer so that $\sum_i E_{\boldsymbol{v}}(tri_i) = 1$ (as a result the process is energy conserved). Then we define the heat ratio from $\boldsymbol{v}$ to group g as

$$R_{\boldsymbol{v}}(g) = \sum_{i \in m_g^1} E_{\boldsymbol{v}}(tri_i)$$

Whether a vertex $\boldsymbol{v}$ belongs to group g is simply determined by checking the condition

$$R_{\boldsymbol{v}}(g) \geq l_g \tag{2.11}$$

If **Equation** (2.11) is satisfied, then $\boldsymbol{v}$ is considered to be a potential member of group g because in such case, $\boldsymbol{v}$ is closer to group g at least than the most marginal point in the group.

Given the strategy, triangles which have vertices that do not belong to the group of itself are subdivided when updating the triangulation.

2.5 Stop condition

Similar to other iterative algorithms, one simple stop criterion could be a max iteration parameter. However, a criterion that reveals more information of the convergence of the algorithm could be more helpful. Here we use number of groups as such criterion.

In practice, as the triangulation becomes more and more subtle, K_j increases and empty groups emerge. We call non-empty groups as valid groups, and denote the set of valid groups as

$$V_G^j = \{g \in \{1, 2, ..., K_j\} : \#m_g^j > 0\} \tag{2.12}$$

Finally, the number of (valid) groups at level j is defined as

$$N_G^j = \#V_G^j \tag{2.13}$$

We use N_G^1 as stop criterion. The stablization of N_G^1 indicates that all additional curves discovered in the latest iteration belong to some existed group at the most meticulous level and no outstanding curves are found. Thus the stablization of N_G^1 is considered as the stop condition. If N_G^1 stops increasing between two iterations, then the dynamic search procedure stops. In practice, we use both this condition and a max iteration parameter.

2.6 Simplified representations

In the above sections of this chapter, we introduced how our method is performed. The outcome of the method is a representation consists of partition $\mathcal{S} = \{tri_1, tri_2, ..., tri_N\}$ and representative curves $\mathcal{C} = \{F_1, F_2, ..., F_N\}$, and grouping outcome $\{G_i^j\}$. In **section** 2.1 we mentioned that the grouping outcome could provide simplified representations for shape analysis, in this section we show the computation of simplified representations.

Simplified representations has similary components as representation in **Definition** 2.1. However, its partition is not composed of connected triangle areas and its representative curves are not sampled directly on $\mathbb{S}^2$. Simplified representations are obtained as below:

Definition 2.2. *Given a representation $\mathcal{S} = \{tri_1, tri_2, ..., tri_N\}$ and $\mathcal{C} = \{F_1, F_2, ..., F_N\}$ of shape M and the grouping outcome $\{G_i^j\}$, the level j simplified representation of M is composed of a partition $\mathcal{S}^j = \{A_g^j : g \in V_G^j\}$ and representative curves $\mathcal{C}^j = \{F_g^j : g \in V_G^j\}$, where*

$$A_g^j = \bigcup_{i \in m_g^j} tri_i \tag{2.14}$$

and

$$F_g^j = \frac{\sum_{i \in m_g^j} s_i * F_i}{\sum_{i \in m_g^j} s_i} \tag{2.15}$$

In words, an element of the partition is the union of triangles in a same group and its representative curve is the average curve weighted with respect to triangle areas. In **section** 3.2 we will show an application of level 1 simplified representation.

