Geometric Smoothing and Modeling of 3D Graphics

Ying Zhang

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science at

Concordia University

Montreal, Quebec, Canada

January 2007

Library and
Archives Canada

Published Heritage
Branch

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque et
Archives Canada

Direction du
Patrimoine de l'édition

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence
ISBN: 978-0-494-30079-4
Our file Notre référence
ISBN: 978-0-494-30079-4

NOTICE:
The author has granted a non-exclusive license allowing Library and Archives Canada to reproduce, publish, archive, preserve, conserve, communicate to the public by telecommunication or on the Internet, loan, distribute and sell theses worldwide, for commercial or non-commercial purposes, in microform, paper, electronic and/or any other formats.

The author retains copyright ownership and moral rights in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

AVIS:
L'auteur a accordé une licence non exclusive permettant à la Bibliothèque et Archives Canada de reproduire, publier, archiver, sauvegarder, conserver, transmettre au public par télécommunication ou par l'Internet, prêter, distribuer et vendre des thèses partout dans le monde, à des fins commerciales ou autres, sur support microforme, papier, électronique et/ou autres formats.

L'auteur conserve la propriété du droit d'auteur et des droits moraux qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

In compliance with the Canadian Privacy Act some supporting forms may have been removed from this thesis.

While these forms may be included in the document page count, their removal does not represent any loss of content from the thesis.

Conformément à la loi canadienne sur la protection de la vie privée, quelques formulaires secondaires ont été enlevés de cette thèse.

Bien que ces formulaires aient inclus dans la pagination, il n'y aura aucun contenu manquant.

Canada

Abstract

Geometric Smoothing and Modeling of 3D Graphics

Ying Zhang

The great challenge in computer graphics and geometric-aided design is to devise computationally efficient and optimal algorithms for estimating 3D models contaminated by noise and preserving their geometrical and topological structure. Motivated by the good performance of anisotropic diffusion in 2D image processing, we propose in this thesis a vertex-based nonlinear flow for 3D mesh smoothing by solving a discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the data. Illustrating experimental results demonstrate a much improved performance of the proposed approach in comparison with existing methods currently used in 3D mesh smoothing.

The major part of this thesis is devoted to a joint exploitation of geometry and topology to design new skeletal graph representation of 3D shapes in the Morse-theoretic framework. We present a multiresolution skeletal graph for topological coding of 3D shapes. The proposed skeletonization algorithm encodes a 3D object into a topological graph using a normalized mixture distance function. The approach is accurate, invariant to Euclidean transformations, computationally efficient, and preserves topology. Experimental results demonstrate the potential of the proposed topological graph which may be used as a shape signature for 3D object matching and retrieval, and also for skeletal animation.

Acknowledgements

I would like to express my deepest gratitude to my advisor, Dr. A. Ben Hamza, for his continuous support and encouragement throughout my graduate studies. His great help is essential to the completion of this thesis, more importantly, the challenging research that lies behind it.

I am also grateful to my colleagues Mohammadreza Ghaderpanah and Khaled Tarmissi for their friendship and valuable comments relating to this research.

At last, I would also like to thank the rest of the folks in our research lab for interesting discussions and being fun to work with.

Table of Contents

List of Figures

CHAPTER 1

Introduction

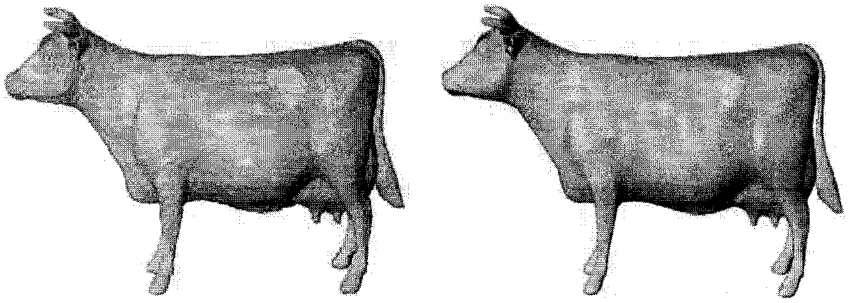
With the increasing use of 3D scanners to create 3D models, there is a rising need for robust mesh denoising to remove inevitable noise in the measurements. Even with high-fidelity scanners, the acquired models are invariably noisy, and therefore require filtering. In this thesis, we propose a nonlinear variational model for 3D mesh denoising by solving partial differential equations. Motivated by the outperformance, in tackling the 2D image denoising problem, of Laplacian smoothing by anisotropic diffusion, we propose in this thesis a vertex-based nonlinear flow for 3D mesh smoothing by solving a discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the data. Illustrating experimental results demonstrate a much improved performance of the proposed approach in comparison with existing methods currently used in 3D mesh smoothing.

The major part of this thesis is devoted to a joint exploitation of geometry and topology to design new skeletal graph representations of 3D shapes in the Morse-theoretic framework and their potential applications to 3D object recognition, matching, retrieval, and animation. The key idea consists of capturing geometry along all topologically homogeneous parts of an object by way of level curves superimposed on a skeletal graph referred to as *Reeb graph*, and it is usually extracted by way of the object critical points. A Reeb graph is a topological representation of the connectivity of a surface between critical points which represent the nodes of the graph, and the edges of the graph represent the connected components of the surface. For the example of using a height function, the Reeb graph would contain nodes for each of the contours for each level set generated by the height function. The resulting skeletal representation, however, is not rotationally invariant due to the rotational non-invariance of the height function. we propose an invariant Reeb graph for topology coding using the Morse distance function [30]. The key idea is to identify and encode regions of topological interest of a 3D object in the Morse-theoretic framework [30–32]. The proposed algorithm preserves efficiently the topology of 3D shapes, and it is robust, accurate, and has a low computational complexity.

1.1 Framework and motivation

1.1.1 Mesh denoising

Mesh denoising refers to the process of recovering a 3D model contaminated by noise. The challenge of the problem of interest lies in faithfully recovering the original model from the observed model, and furthering the estimation by making use of any prior knowledge/assumptions about the noise process. The problem of mesh denoising has been addressed using a number of different techniques including Laplacian flow [12], mean and median filtering [14] and bilateral filtering [16] which are all adopted from the image processing literature. Also, a number of anisotropic diffusion methods for triangle meshes and implicit surfaces have been proposed recently. Much of the appeal of partial differential equations-based methods lies in the availability of a vast arsenal of mathematical tools which at the very least act as a key guide in achieving numerical accuracy as well as stability.



Noisy 3D Cow Model Denoised 3D Cow Model

Figure 1.1: Illustration of 3D mesh denoising.

1.1.2 Object recognition

3D objects consist of geometric and topological data, and their compact representation is an important step towards a variety of computer graphics applications including indexing, retrieval, and matching in a database of 3D models. The latter will be the focus of Chapter

5, and the motivation behind considering 3D objects is illustrated in Figure 1.2.

- 2D provides the grayscale/color information in the plane: lost of depth information
- e.g.: 2D images of F-16s and MiG-23s look very similar, but in 3D are different
- 3D is much more effective for recognition and display
- 3D applications: industry, medicine, search, video games and cinema

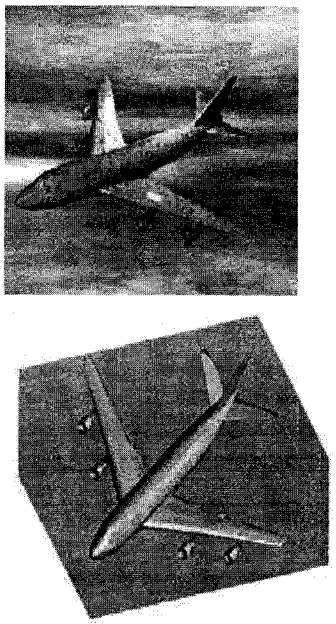


Figure 1.2: Motivation of 3D matching.

3D models do not depend on the configuration of cameras, light sources, or surrounding objects. As a result, they do not contain reflections, shadows, occlusions, projections, or partial objects, which in turn greatly simplifies finding matches between objects of the same type. For example, it is plausible to expect that the 3D model of a Boeing747 contains exactly four engines. In contrast, any 2D image of this Boeing747 may contain fewer than four engines (if some of the engines are occluded), or it may contain "extra engines" appearing as the result of shadows.

In other respects, representing and processing 3D models is more complicated than for sampled multimedia data. The main difficulty is that 3D surfaces rarely have simple parameterizations. Since 3D surfaces can have arbitrary topologies, many useful methods for analyzing other media (e.g., Fourier analysis) have no obvious analogs for 3D surface models. Moreover, the dimensionality is higher, and this makes searches for pose registration, feature correspondences, and model parameters more difficult.

In order to perform 3D recognition and to carry out the experiments, first we need a database of 3D models, and a small subset of our large database is depicted in Figure 1.3. We collected several hundred models which consist of military objects, human body parts, animals and other objects.

There are two major techniques for 3D object recognition: feature-based and global methods as depicted in Figure 1.4. Most three-dimensional shape matching techniques proposed in the literature of computer graphics, computer vision and computer-aided design are based on geometric representations which represent the features of an object in such a way that the shape dissimilarity problem reduces to the problem of comparing two such object representations. Feature-based methods require that features be extracted and described before two objects can be compared.

An alternative to feature-based representations is global methods. The idea here is to represent an object by a global measure or shape distribution defined on the surface of the object. The shape matching problem is then performed by computing a dissimilarity measure between the shape distributions of two arbitrary objects.

1.1.3 Joint exploitation of geometry and topology

Although topology is the study of the "shape" of curves and surfaces, topology typically is not concerned with the embedding of that curve or surface. For example, topology is

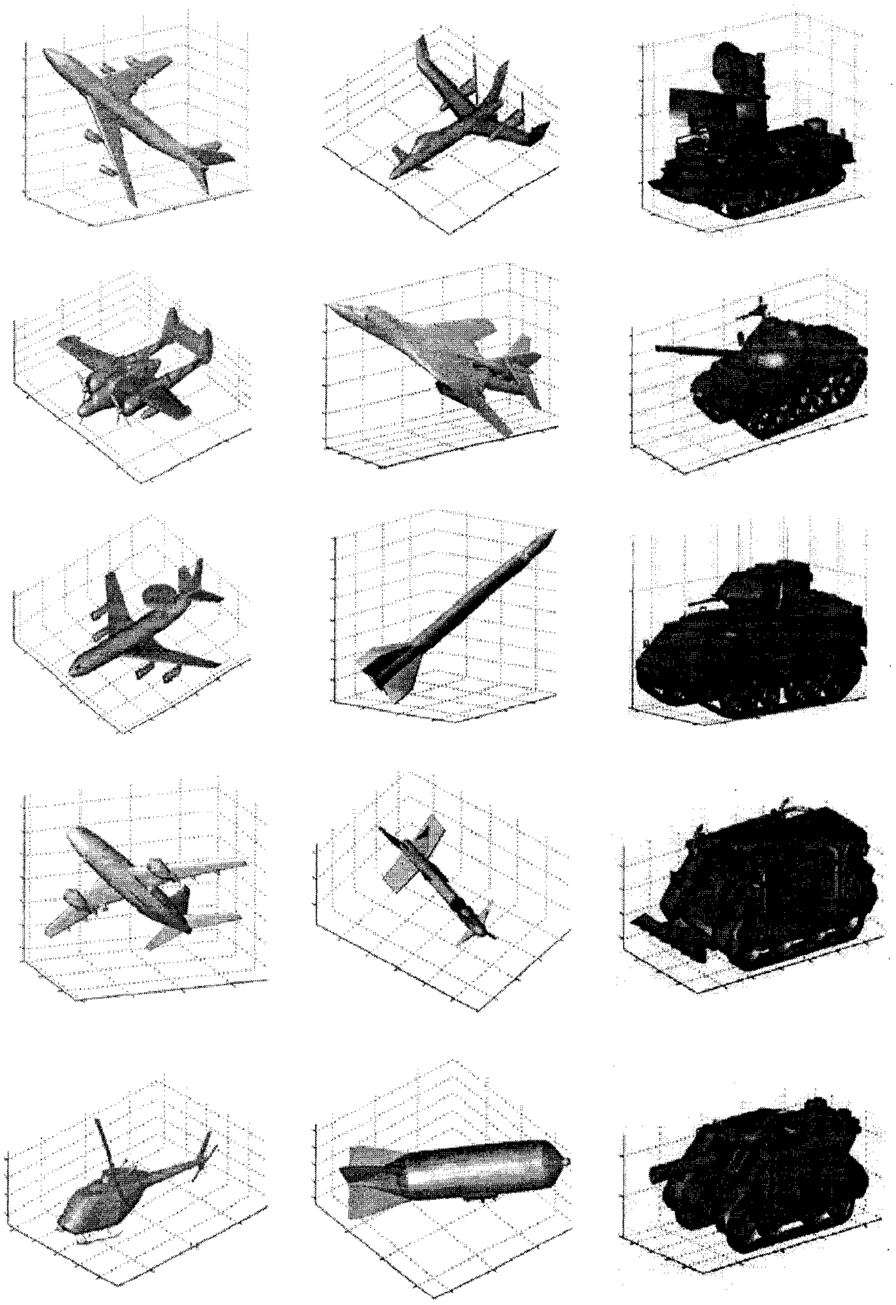


Figure 1.3: Examples of 3D models.

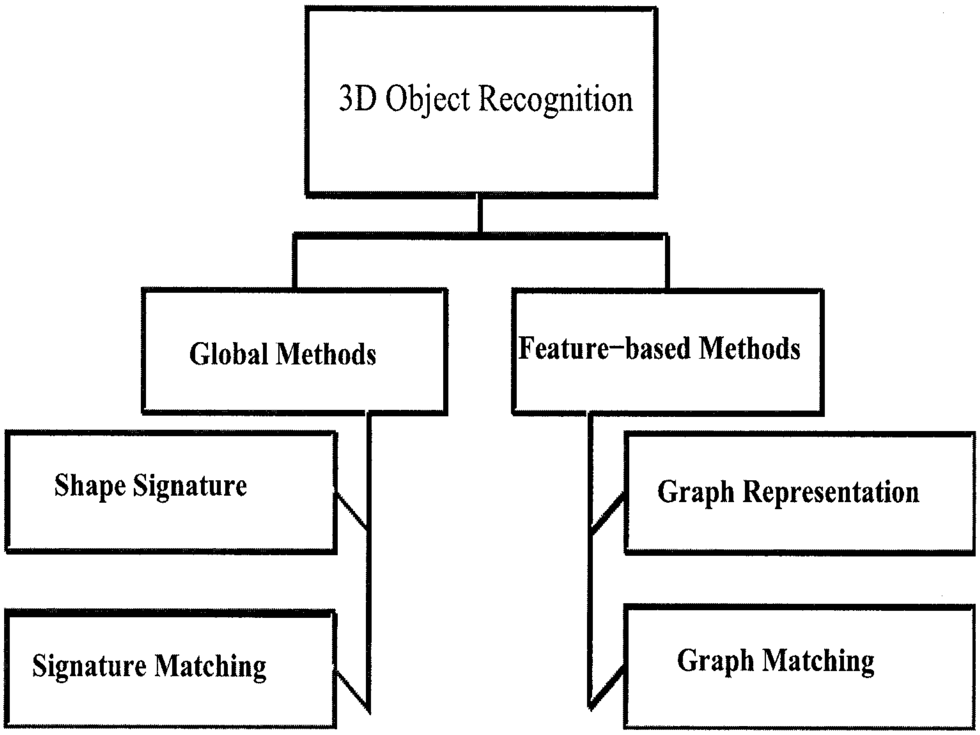


Figure 1.4: 3D object matching diagram.

concerned with the fact that if you remove a point from a circle, it becomes a line segment. This is true whether the circle is an ellipse or whether the circle has knots in it. In computer graphics, one cares about the embedding and geometry of a surface. If one were asked to create a digital representation of a coffee cup, no one would be happy if you returned a model that looked like a doughnut. Even though you have returned an object with the correct topological shape, the geometrical shape is incorrect. For computer graphics, we typically are not concerned with purely topological aspects of a surface. This interplay of geometry and topology is inherent in the discrete nature of the surfaces used in the field of computer graphics where great care is taken with the geometry of a surface, as the geometry plays such an important role in determining the appearance of a surface. Although a coffee cup is topologically equivalent to a doughnut, geometrically the shapes differ as shown in Figure 1.5. And the difference in their appearance matters greatly when the goal is to accurately represent the appearance of real world objects. Thus, a great deal of work in computer graphics has focused on geometric aspects of a surface, including geometry acquisition, geometry simplification, geometry smoothing and geometry compression. However, there is a direct relationship between the topology and the geometry of a surface that cannot be ignored. Alternatively, many mathematicians and computational topologists are concerned with studying purely topological properties of a surface. This thesis takes a combined approach and identifies and localizes topological features within a surface by mixing topological and geometrical approaches.

The connection between geometry and topology is given by a topological invariant called *genus.* The genus of a surface counts how many "handles" or "holes" the surface has, and two surfaces having the same genus are called *topologically equivalent.* Genus is a global invariant for the surface and its scalar value provides one measure of the complexity of the surface (e.g., a genus zero shape is much less complex than a surface with higher genus). For example, the coffee cup and doughnut shown in Figure 1.5 have the same genus equal to one.

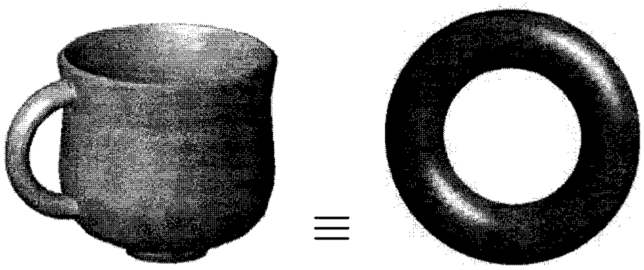


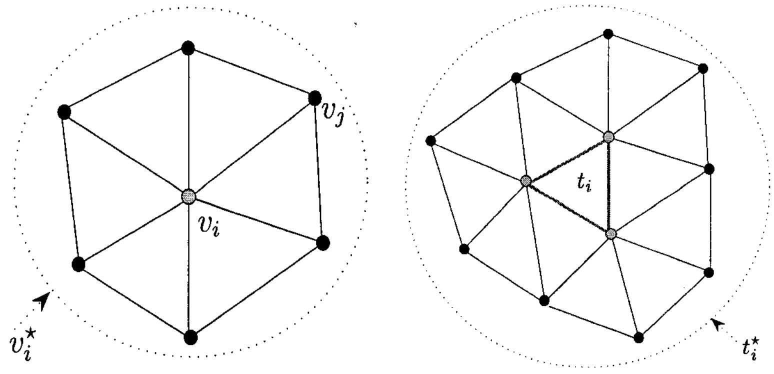
Figure 1.5: Topological equivalence of coffee cup and doughnut.

1.2 Background

This thesis addresses the application of computational geometry and topology algorithms to three-dimensional surfaces. The following background material is presented to provide context for this work.

1.2.1 Triangle mesh

In computer graphics and geometric-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, and $\mathcal{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ is the set of triangles. Each edge $e_{ij} = [\boldsymbol{v}_i, \boldsymbol{v}_j]$ connects a pair of vertices $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$. Two distinct vertices $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}$ are adjacent (written $\boldsymbol{v}_i \sim \boldsymbol{v}_j$) if they are connected by an edge, i.e. $e_{ij} \in \mathcal{E}$. The neighborhood (also referred to as a ring) of a vertex $\boldsymbol{v}_i$ is the set $\boldsymbol{v}_i^\star = \{\boldsymbol{v}_j \in \mathcal{V} : \boldsymbol{v}_i \sim \boldsymbol{v}_j\}$. The degree d_i of a vertex $\boldsymbol{v}_i$ is simply the cardinality of $\boldsymbol{v}_i^\star$. We denote by $\mathcal{T}(\boldsymbol{v}_i^\star)$ the set of triangles of the ring $\boldsymbol{v}_i^\star$, and by $\boldsymbol{t}_i^\star$ the set of all triangles sharing a vertex or an edge with a triangle $\boldsymbol{t}_i \in \mathcal{T}$ of a mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$. Fig. 2.1(a) depicts an example of a neighborhood $\boldsymbol{v}_i^\star$, where the degree of the vertex $\boldsymbol{v}_i$ is $d_i = 6$, and the number of triangles of the set $\mathcal{T}(\boldsymbol{v}_i^\star)$ is also equal to 6. An illustration of $\boldsymbol{t}_i^\star$ is provided in Fig. 2.1(b).



(a) (b)

Figure 1.6: (a) Vertex neighborhood $\boldsymbol{v}_i^\star$, (b) triangle neighborhood $\boldsymbol{t}_i^\star$.

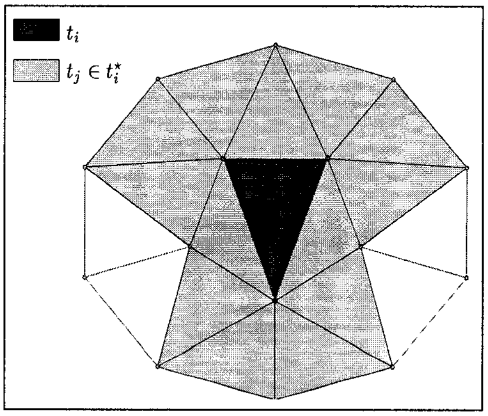


Figure 1.7: triangle neighborhood $\boldsymbol{t}_i^\star$.

Examples of triangle meshes are depicted in Figure 1.8. For triangulation, we use the barycentric subdivision illustrated in Figure 1.9. This technique consists in introducing a new vertex at the center of each triangle and a new vertex at the midpoint of each edge and drawing edges from the centroid of the triangle to each of the new midpoint vertices and to the original vertices.

Given a triangle $\boldsymbol{t}_j \in \mathcal{T}$, we denote by area($\boldsymbol{t}_j$) and $\boldsymbol{n}(\boldsymbol{t}_j)$ the area and the unit normal of $\boldsymbol{t}_j$ respectively. Consider a triangle $\boldsymbol{t}_j$ with vertices A, B and C, angles α, β and γ and sides a, b and c as illustrated in Fig. 1.10. The triangle normal $\boldsymbol{n}(\boldsymbol{t}_j)$ can be calculated as the vector cross product of two edges of the triangle, and a numerically stable Heron's formula for computing area($\boldsymbol{t}_j$) is given by

$$\text{area}(\boldsymbol{t}_j) = \frac{1}{4}\sqrt{(a+(b+c))(a+(b-c))(c+(a-b))(c-(a-b))},$$

where the length of the sides are arranged such that $a \geq b \geq c$.

The normal $\boldsymbol{n}_i$ at a vertex $\boldsymbol{v}_i$ is obtained by averaging the normals its neighboring triangles

$$\boldsymbol{n}_i = \frac{1}{d_i} \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} \boldsymbol{n}(\boldsymbol{t}_j).$$

Fig. 1.11 depicts the vertex normals of a triangle mesh.