Chapter 3

Results

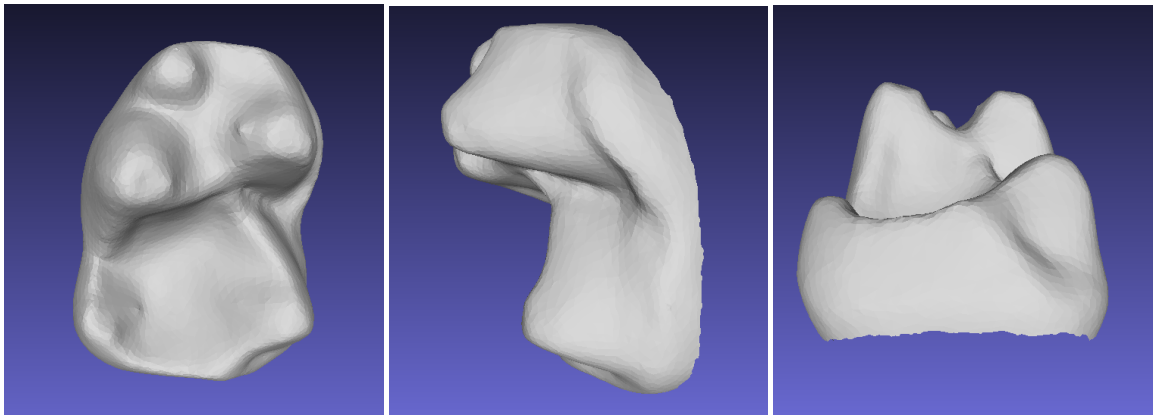
In this chapter we will show an experimental result of our method and one application in shape classification. We make use of a Lemuridae teeth data set [8, 9]. Figures of 3 views are captured in MeshLab [10] and other 3D figures are drawn using plotly [11].

3.1 Visualization

In this section we visualize the outcome of our method. The example we use is the outcome of a tooth shape 'a10_sas.off' from the Lemuridae teeth data set. **Figure** 3.1 shows the three view of the mesh. Our method is run with parameters $n = 1000$, $\kappa = 10$, $J = 5$, $\epsilon = 5\%$ and initiates from an 320-triangles subdivision shown in **Figure** 2.3b.

The first result we present here is the convergence of the method. **Figure** 3.2 shows the variation of N_G^1 with iterations. At 6th iterations, N_G^1 decreases compared to 5th teration and the iteration stops. In the outcome, the partition consists of 1649 triangles and the level 1 simplified representation consists of 52 valid groups.

Next, we present 5 groups at level 1 that cover most area of $\mathbb{S}^2$. **Figure** 3.3 and **Figure** 3.4 shows how these groups are located on $\mathbb{S}^2$ together with their representative EC curves.



(a) (b) (c)

Figure 3.1: Three view of 'a10_sas.off'

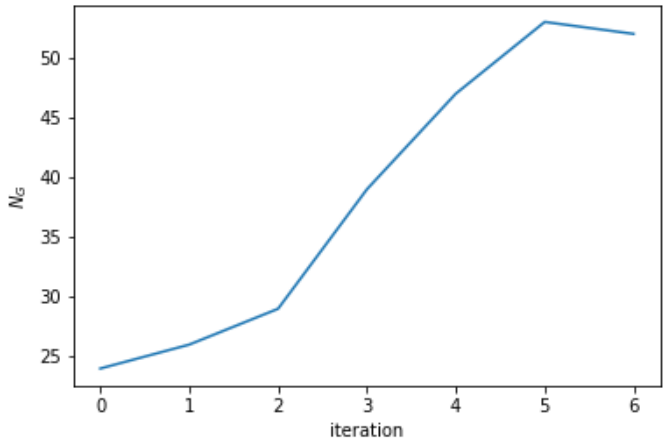
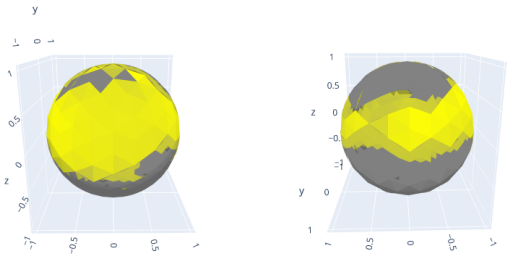
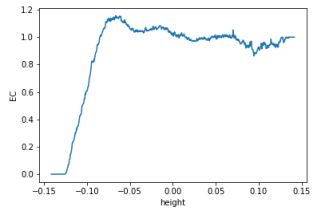


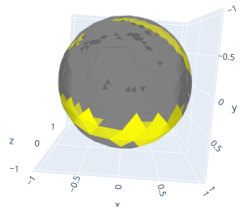
Figure 3.2: Method convergence.