The normal may be defined by weight-averaging the normals of the neighboring triangles (followed by a normalization step)

$$\boldsymbol{n}_i = \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} \omega_{ij}\, \boldsymbol{n}(\boldsymbol{t}_j).$$

where ω_{ij} is a normalized weight given e.g. by $1/d_i$, or by the angle formed by the edges of $\boldsymbol{t}_j$ incident to $\boldsymbol{v}_i$, or by the area of each triangle $\boldsymbol{t}_j$.

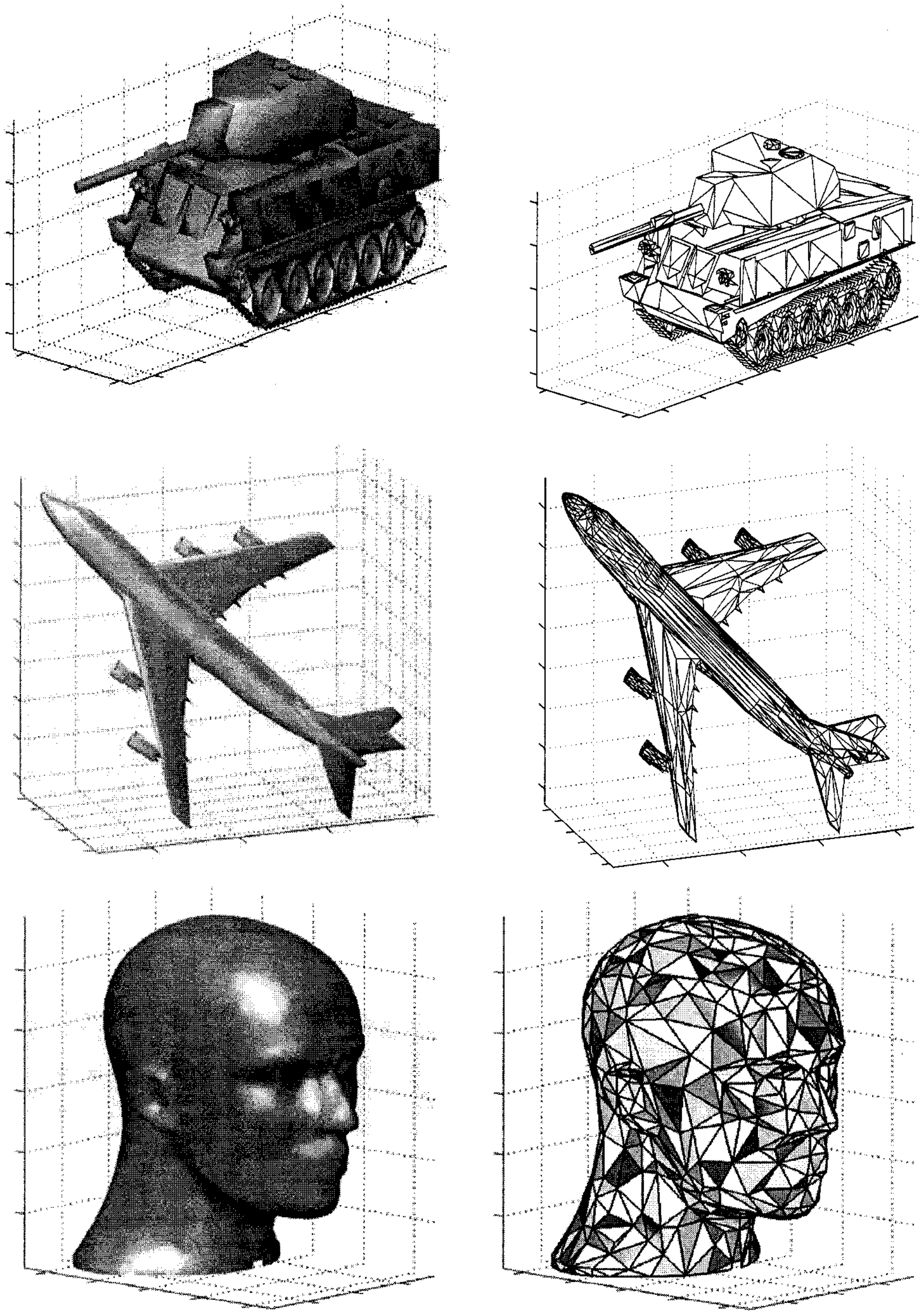


Figure 1.8: Triangle meshes.

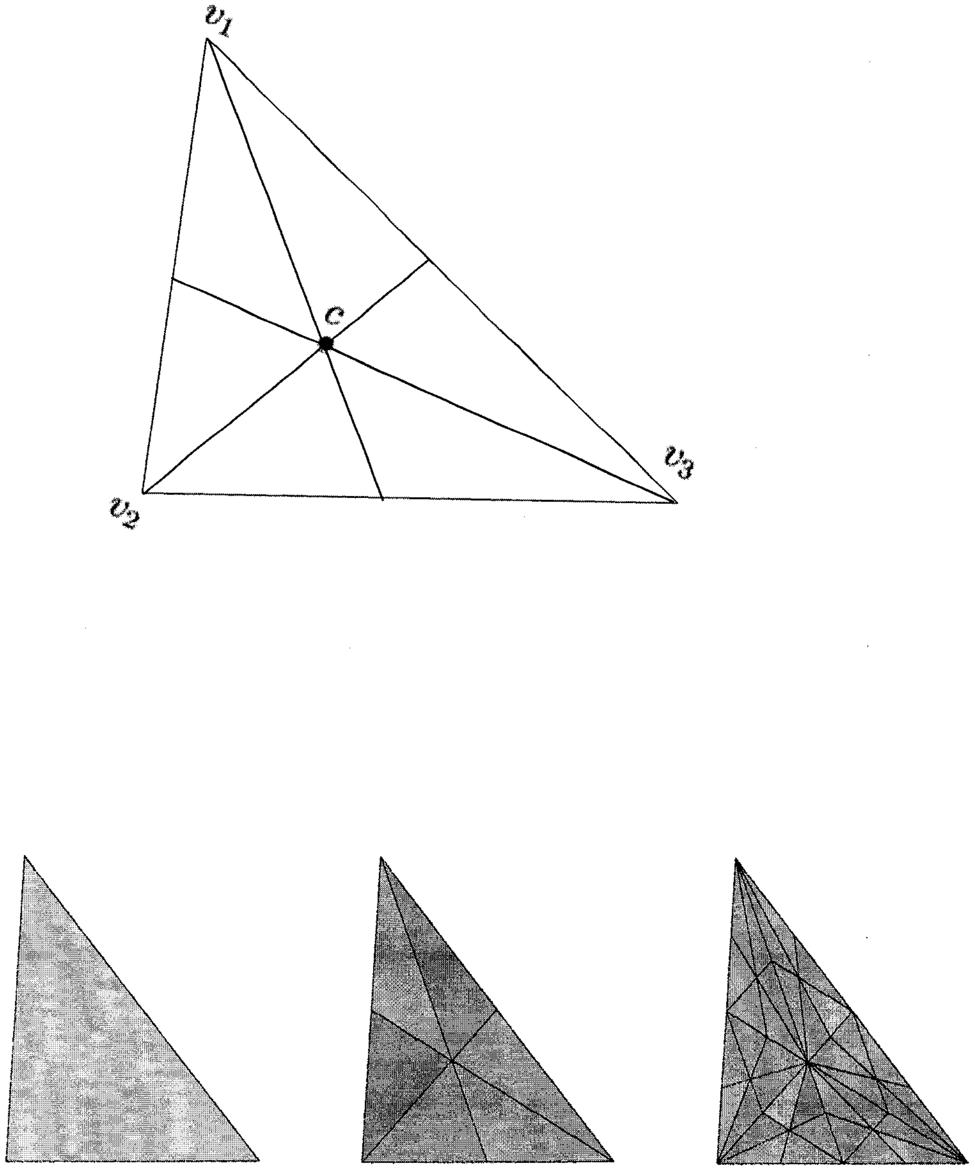


Figure 1.9: Barycentric triangulation.

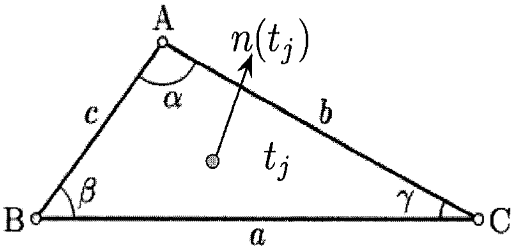


Figure 1.10: Illustration of $A(\boldsymbol{t}_j)$ and $\boldsymbol{n}(\boldsymbol{t}_j)$.

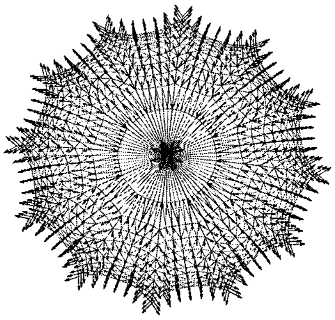


Figure 1.11: Illustration of the vertex normals.

1.2.2 Edge matrix of a triangle mesh

Given a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, the mean edge length $\bar{\ell}$ of the mesh is given by

$$\bar{\ell} = \frac{1}{|\mathcal{E}|} \sum_{e_{ij} \in \mathcal{E}} \|e_{ij}\|,$$

where $\|e_{ij}\| = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|$ if $\boldsymbol{v}_i \sim \boldsymbol{v}_j$, and $\|e_{ij}\| = 0$ otherwise.

The edge matrix of a triangle mesh is then given by

$$E = (e_{ij}) = \begin{cases} \|\boldsymbol{v}_i - \boldsymbol{v}_j\| & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{otherwise} \end{cases}$$

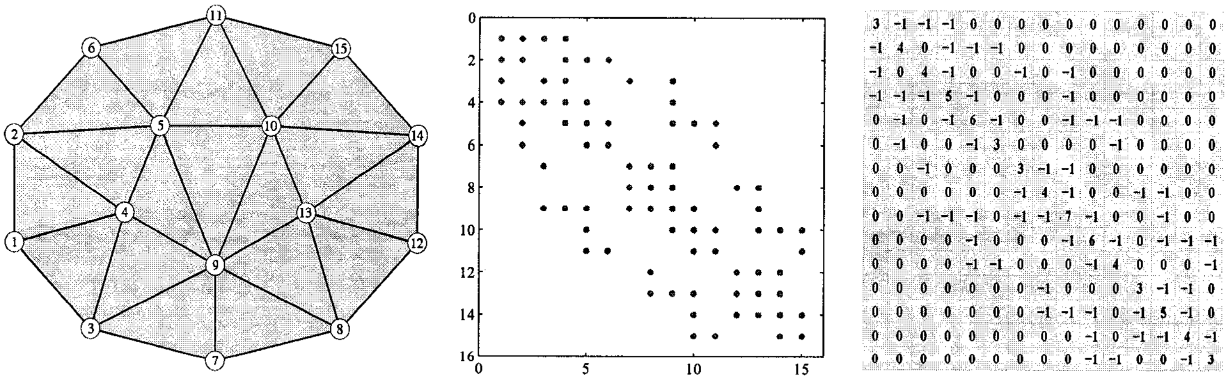
Spectral graph theory uses the spectra of matrices associated with the graph, such as the adjacency matrix, the Laplacian matrix, or the normalized Laplacian, to provide information about the graph. One goal is to characterize a graph or obtain information about the graph from the spectra of these matrices.

1.2.3 Normalized Laplacian matrix of a triangle mesh

The Laplacian matrix of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by $L = D - A$, where $A = (a_{ij})$ is the adjacency matrix between the vertices, that is $a_{ii} = 0$ and $a_{ij} = 1$ if $\boldsymbol{v}_i \sim \boldsymbol{v}_j$; and $D = \operatorname{diag}\{d_i : i = 1, \ldots, m\}$ is the degree matrix (diagonal matrix whose (i,i) entry is d_i). It is worth pointing out that the number of edges of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by $|\mathcal{E}| = \operatorname{trace}(D)/2 = \operatorname{trace}(A^2)/2$. The Euler characteristic is then given by $\chi(\mathbb{M}) = |\mathcal{V}| - |\mathcal{E}| + |\mathcal{T}|$.

The Laplacian matrix is defined as [25]

$$L = (\ell_{ij}) = \begin{cases} d_i & \text{if } \boldsymbol{v}_i = \boldsymbol{v}_j \\ -1 & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{o.w.} \end{cases}$$



3	-1	-1	-1	0	0	0	0	0	0	0	0	0	0	0
-1	4	0	-1	-1	-1	0	0	0	0	0	0	0	0	0
-1	0	4	-1	0	0	-1	0	-1	0	0	0	0	0	0
-1	-1	-1	5	-1	0	0	0	-1	0	0	0	0	0	0
0	-1	0	-1	6	-1	0	0	-1	-1	-1	0	0	0	0
0	-1	0	0	-1	3	0	0	0	0	-1	0	0	0	0
0	0	-1	0	0	0	3	-1	-1	0	0	0	0	0	0
0	0	0	0	0	0	-1	4	-1	0	0	-1	-1	0	0
0	0	-1	-1	-1	0	-1	-1	7	-1	0	0	-1	0	0
0	0	0	0	-1	0	0	0	-1	6	-1	0	-1	-1	-1
0	0	0	0	-1	-1	0	0	0	-1	4	0	0	0	-1
0	0	0	0	0	0	0	-1	0	0	0	3	-1	-1	0
0	0	0	0	0	0	0	-1	-1	-1	0	-1	5	-1	0
0	0	0	0	0	0	0	0	0	-1	0	-1	-1	4	-1
0	0	0	0	0	0	0	0	0	-1	-1	0	0	-1	3

Figure 1.12: 2D triangle mesh and its Laplacian matrix.

Fig. 1.13 illustrates an example of a 3D triangle mesh and its Laplacian matrix.

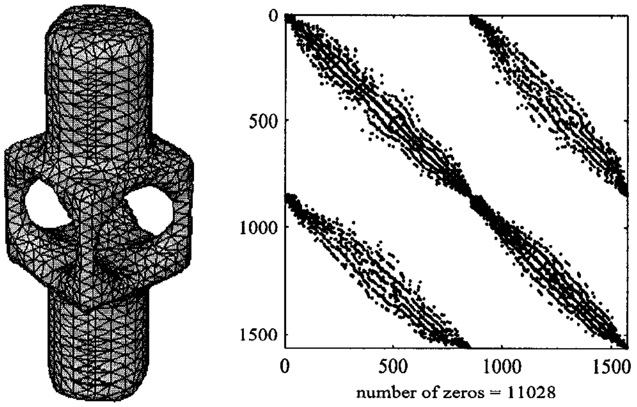


Figure 1.13: 3D triangle mesh and its Laplacian matrix.

The normalized Laplacian matrix $\mathcal{L}$ is given by

$$\mathcal{L} = D^{-1/2}LD^{-1/2} = I - D^{-1/2}AD^{-1/2} = \begin{cases} 1 & \text{if } \boldsymbol{v}_i = \boldsymbol{v}_j \\ -\frac{1}{\sqrt{d_i d_j}} & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{o.w.} \end{cases}$$

and may be viewed as an operator defined on the space of functions $\varphi : \mathcal{V} \longrightarrow \mathbb{R}$ as follows

$$\mathcal{L}\varphi(\boldsymbol{v}_i) = \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\varphi(\boldsymbol{v}_i)}{\sqrt{d_j}} - \frac{\varphi(\boldsymbol{v}_j)}{\sqrt{d_i}} \right), \quad \forall \boldsymbol{v}_i \in \mathcal{V}.$$

The eigenvalue spectrum of the normalized Laplacian matrix allows us to compare the structure of graphs of different sizes.

1.3 Affine invariant distance function

Let $\boldsymbol{c} = (\bar{x}, \bar{y}, \bar{z})^T$ be the centroid of the triangle mesh, and define the centered vertex matrix V_c as

$$V_c = (\boldsymbol{v}_1 - \boldsymbol{c} \quad \boldsymbol{v}_2 - \boldsymbol{c} \quad \dots \quad \boldsymbol{v}_m - \boldsymbol{c})^T = \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ x_2 - \bar{x} & y_2 - \bar{y} & z_2 - \bar{z} \\ \vdots & \vdots & \vdots \\ x_m - \bar{x} & y_m - \bar{y} & z_m - \bar{z} \end{pmatrix}$$

Let $A = m(V_c^T V_c)^{-1}$, it can be shown that the squared norm defined by

$$\|\boldsymbol{v}\|_A^2 = \boldsymbol{v}^T A \boldsymbol{v}$$

is invariant under affine transformations [37]. To achieve affine invariant distance function we simply replace the Euclidean norm $\|\boldsymbol{v}\|$ with the affine invariant norm $\|\boldsymbol{v}\|_A$.

1.4 Contributions

The contributions of this thesis are as follows:

- ☞ **Vertex-based anisotropic mesh denoising:** We introduced a vertex-based anisotropic diffusion for 3D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the 3D mesh data. The experimental results clearly showed a much improved performance of the proposed approach in comparison with the current methods used in 3D mesh smoothing. For future work, we plan to incorporate the curvature information as well as additional regularization terms into the proposed model.

☞ **Statistical 3D mesh distributions:** We proposed several statistical measures to analyze the topological properties of 3D models. The proposed statistical measures include the mesh degree, the mesh assortativity, the mesh clustering coefficient, and the mesh geodesic distance distributions. The experimental results clearly showed the effectiveness of the proposed measures in quantifying the topological features of 3D objects.

☞ **Euclidean distance-based skeletal graph:** We proposed a distance function-based approach to topological coding of 3D objects in the Morse-theoretic framework. Using a distance function, we constructed invariant Reeb graphs of 3D objects. The main attractive properties of the proposed approach are: invariance to rotation, translation, and scaling; robustness to decimation, and low computational complexity. Future work will be focused on using the distance function based Reeb graph as a shape signature for 3D object matching, as well as the reconstruction of objects from the level sets of the distance function.

☞ **Multiresolution mixture skeletal graph:** We introduced a mixture distance function-based approach to topological coding of 3D objects in the Morse-theoretic framework. The proposed algorithm preserves efficiently the topology of 3D shapes, and it is robust, accurate, and has a low computational complexity. The main attractive properties of the proposed multiresolution approach are: invariance to rotation, translation, and scaling; and robustness to mesh decimation. We illustrated the approach with several examples of skeletal graphs for a variety of 3D objects. Future work will be focused on using this skeletal graph as a shape signature for 3D object matching and retrieval, and also for skeletal animation. Theoretically we hope to develop more rigorous way of finding the optimal resolution parameter of the proposed algorithm.

1.5 Thesis overview

The organization of this thesis is as follows:

- The first Chapter contains a brief review of essential concepts and definitions which we will refer to throughout the thesis, and presents a short summary of material relevant to 3D mesh denoising methods, geometric modeling and computational topology.
- In Chapter 2, we present a vertex-based diffusion for 3D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the data. Illustrating experimental results demonstrate a much improved performance of the proposed approach in comparison with existing methods currently used in 3D mesh smoothing.
- In Chapter 3, we introduce several statistical distributions to analyze the topological properties of 3D graphics. The proposed statistical measures include the mesh degree, the mesh assortativity, the mesh clustering coefficient, and the mesh geodesic

distance distributions. These probabilistic distributions provide useful information about the way 3D mesh models are connected. Illustrating experimental results show the effectiveness of the proposed measures in quantifying the topological features of 3D objects.

☐ In Chapter 4, we present an invariant shape descriptor for topological coding of 3D objects. The proposed approach encodes a 3D object into a topological graph using a normalized distance function. Unlike the height function which has been traditionally used to model topology, the proposed distance function-based graph is invariant to rigid motion transformations. Simulations results demonstrate the potential of the proposed topological graph which may be used as a shape signature for object matching and reconstruction.

☐ In Chapter 5, we present a multiresolution skeletal graph for topological coding of 3D shapes. The proposed skeletonization algorithm encodes a 3D object into a topological graph using a normalized mixture distance function. The approach is accurate, invariant to Euclidean transformations, robust to mesh decimation, computationally efficient, and preserves topology. Experimental results demonstrate the potential of the proposed topological graph which may be used as a shape signature for 3D object matching and retrieval, and also for skeletal animation.

☐ In the **Conclusions** Chapter, we summarize the contributions of this thesis, and we propose several future research directions that are directly or indirectly related to the work performed in this thesis.

1.6 Publications

- Y. Zhang and A. Ben Hamza, "Vertex-based Diffusion for 3D Mesh Denoising," *IEEE Transactions on Image Processing*, to appear 2007.
- Y. Zhang and A. Ben Hamza, "PDE-based smoothing for 3D mesh quality improvement," *Proc. IEEE International Conference on Electro/Information Technology*, Michigan, USA, 2006.
- Y. Zhang and A. Ben Hamza, "Vertex-based anisotropic smoothing of 3D mesh data," *Proc. IEEE Canadian Conference on Electrical and Computer Engineering*, Ottawa, Canada, 2006.
- M. Aguilera, Y. Zhang, and A. Ben Hamza, "Topological shape signature for 3D object modelling," *Proc. 17th IASTED International Conference on Modelling and Simulation*, Montréal, Canada, 2006.
- M. Qasaimeh, Y. Zhang, K. Tarmissi, and A. Ben Hamza, "Statistical mesh distributions for 3D object topology," *Proc. IEEE International Symposium on Signal Processing and its Applications*, Sharjah, UAE, 2007.
- Y. Zhang and A. Ben Hamza, "Multiresolution Topological Skeletal Graph for 3D Shapes," under review, December 2006.

CHAPTER 2

Anisotropic 3D Mesh Denoising

We present a vertex-based diffusion for 3D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the data. Illustrating experimental results demonstrate a much improved performance of the proposed approach in comparison with existing methods currently used in 3D mesh smoothing.

2.1 Introduction

The great challenge in image processing and computer graphics is to devise computationally efficient and optimal algorithms for recovering images and 3D models contaminated by noise and preserving their geometrical structure. With the increasing use of scanners to create 3D models which are usually represented as triangle meshes in computer graphics and geometric-aided design, there is a rising need for robust and efficient 3D mesh denoising techniques to remove undesirable noise from the data.