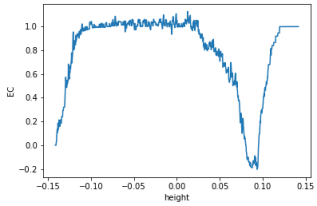
(a) Region of group 1



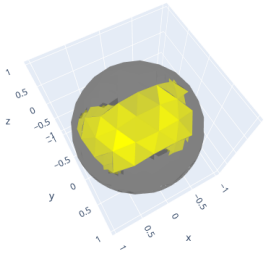
(b) Representative curve of group 1



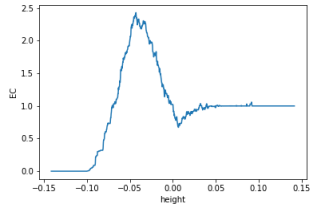
(c) Region of group 2



(d) Representative curve of group 2

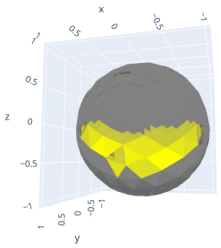


(e) Region of group 3

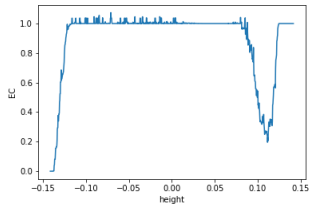


(f) Representative curve of group 3

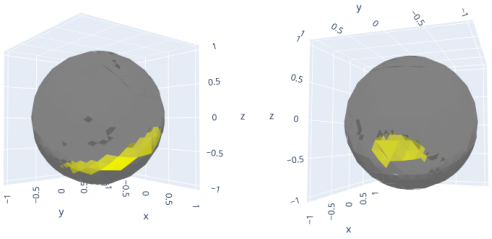
Figure 3.3: Groups 1,2,3 of 'a10_sas.off' and their representative EC curves.



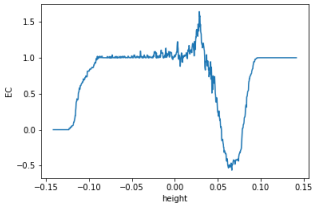
(a) Region of group 4



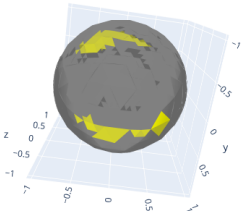
(b) Representative curve of group 4



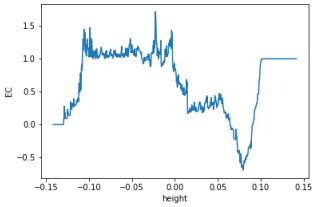
(c) Region of group 5



(d) Representative curve of group 5



(e) Region of group 6



(f) Representative curve of group 6

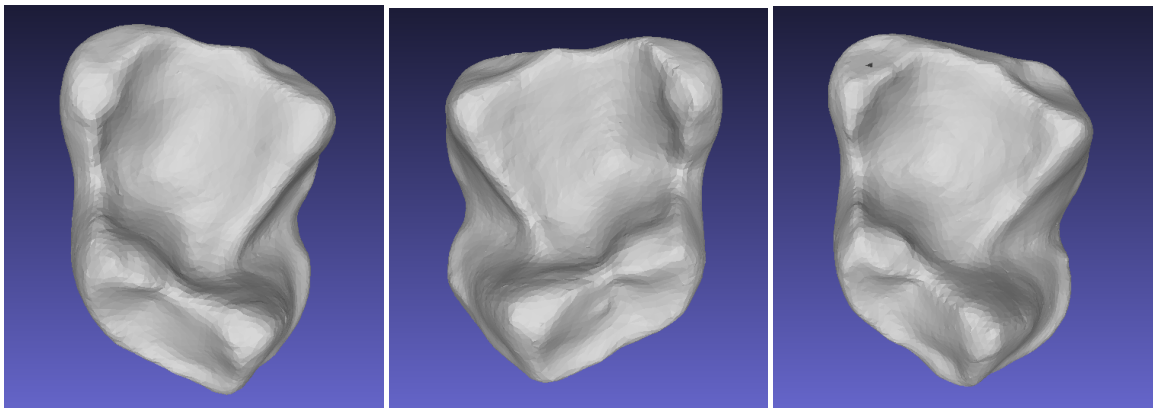
Figure 3.4: Groups 4,5,6 of 'a10_sas.off' and their representative EC curves.