In recent years, various partial differential equations (PDE)-based methods have been proposed to tackle the problem of 2-D image denoising with a good preservation of features [1–10]. Much of the appeal of PDE-based methods lies in the availability of a vast arsenal of mathematical tools which at the very least act as a key guide in achieving numerical accuracy as well as stability. Partial differential equations or gradient descent flows are generally a result of variational problems [11]. The 3D mesh denoising problem, however, has received much less attention [12, 13]. The most commonly used mesh smoothing method is Laplacian flow which repeatedly and simultaneously adjusts the location of each mesh vertex to the geometric center of its neighboring vertices [12]. Although the Laplacian smoothing flow is simple and fast, it produces, however, the shrinking effect and an over-smoothing result. The most recent mesh denoising techniques include the mean, median, and bilateral filters [14–16] which are all adopted from the image processing literature. Also, a number of anisotropic diffusion methods for triangle meshes and implicit surfaces have been proposed recently. Desbrun *et al.* [17, 18] introduce a weighted Laplacian smoothing technique by choosing new edge weights based on curvature flow operators. This denoising method avoids the undesirable edge equalization from Laplacian flow and helps to preserve curvature for constant curvature areas. However, re-computing new edge weights after each iteration results in more expensive computational cost. Clarenz *et al.* [19] propose a multi-scale surface smoothing method based on the anisotropic curvature evolution problem. By discretizing nonlinear partial differential equations, this method aims to detect and preserve sharp edges by two user defined parameters which are a regularization parameter for filtering out high frequency noisy and a threshold for edge detection. This multiscale method was also extended to the texture mapped surfaces [20] in order to enhance edge type features of the texture maps. Different regularization parameters and edge detection threshold values, however, need to be defined by users onto noisy surfaces and textures respectively before the smoothing process. Bajaj *et al.* [21] present a unified anisotropic diffusion for 3D mesh smoothing by treating discrete surface data as a discretized version of a 2-D Riemannian manifold and establishing a PDE diffusion model for such a manifold. This method helps enhancing sharp features while filtering out noise by considering 3-ring neighbors of each vertex to achieve non-linear approach of smoothing process. Tasdizen *et al.* [22, 23] introduce a two-step surface smoothing method by solving a set of coupled second-order PDEs on level set surface models. Instead of filtering the positions of points on a mesh, this method operates on the normal map of a surface and manipulates the surface to fit the processed normals. All the surfaces normals are processed by solving second-order equations using implicit surfaces. In [24], Hildebrandt *et al.* present a mesh smoothing method by using a prescribed mean curvature flow for simplicial surfaces. This method develops an improved anisotropic diffusion algorithm by defining a discrete shape operator and principal curvatures of simplicial surfaces.

Motivated by the outperformance, in tackling the 2-D image denoising problem, of Laplacian smoothing by anisotropic diffusion [1, 2, 5], we propose in this chapter a vertex-based nonlinear flow for 3D mesh smoothing by solving a discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the data.

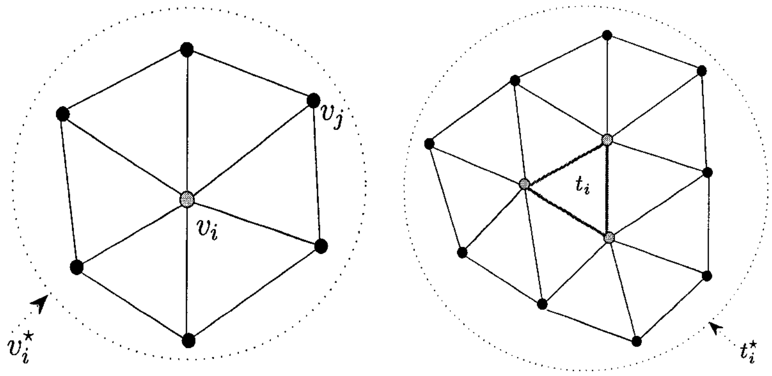
The rest of this chapter is organized as follows. In the next section, we briefly recall some basic concepts of 3D mesh data, and we introduce the vertex differential operators, then a general formulation of 3D mesh smoothing problem is stated. In Section 2.3, we briefly review some recent 3D mesh denoising techniques that are closely related to our proposed approach. In Section 2.4, a vertex-based nonlinear diffusion for 3D mesh smoothing is introduced. In Section 2.5, we provide experimental results to demonstrate a much improved performance of the proposed method in 3D mesh smoothing. Finally, some conclusions are included in Section 2.6.

2.2 Problem formulation

In computer graphics and geometric-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges with cardinality $|\mathcal{E}|$, and $\mathcal{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ is the set of triangles. Each edge $e_{ij} = [\boldsymbol{v}_i, \boldsymbol{v}_j]$ connects a pair of vertices $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$. Two distinct vertices $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}$ are adjacent (written $\boldsymbol{v}_i \sim \boldsymbol{v}_j$) if they are connected by an edge $e_{ij} \in \mathcal{E}$. The neighborhood (also referred to as a ring) of a vertex $\boldsymbol{v}_i$ is the set $\boldsymbol{v}_i^\star = \{\boldsymbol{v}_j \in \mathcal{V} : \boldsymbol{v}_i \sim \boldsymbol{v}_j\}$. The degree d_i of a vertex $\boldsymbol{v}_i$ is simply the cardinality of $\boldsymbol{v}_i^\star$. We denote by $\mathcal{T}(\boldsymbol{v}_i^\star)$ the set of triangles of the ring $\boldsymbol{v}_i^\star$, and by $\boldsymbol{t}_i^\star$ the set of all triangles sharing a vertex or an edge with a triangle $\boldsymbol{t}_i \in \mathcal{T}$ of a mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$. Fig. 2.1(a) depicts an example of a neighborhood $\boldsymbol{v}_i^\star$, where the degree of the vertex $\boldsymbol{v}_i$ is $d_i = 6$, and the number of triangles of the set $\mathcal{T}(\boldsymbol{v}_i^\star)$ is also equal to 6. An illustration of $\boldsymbol{t}_i^\star$ is provided in Fig. 2.1(b).

Given a triangle $\boldsymbol{t}_j \in \boldsymbol{t}_i^\star$, we denote by $A(\boldsymbol{t}_j)$ and $\boldsymbol{n}(\boldsymbol{t}_j)$ the area and the unit normal of $\boldsymbol{t}_j$ respectively. The normal $\boldsymbol{n}_i$ at a vertex $\boldsymbol{v}_i$ is obtained by averaging the normals of its neighboring triangles and is given by

$$\boldsymbol{n}_i = \frac{1}{d_i} \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} \boldsymbol{n}(\boldsymbol{t}_j). \tag{1}$$



(a) (b)

Figure 2.1: (a) Vertex neighborhood $\boldsymbol{v}_i^\star$, (b) triangle neighborhood $\boldsymbol{t}_i^\star$.

The mean edge length $\bar{\ell}$ of the mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by

$$\bar{\ell} = \frac{1}{|\mathcal{E}|} \sum_{e_{ij} \in \mathcal{E}} \|e_{ij}\|, \tag{2}$$

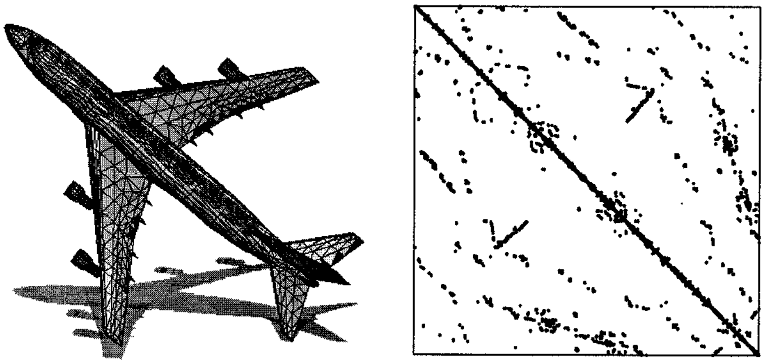
where $\|e_{ij}\| = \|\boldsymbol{v}_i - \boldsymbol{v}_j\|$ if $\boldsymbol{v}_i \sim \boldsymbol{v}_j$, and $\|e_{ij}\| = 0$ otherwise.

2.2.1 Laplacian matrix of a triangle mesh

The Laplacian matrix of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by $L = D - A$, where $A = (a_{ij})$ is the adjacency matrix between the vertices, that is $a_{ii} = 0$ and $a_{ij} = 1$ if $\boldsymbol{v}_i \sim \boldsymbol{v}_j$; and $D = \mathrm{diag}\{d_i : i \in \mathcal{V}\}$ is the degree matrix (diagonal matrix whose (i,i) entry is d_i). The Laplacian matrix is defined as [25]

$$L = (\ell_{ij}) = \begin{cases} d_i & \text{if } \boldsymbol{v}_i = \boldsymbol{v}_j \\ -1 & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{o.w.} \end{cases}$$

Fig. 2.2 illustrates an example of a 3D triangle mesh and its Laplacian matrix.



(a) (b)

Figure 2.2: (a) 3D triangle mesh and (b) its Laplacian matrix.

The normalized Laplacian matrix $\mathcal{L}$ is given by [25]

$$\mathcal{L} = D^{-1/2} L D^{-1/2},$$

and may be viewed as an operator defined on the space of functions $\varphi : \mathcal{V} \to \mathbb{R}$ as follows

$$\mathcal{L}\varphi(\boldsymbol{v}_i) = -\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\varphi(\boldsymbol{v}_j)}{\sqrt{d_j}} - \frac{\varphi(\boldsymbol{v}_i)}{\sqrt{d_i}} \right), \quad \forall \boldsymbol{v}_i \in \mathcal{V}.$$

2.2.2 Vertex differential operators

Given a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, we define the vertex gradient operator $\nabla \boldsymbol{v}_i$ and its magnitude as

$$\nabla \boldsymbol{v}_i = \left\{ \frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} : \boldsymbol{v}_j \in \boldsymbol{v}_i^\star \right\}.$$

We also define the vertex Laplace operator as

$$\operatorname{div}(\nabla \boldsymbol{v}_i) = \Delta \boldsymbol{v}_i = \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right),$$

where $\operatorname{div}(\cdot)$ denotes the divergence operator. Note the analogy between the vertex Laplace operator and the normalized Laplacian matrix $\mathcal{L}$ defined as an operator.

2.2.3 Mesh smoothing model

In all real applications, measurements are perturbed by noise. In the course of acquiring, transmitting or processing a 3D model for example, the noise-induced degradation may be dependent or independent of data. The noise is usually described by its probabilistic model. Application-dependent, a degradation often yields a resulting vertex observation model, and the most commonly used is the additive one,

$$\boldsymbol{v} = \boldsymbol{u} + \boldsymbol{\eta}, \tag{3}$$

where the observed vertex $\boldsymbol{v}$ includes the original vertex $\boldsymbol{u}$, and the random noise process $\boldsymbol{\eta}$ which is usually assumed to be Gaussian with zero mean and standard deviation σ.

Mesh smoothing refers to the process of recovering a 3D model contaminated by noise. The challenge of the problem of interest lies in recovering the underlying vertex $\boldsymbol{u}$ from the observed vertex $\boldsymbol{v}$, and furthering the estimation by making use of any prior knowledge/assumptions about the noise process $\boldsymbol{\eta}$.

The PDE-based smoothing approach is commonly formulated in a continuous domain which enjoys a large arsenal of analytical tools, and hence offers a greater flexibility. Laplacian smoothing is the most commonly used mesh smoothing method which repeatedly and simultaneously adjusts the location of each mesh vertex to the geometric center of its neighboring vertices using the following update rule

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \left(\frac{\boldsymbol{v}_j - \boldsymbol{v}_i}{d_i} \right). \tag{4}$$

It is worth pointing out that the Laplacian flow given by (4) is the discrete form of the isotropic heat equation $\boldsymbol{v}_t = \Delta \boldsymbol{v}$ applied to each vertex of the triangle mesh, where we assume that all vertices have the same degree.