3.2 Application in shape classification

In this section we exhibit an application of simplified representations in shape class prediction. We use logististic regression to build the classification model where the features are constructed from representative curves in level 1 simplified representations (In the rest of this chapter, simplified representation specifically refers to level 1 simplified representation). The parameters we use to generate simplified representations are $n = 1000$,$\kappa = 10$,$J = 5$,$\epsilon = 5$ and max iteration $= 5$.

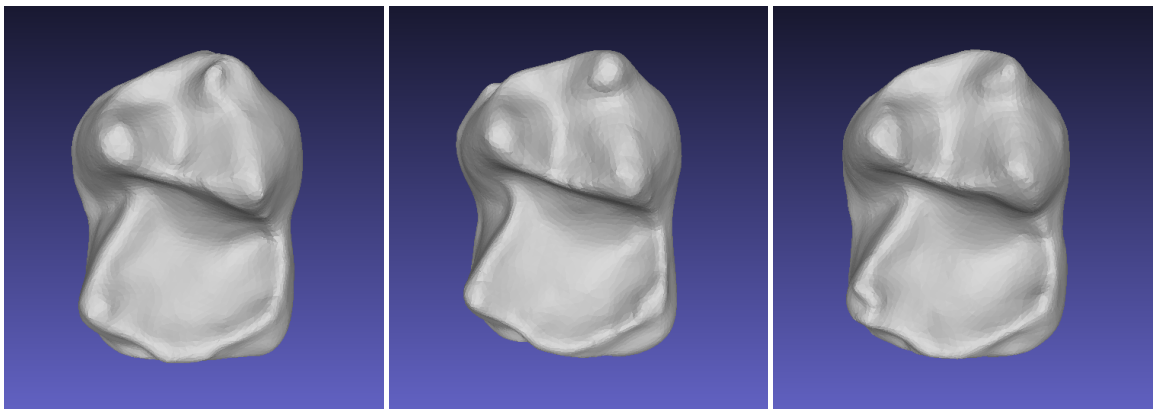
3.2.1 Data

The data we make use of are two classes (we call them class 0 and class 1) of teeth from the Lemuridae teeth set with 13 samples in class 0 and 12 samples in class 1. **Figure** 3.5 and **Figure** 3.6 show 3 typical samples respectively from each class. From the figures, these two classes are distinguishable because they have different numbers of protrusions and the locations of protrusions are characteristical.



(a) 'j01_sas.off' (b) 'j02_sas.off' (c) 'j06_sas.off'

Figure 3.5: 3 samples in class 0



(a) 'q02_sas.off' (b) 'q03_sas.off' (c) 'q04_sas.off'

Figure 3.6: 3 samples in class 1

3.2.2 Model

The purpose of this application is to find the typical curves for distinguishing two classes of shapes and examine their effectiveness in shape class prediction. Thus we need to establish a feature curves set $\mathcal{C}_{feature}$ at first and then discover typical curves from it. $\mathcal{C}_{feature}$ is composed of all representative curves from 6 simplified representations (3 shapes randomly selected from each class).

The next step is to build the classification model. We use the rest shapes to build a logistic regression model in order to discover typical curves of the two classes from $\mathcal{C}_{feature}$. The predictor vectors of the these shapes are the "similarity" between these shapes and the cruves in $\mathcal{C}_{feature}$.

Specifically, denote the curves in $\mathcal{C}_{feature}$ as $\{H_1, H_2, ..., H_p\}$, and the simplified representation of a shape M as $\mathcal{S}^1 = \{A_g\}$ and $\mathcal{C}^1 = \{F_g\}$. The predictor vector $(x_1, x_2, ..., x_p)$ of M is computed as

$$x_i = \sum_g S(A_g) * exp(-\kappa' * ||H_i - F_g||_2^2/n) \tag{3.1}$$

where $S(\cdot)$ denotes area and κ' is set to be 5.

Now, a classification model could be built by performing logististic regression on labels

and features. By inspecting the coefficients of the features, intuitively typical curves of the two classes could be discovered. However, two problems exist. First, because we select 3 shapes from each class, similar curves exist in $\mathcal{C}_{feature}$ and multicollinearity between the features presents. Second, the quantity of feature curves in $\mathcal{C}_{feature}$ could be large (ranges from 200 to 1000 in experiments). The two factors cause difficulty in identifying typical curves. In order to handle multicollinearity and do variable selection simultaneously, we build the model using lasso logististic regression [12, 13, 14] instead.