Although the Laplacian smoothing flow is simple and fast, it tends, however, to produce a shrinking effect and an oversmoothing result. Motivated by the good performance of anisotropic diffusion, we propose in Section 2.4 a vertex-based flow defined by a nonlinear partial differential equation.

2.3 Related work

In this section, we will review some representative methods for 3D mesh smoothing that are closely related to our proposed approach, and we briefly show their mathematical foundations and algorithmic methodologies as well as their limitations.

2.3.1 Mean filter for averaging face normals

The mean filter procedure is depicted in Fig. 2.3 and is applied in three successive steps [14]:

Step 1 : compute the area weighted average face normal $\mathbf{m}(\boldsymbol{t}_i)$ for each mesh triangle $\boldsymbol{t}_i$:

$$\boldsymbol{m}(\boldsymbol{t}_i) = \frac{1}{\sum\limits_{\boldsymbol{t}_j \in \boldsymbol{t}_i^\star} A(\boldsymbol{t}_j)} \sum_{\boldsymbol{t}_j \in \boldsymbol{t}_i^\star} A(\boldsymbol{t}_j)\boldsymbol{n}(\boldsymbol{t}_j). \tag{5}$$

Step 2 : normalize the averaged normal $\boldsymbol{m}(\boldsymbol{t}_i)$:

$$\boldsymbol{m}(\boldsymbol{t}_i) \leftarrow \frac{\boldsymbol{m}(\boldsymbol{t}_i)}{\|\boldsymbol{m}(\boldsymbol{t}_i)\|}$$

Step 3 : update each vertex $\boldsymbol{v}_i$ in the mesh as follows:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \frac{1}{\sum\limits_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} A(\boldsymbol{t}_j)} \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} A(\boldsymbol{t}_j)\boldsymbol{\pi}_i(\boldsymbol{t}_j)$$

where $\boldsymbol{\pi}_i(\boldsymbol{t}_j) = \langle \boldsymbol{e}_{ij}, \boldsymbol{m}(\boldsymbol{t}_j)\rangle\, \boldsymbol{m}(\boldsymbol{t}_j)$, and $\boldsymbol{e}_{ij} = \boldsymbol{c}_j - \boldsymbol{v}_i$ is the vector from vertex $\boldsymbol{v}_i$ to the centroid $\boldsymbol{c}_j$ of the triangle $\boldsymbol{t}_j$. Note that by definition of the inner product, the vector $\boldsymbol{\pi}_i(\boldsymbol{t}_j)$ is the projection of the vector $\boldsymbol{e}_{ij}$ onto the direction of the normal $\boldsymbol{t}_j$.

2.3.2 Angle median filtering for face normals

For each triangle $\boldsymbol{t}_i \in \mathcal{T}$, denote by $\Theta_i = \{\theta_{ij} = \angle(\boldsymbol{n}(\boldsymbol{t}_i), \boldsymbol{n}(\boldsymbol{t}_j)) \;:\; \boldsymbol{t}_j \in \boldsymbol{t}_i^\star\}$ the set of angles between $\boldsymbol{n}(\boldsymbol{t}_i)$ and $\boldsymbol{n}(\boldsymbol{t}_j)$, where $\boldsymbol{n}(\boldsymbol{t}_i)$ is the normal of $\boldsymbol{t}_i$ and $\boldsymbol{n}(\boldsymbol{t}_j)$ is the normal of $\boldsymbol{t}_j$. As illustrated in Fig. 2.4, instead of computing the average face normal in Step 1 of the mean filter, in the angle median filtering method we first compute the median

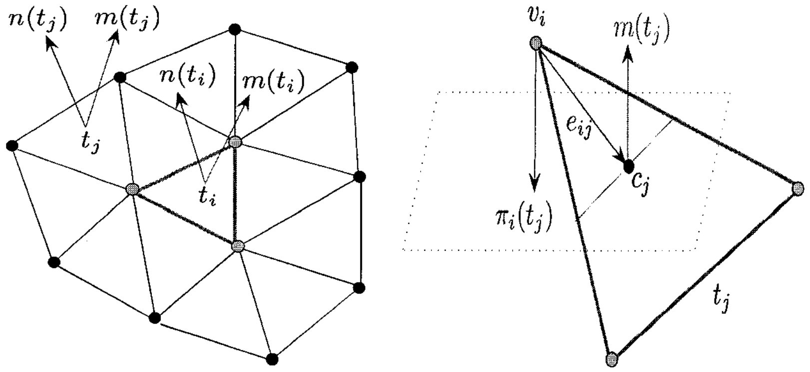


Figure 2.3: Illustration of the mesh mean filter algorithm.

angle $\hat{\theta}_i = \text{median}(\Theta_i) = \angle(\boldsymbol{n}(\boldsymbol{t}_i), \boldsymbol{n}(\hat{\boldsymbol{t}}_j))$ where $\hat{\boldsymbol{t}}_j$ is the triangle where the median angle is achieved, and then we replace the weighted average normal $\boldsymbol{m}(\boldsymbol{t}_i)$ by $\boldsymbol{n}(\hat{\boldsymbol{t}}_j)/\|\boldsymbol{n}(\hat{\boldsymbol{t}}_j)\|$.

The mean and median filtering methods show better performance than the Laplacian flow. These two methods, however, require a large number of iterations to reach stable results.

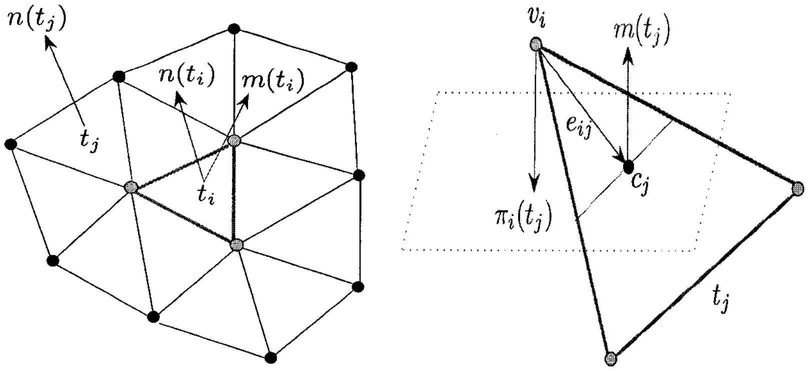


Figure 2.4: Illustration of the mesh median filter algorithm.

2.3.3 Weighted Laplacian filter

Instead of using unit edge costs, the weighted Laplacian smoothing method [17] chooses edge weights based on the approximation to the curvature normal. The edge weights w_{ij} are defined as

$$w_{ij} = \cot \alpha_{ij} + \cot \beta_{ij}$$

where α_{ij} and β_{ij} are the angles $\angle \boldsymbol{v}_i \boldsymbol{v}_{j-1} \boldsymbol{v}_j$ and $\angle \boldsymbol{v}_i \boldsymbol{v}_{j+1} \boldsymbol{v}_j$ depicted in Fig. 2.5. Then, the update rule of the weighted Laplacian smoothing procedure is given by

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \frac{1}{\sum\limits_{\boldsymbol{v}_j \in \boldsymbol{v}_i^{\star}} w_{ij}} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^{\star}} w_{ij} (\boldsymbol{v}_j - \boldsymbol{v}_i). \tag{6}$$

The improved edge weights are used to compensate for the irregularities of the triangle mesh and helps to avoid the edge equalization.

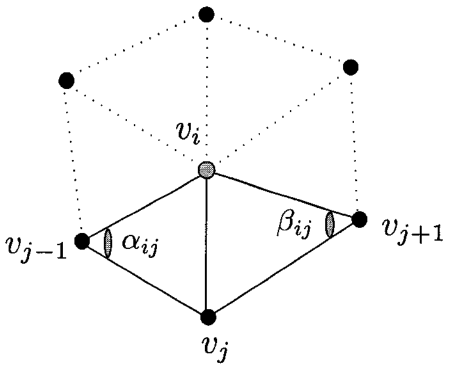


Figure 2.5: Illustration of the angles α_{ij} and β_{ij}.

2.3.4 Bilateral mesh denoising

Similar to the mean and angle median filters, the bilateral 3D mesh denoising method [15] was also adopted from the bilateral filtering technique used in image denoising. This algorithm filters each vertex $\boldsymbol{v}_i$ of the mesh in the normal direction using local neighborhoods according the following update rule:

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \boldsymbol{n}_i \left(\frac{\sum\limits_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} (w_{ij}^c \, w_{ij}^s) \langle \boldsymbol{n}_i, \boldsymbol{v}_i - \boldsymbol{v}_j \rangle}{\sum\limits_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} w_{ij}^c \, w_{ij}^s} \right),$$

where $\boldsymbol{n}_i$ is the vertex normal, w_{ij}^c is the standard Gaussian filter $w_{ij}^c = e^{-\|\boldsymbol{v}_i - \boldsymbol{v}_j\|^2/2\sigma_c^2}$ with parameter σ_c, and w_{ij}^s is feature-preserving weight function $w_{ij}^s = e^{-\langle \boldsymbol{n}_i, \boldsymbol{v}_i - \boldsymbol{v}_j \rangle^2/2\sigma_s^2}$ with parameter σ_s. Bilateral mesh denoising algorithm is parameter-dependent and requires the user to assign the two parameters σ_c and σ_s interactively. The lack of object information, however, might affect the smoothing result.

2.4 Proposed Method

The proposed vertex-based method for 3D mesh smoothing is motivated by the good performance of anisotropic diffusion in 2-D image denoising, and it is defined by the following nonlinear partial differential equation

$$\boldsymbol{v}_t = \operatorname{div}(g(|\nabla \boldsymbol{v}|)\nabla \boldsymbol{v}), \tag{7}$$

where g is Cauchy weight function (see Fig. 2.6) given by

$$g(x) = \frac{1}{1 + x^2/c^2}, \tag{8}$$

and c is a constant tuning parameter that needs to be estimated.

Intuitively, the smoothing effect of the proposed flow may be explained as follows: in flat regions of a 3D mesh where the vertex gradient magnitudes are relatively small, Eq. (7) is reduced to the heat equation which tends to smooth more but the smoothing effect is unnoticeable. And around the sharp features of the 3D mesh where the vertex gradient magnitudes are large, the diffusion flow given by Eq. (7) tends to smooth less and hence leads to a much better preservation of the mesh geometric structures.

It can be shown (see [26]) that the 95% asymptotic efficiency on the standard Gaussian distribution is obtained with the tuning constant $c = 2.3849$. This tuning value is used in all the experiments results of Section 2.5.

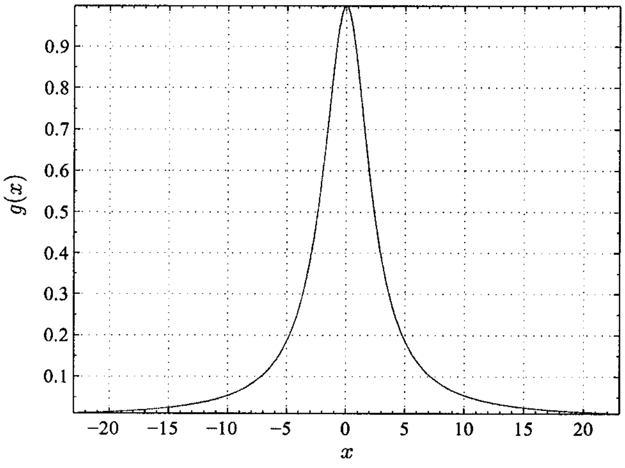


Figure 2.6: Cauchy weight function with $c = 2.3849$.