We use scikit-learn [15] to compute the lasso logististic regression model. In order to do lasso regression, a parameter C, inverse of regularization strength, should be clarified. We do the selection of C by cross validation. The candidate set of C is {1, 4, 16, 64, 256, 1024, 4096, 8192}. For each C, we calculate its expected accuracy by averaging the test accuracy on 100 trials, where in each trial we randomly select 3 shapes from each class to compose the feature curves set, 5 shapes from each class to compose the training set and 4 from each class to compose the test set. Additionally, the predictors are normalized for impartial variable selection via lasso. **Figure** 3.7 shows the result of cross validation. The overall accuracy is between 83% $\sim$ 89% and for $C >= 4$ the accuracy is above 88%. The result tells that $C = 4$ is an appropirate choice because it not only has high expected accuracy but also, due to high regularization strength, identifies fewer typical curves compared to larger C.

3.2.3 Examples of typical curves

Next we visualize the typical curves obtained from one trial. We run 100 trials with parameter $C = 4$ and randomly assigned feature curves set, train set and test set. The model we use in this section is selected from these trials and exihibited a 100% test accuracy.

Table 3.1 shows the coefficients of the typical curves obtained from the lasso logististic regression model and the original shape of these features. As expected, curves from class 0 shapes contributes to the likelihood of class 0 and samely for class 1. In **Figure** 3.8 we

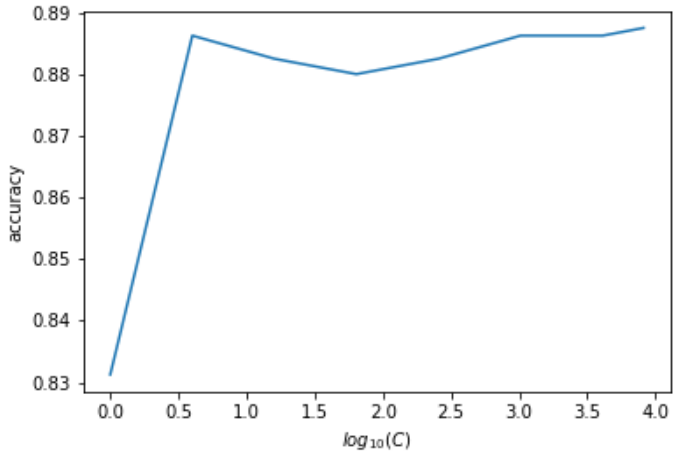


Figure 3.7: Cross validation accuracy

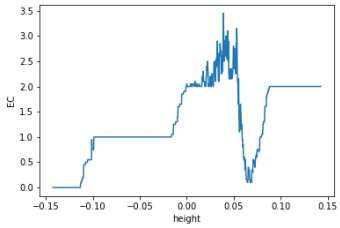
present these curves. **Figure** 3.8a to **Figure** 3.8d show the typical curves of class 0 and **Figure** 3.8e to **Figure** 3.8g show that of class 1.

Table 3.1: Coefficients of typical curves

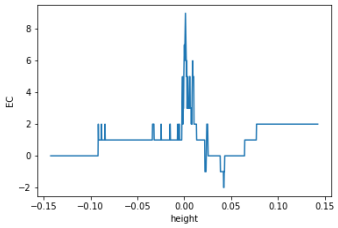
No	coefficient	class	file	Curve Plot
1	-1.072	0	j17_sas.off	**Figure** 3.8a
2	-0.073	0	j17_sas.off	**Figure** 3.8b
3	-0.497	0	j08_sas.off	**Figure** 3.8c
4	-0.261	0	j07_sas.off	**Figure** 3.8d
5	2.374	1	q04_sas.off	**Figure** 3.8e
6	0.644	1	q26_sas.off	**Figure** 3.8f
7	0.298	1	q26_sas.off	**Figure** 3.8g

Another visualization we present in **Figure** 3.9 illustrates the regions of original shapes that resulted in the jumps of typical curves. From the calculation of EC curves in **section** 1.1, EC curves jumps only at vertices. Figures in **Figure** 3.9 are drawn by coloring the

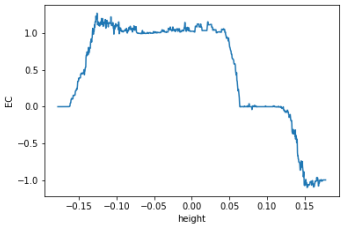
faces that contain vertices where typical curves jumps at. **Figure** 3.9a to **Figure** 3.9d show such regions of class 0 and **Figure** 3.9e to **Figure** 3.9g show that of class 1. From the figures, the typical curves except the one in **Figure** 3.9c mainly detect the protrusions and ditches. The typical curve in **Figure** 3.9c detects a large lateral region. However, according to its coefficient, the contribution of the curve is small compared to other curves.



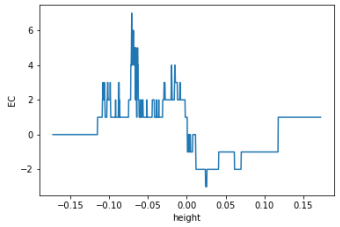
(a) Curve No: 1



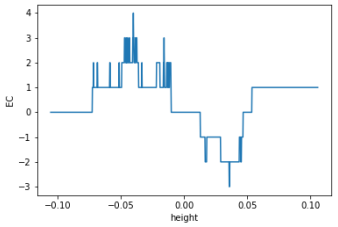
(b) Curve No: 2



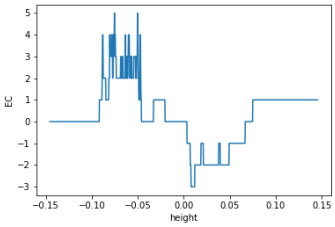
(c) Curve No: 3



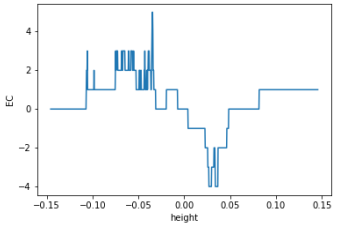
(d) Curve No: 4



(e) Curve No: 5

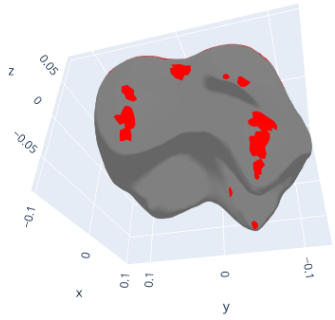


(f) Curve No: 6

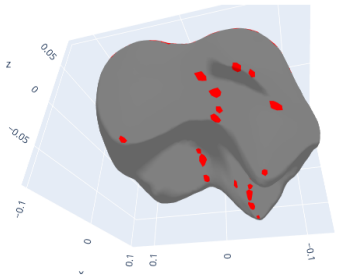


(g) Curve No: 7

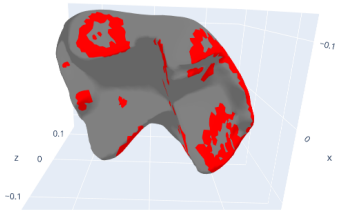
Figure 3.8: Typical curves.



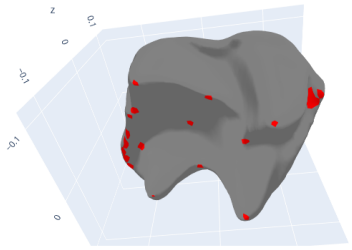
(a) Curve No: 1



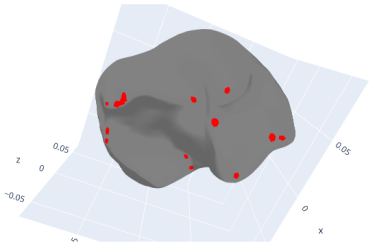
(b) Curve No: 2



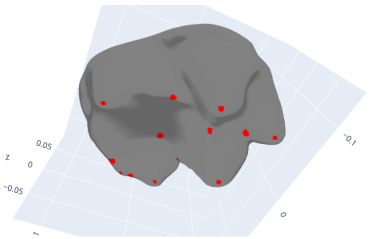
(c) Curve No: 3



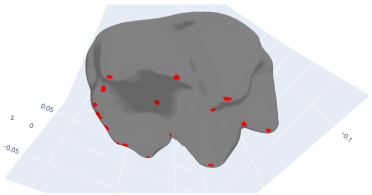
(d) Curve No: 4



(e) Curve No: 5



(f) Curve No: 6



(g) Curve No: 7

Figure 3.9: Original shape regions detected by typical curves.