In discrete form, it is easy to show that the proposed vertex-based nonlinear diffusion is reduced to the following update rule

$$\boldsymbol{v}_i \leftarrow \boldsymbol{v}_i + \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \frac{1}{\sqrt{d_i}} \left(\frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_i}{\sqrt{d_i}} \right) \Big(g(|\nabla \boldsymbol{v}_i|) + g(|\nabla \boldsymbol{v}_j|) \Big), \tag{9}$$

where the gradient magnitudes are given by

$$|\nabla \boldsymbol{v}_i| = \sqrt{\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \left\| \frac{\boldsymbol{v}_i}{\sqrt{d_i}} - \frac{\boldsymbol{v}_j}{\sqrt{d_j}} \right\|^2}, \tag{10}$$

and

$$|\nabla \boldsymbol{v}_j| = \sqrt{\sum_{\boldsymbol{v}_k \in \boldsymbol{v}_j^\star} \left\| \frac{\boldsymbol{v}_j}{\sqrt{d_j}} - \frac{\boldsymbol{v}_k}{\sqrt{d_k}} \right\|^2}. \tag{11}$$

Note that the update rule of the proposed method requires the use of two neighboring rings as depicted in Fig. 2.7.

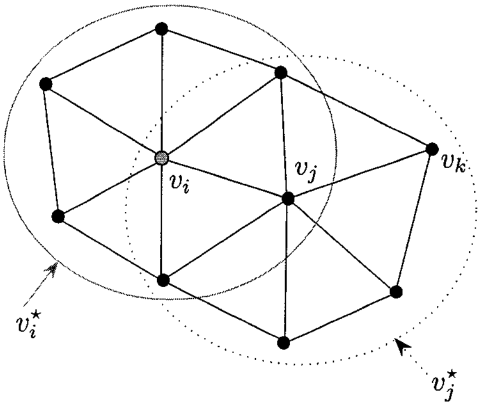
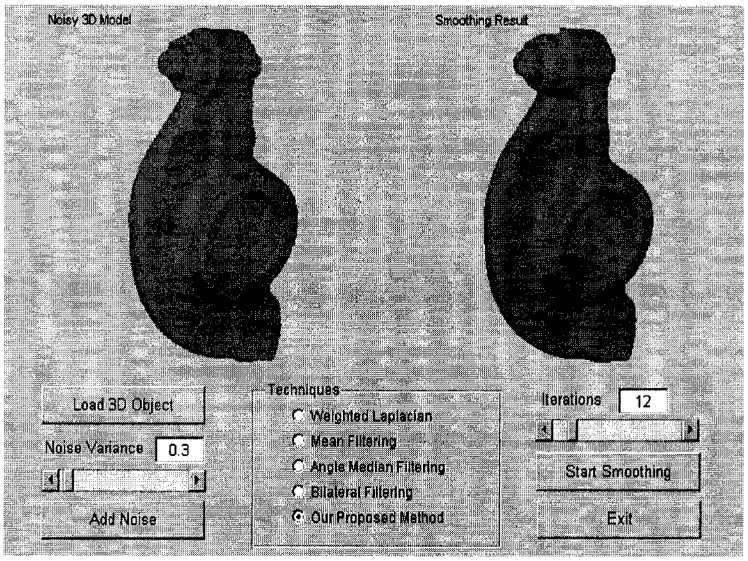


Figure 2.7: Illustration of two neighboring rings.

2.5 Experimental Results

This section presents simulation results where the mean filtering [14], angle median filtering [14], weighted Laplacian [17,18], bilateral filtering [15], and the proposed method are applied to noisy 3D models obtained by adding Gaussian noise to the original models shown in Fig. 2.9. The standard deviation of the noise was set to 2% of the mean edge length, that is $\sigma = 0.02\,\bar{\ell}$, where $\bar{\ell}$ is given by Eq. (2).

For ease of visualization, we designed a user-friendly Graphical User Interface (GUI) as depicted in Fig. 2.8 to test the performance of the proposed technique with different 3D models, and to also perform a comparison with the most prevalent methods used in 3D mesh smoothing. Fig. 2.10(a) depicts a noisy 3D cow model, and Fig. 2.10(b) through Fig. 2.10(f) show the denoising results using the weighted Laplacian flow, mean filtering, angle median filtering, bilateral filtering, and the proposed method respectively. These results clearly show that our method outperforms all the mesh filtering techniques used for comparison. Moreover, the proposed method is simple and easy to implement. One main advantage of the proposed algorithm is that it requires only few iterations to smooth out the noise, whereas the mean and angle median filters require substantial computational time. On the other hand, the bilateral filtering technique is also computationally fast, but has a poor smoothing performance in comparison with the proposed method as illustrated in Fig. 2.11 where we use the zoom tool to enlarge the view of the 3D cow model's head in order to clearly show the better performance of our proposed algorithm. In particular, the geometric structures and the fine details around the eye and the ear of the 3D cow model are very well preserved by our method. Also, note that the bilateral filter produces pointy horns, whereas the proposed method preserves the structure of the original horns pretty well. More experimental results showing the better performance of the proposed algorithm are presented in Fig. 2.12 and Fig. 2.13.



Noise Variance 0.3

Techniques
○ Weighted Laplacian
○ Mean Filtering
○ Angle Median Filtering
○ Bilateral Filtering
● Our Proposed Method

Iterations 12

Figure 2.8: Graphical User Interface.

In all the experiments, we observe that the proposed technique is able to suppress noise while preserving important geometric structure of the 3D models in a very fast and efficient way. This better performance is in fact consistent with a large number of 3D models used for experimentation.

To quantify the better performance of the proposed approach in comparison with the mean, angle median, weighted Laplacian and bilateral filters, we propose a nonlinear visual metric error $D(\mathbb{M}, \widehat{\mathbb{M}})$ defined between the original model $\mathbb{M}$ and the smoothing result model $\widehat{\mathbb{M}}$ as follows

$$D(\mathbb{M}, \widehat{\mathbb{M}}) = \frac{1}{2m} \left(\sum_{i=1}^{m} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\|^2 + \sum_{i=1}^{m} \|\mathcal{A}(\boldsymbol{v}_i) - \mathcal{A}(\hat{\boldsymbol{v}}_i)\|^2 \right), \qquad (12)$$

where $\{\boldsymbol{v}_i\}_{i=1}^{m}$ and $\{\hat{\boldsymbol{v}}_i\}_{i=1}^{m}$ are the mesh vertex sets of $\mathbb{M}$ and $\widehat{\mathbb{M}}$ respectively, and $\mathcal{A}$ is a

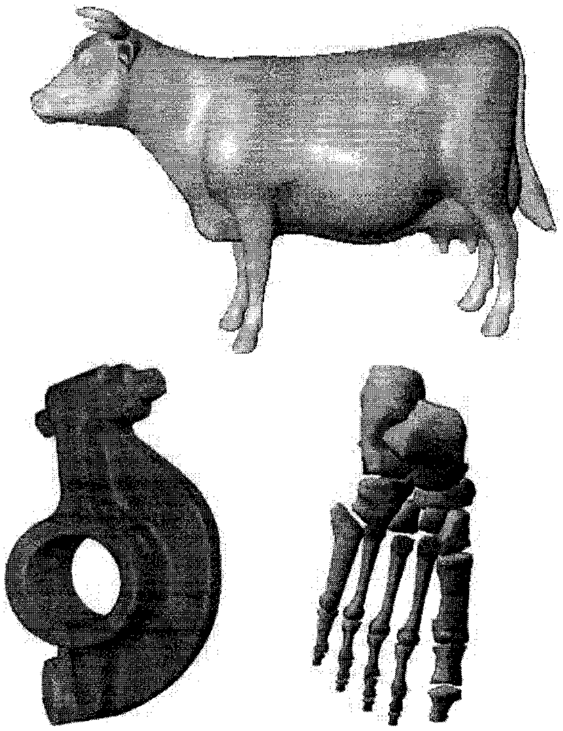


Figure 2.9: Original 3D models used for experimentation: cow, rocker arm, and foot bones.

nonlinear diffusion operator given by

$$\mathcal{A}(\boldsymbol{v}_i) = \frac{1}{d_i} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} (\boldsymbol{v}_i - \boldsymbol{v}_j)\Big(g(|\nabla \boldsymbol{v}_i|) + g(|\nabla \boldsymbol{v}_j|)\Big).$$

The values of visual error metric for all the experiments are depicted in Fig. 2.14-2.16 which clearly show that the proposed method gives the best results indicating the consistency with the subjective comparison.

2.6 Conclusions

In this chapter, we introduced a vertex-based anisotropic diffusion for 3D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the 3D mesh data. The experimental results clearly showed a much improved performance of the proposed approach in comparison with the current methods used in 3D mesh smoothing. For future work, we plan to incorporate the curvature information as well as additional regularization terms into the proposed model.

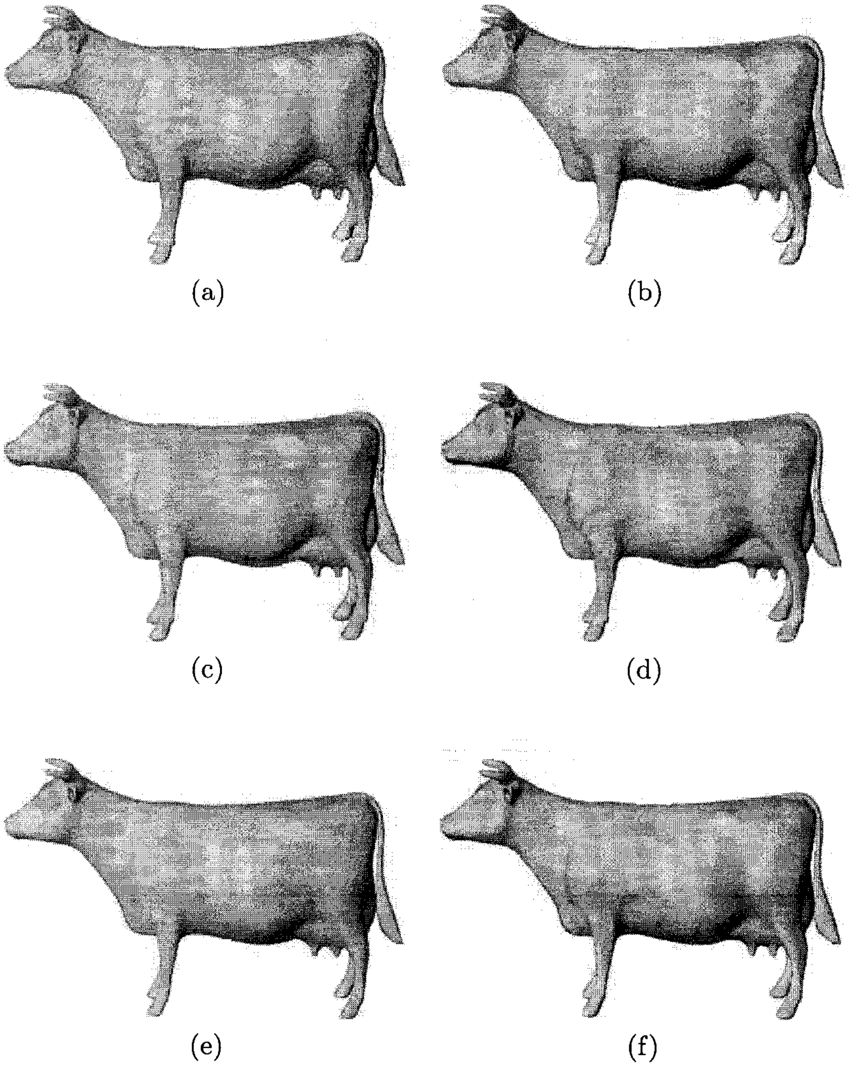
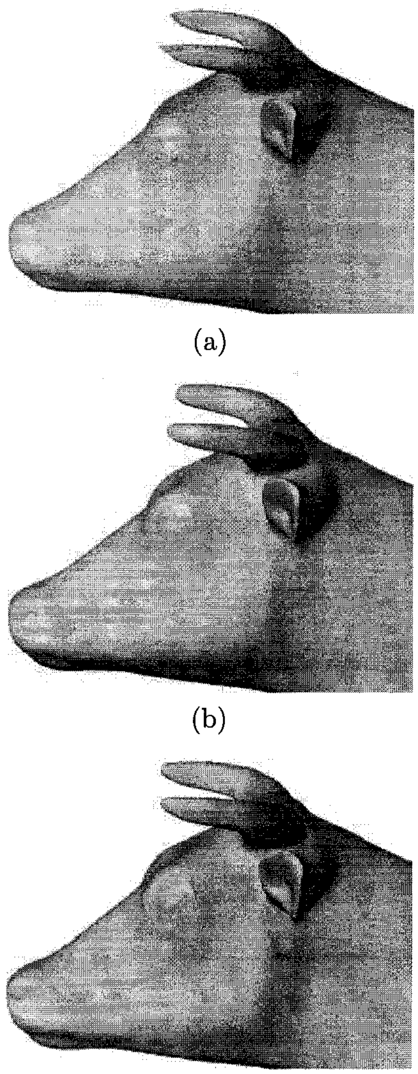