Chapter 4

Conclusions

In this paper, we proposed a method to generate a specific form of 3D shape representation that could be applied in statistical analysis. The representation exploits Euler Characteristic curves and the computation of the representation is based on scaling function bases from diffusion wavelet. We described the details of our method and showed an application of the representation in a shape classification problem. In the application we demonstrated the ability of our method in discovering typical EC curves of a class of shapes and in predicting shape class.

There are some possible improvements in our work. First, we use sparse QR decomposition in computing scaling function bases. The choice originates from [3] because its computationally efficient. However, the purpose of the decomposition is to cluster the triangles, in which case some better choices exist. For example, spectral decomposition could replace QR decomposition in clustering the triangles, however it's computationally expensive especially when multi-scale analysis is required. Second, in the application of shape classification, we made use of only 25 shapes. After excluding the shapes for feature curves set and test set, only 10 shapes are used in training the regression model. As a result, the coefficients estimated are not accurate. A larger data set could help better understanding the performance of our method in shape classification.

Bibliography

[1] Katharine Turner, Sayan Mukherjee, and Doug M Boyer. Persistent homology transform for modeling shapes and surfaces. *Information and Inference: A Journal of the IMA*, 3(4):310–344, 2014.

[2] Bruce Wang, Timothy Sudijono, Henry Kirveslahti, Tingran Gao, Douglas M. Boyer, Sayan Mukherjee, and Lorin Crawford. A statistical pipeline for identifying physical features that differentiate classes of 3d shapes. *bioRxiv*, 2020.

[3] Ronald R Coifman and Mauro Maggioni. Diffusion wavelets. *Applied and Computational Harmonic Analysis*, 21(1):53–94, 2006.

[4] Justin Curry, Sayan Mukherjee, and Katharine Turner. How many directions determine a shape and other sufficiency results for two topological transforms. *arXiv preprint arXiv:1805.09782*, 2018.

[5] David Cohen-Steiner, Herbert Edelsbrunner, and John Harer. Stability of persistence diagrams. *Discrete & computational geometry*, 37(1):103–120, 2007.

[6] Katharine Turner, Yuriy Mileyko, Sayan Mukherjee, and John Harer. Fréchet means for distributions of persistence diagrams, 2013.

[7] Risi Kondor and John Lafferty. Diffusion kernels on graphs and other discrete input spaces. *ICML*, Vol. 2, 05 2002.

[8] Doug M. Boyer, Yaron Lipman, Elizabeth St. Clair, Jesus Puente, Biren A. Patel, Thomas Funkhouser, Jukka Jernvall, and Ingrid Daubechies. Algorithms to automatically quantify the geometric similarity of anatomical surfaces. *Proceedings of the National Academy of Sciences*, 108(45):18221–18226, 2011.

[9] Jeffrey H Schwartz and Ian Tattersall. *Evolutionary relationships of living lemurs and lorises (Mammalia, Primates) and their potential affinities with European Eocene Adapidae*, volume 60. American Museum of Natural History, 1985.

[10] Paolo Cignoni, Marco Callieri, Massimiliano Corsini, Matteo Dellepiane, Fabio Ganovelli, and Guido Ranzuglia. MeshLab: an Open-Source Mesh Processing Tool. In Vittorio Scarano, Rosario De Chiara, and Ugo Erra, editors, *Eurographics Italian Chapter Conference*. The Eurographics Association, 2008.

[11] Plotly Technologies Inc. Collaborative data science, 2015.

[12] Robert Tibshirani. The lasso method for variable selection in the cox model. *Statistics in medicine*, 16(4):385–395, 1997.

[13] Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of Statistical Software, Articles*, 33(1):1–22, 2010.

[14] Noah Simon, Jerome H. Friedman, Trevor Hastie, and Rob Tibshirani. Regularization paths for cox's proportional hazards model via coordinate descent. *Journal of Statistical Software, Articles*, 39(5):1–13, 2011.

[15] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.