(a) (b) (c) (d) (e) (f)

Figure 2.10: 3D mesh smoothing results. (a) Noisy 3D cow model with 92,864 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by bilateral mesh flow, (f) smoothed by the proposed approach. The number of iteration times is 18 for each case.

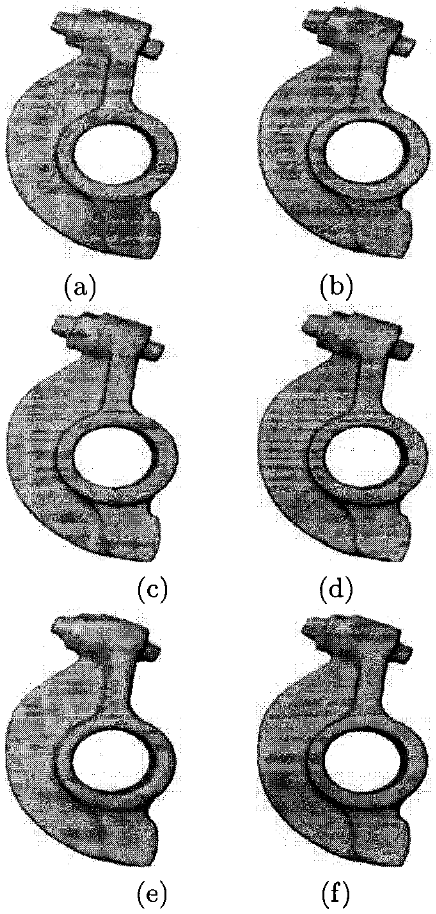


(a)

(b)

(c)

Figure 2.11: 3D mesh smoothing results after zooming on the head of the 3D cow model. (a) Smoothed by bilateral mesh flow, (b) smoothed by the proposed approach. (c) original model

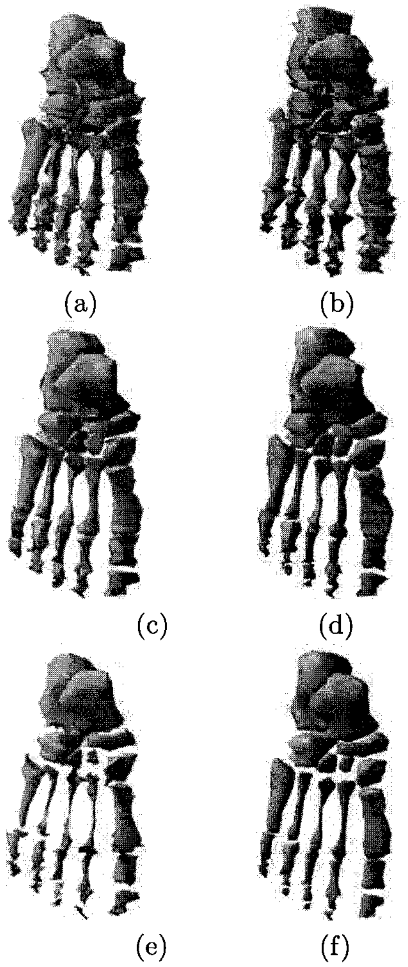


(a) (b)

(c) (d)

(e) (f)

Figure 2.12: 3D mesh smoothing results. (a) Noisy 3D rocker arm model with 80,354 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by bilateral mesh flow, (f) smoothed by the proposed approach. The number of iteration times is 12 for each case.



(a) (b)

(c) (d)

(e) (f)

Figure 2.13: 3D mesh smoothing results. (a) Noisy 3D foot bones model with 4,204 triangles, (b) smoothed by weighted Laplacian flow, (c) smoothed by mean filtering, (d) smoothed by angle median filtering, (e) smoothed by bilateral mesh flow, (f) smoothed by the proposed approach. The number of iteration times is 3 for each case.

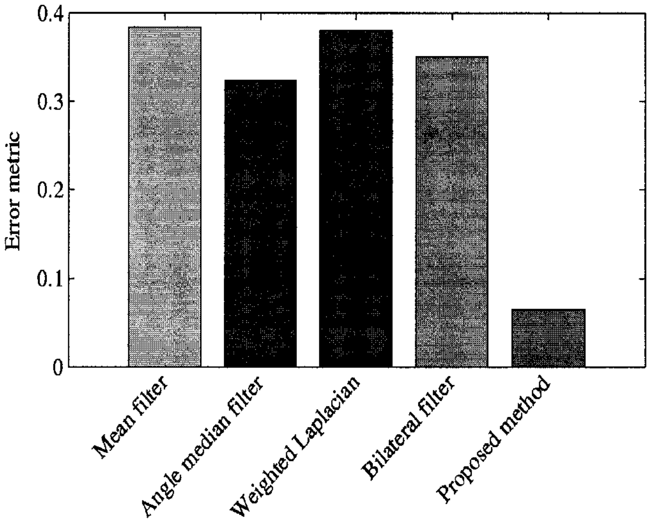


Figure 2.14: Nonlinear visual metric error for the 3D cow model.

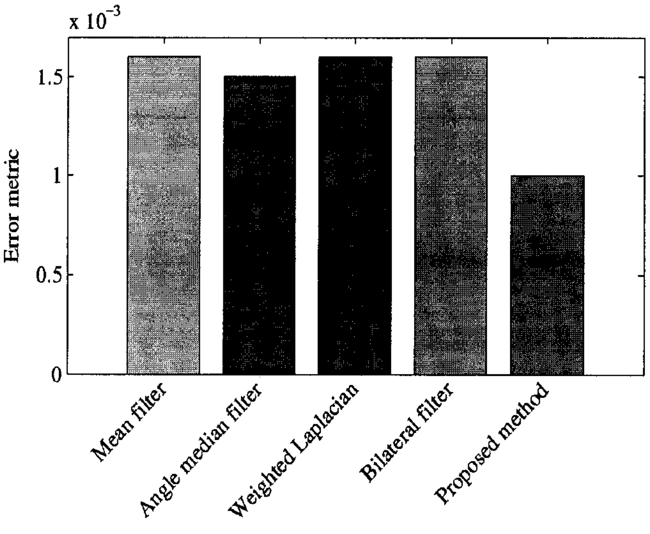


Figure 2.15: Nonlinear visual metric error for the 3D rocker arm model.

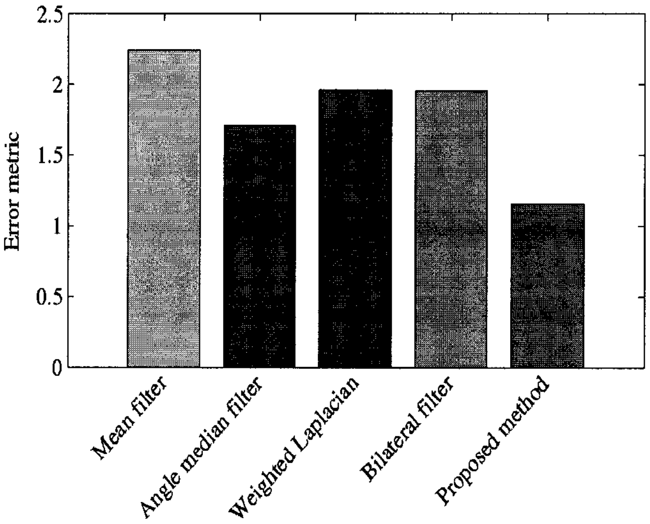


Figure 2.16: Nonlinear visual metric error for the 3D foot bones model.

CHAPTER 3

Statistical 3D Mesh Distributions

In this chapter we introduce several statistical distributions to analyze the topological properties of 3D graphics. The proposed statistical measures include the mesh degree, the mesh assortativity, the mesh clustering coefficient, and the mesh geodesic distance distributions. These probabilistic distributions provide useful information about the way 3D mesh models are connected. Illustrating experimental results show the effectiveness of the proposed measures in quantifying the topological features of 3D objects.

3.1 Introduction

Shape analysis of three-dimensional (3D) objects has become an active research field with the recent developments in solid modeling and visualization [44]. Nowadays, vast amounts of 3D models are being developed and are distributed freely or commercially on the internet. 3D graphics are commonly used in several multimedia applications such as video gaming, engineering design, virtual reality, e-commerce and scientific visualization. 3D objects consist of geometric and topological information. Topology is the property that determines which parts of the shape of objects are connected to which other parts, while geometry determines where, in a given coordinate system, each part is located. The topological properties of 3D objects in the statistical framework are the focus of this chapter, and the main objective is to convert the 3D mesh data into useful and meaningful information. To convert the 3D mesh data into information, we need appropriate probabilistic tools and techniques [45–48]. Such statistical methods will help us quantify the mesh topological properties, and to also gain very useful information by presenting the same data graphically. Even with modern 3D computer graphics tools, we cannot draw a meaningful conclusion about a large 3D model or graph using direct examination by the human eye [46]. This examination analysis can be done only by statistical methods for quantifying the connectivity information of large graphs. Generally, graph-theoretic representations of a problem are based towards highlighting aspects such as the connectivity of components in the problem. In recent years, we have witnessed a substantial development in graph theory research, with the attention shifted from the analysis of small graphs and the properties of individual vertices or edges within such graphs to the consideration of large-scale statistical properties of graphs [46]. This change has been driven fast by the availability of powerful computers and communication networks that allow us to gather and analyze data for larger 3D mesh models.

In this chapter, we introduce some statistical distributions to analyze the connectivity information of 3D mesh models. The proposed distributions are defined in terms of the topological properties of the mesh. In particular, we introduce the vertex degree distribution, and the geodesic distance distribution which captures the nonlinearity structure and the intrinsic geometry of a 3D object. We also propose the clustering coefficient distribution which provide useful information about a 3D mesh and the way it is connected. Clustering techniques have been well studied in many applications such as data mining,image segmentation, pattern classification and text categorization in which a group of similar words characteristics are clustered in the same group. Recently a graph based-representation approach has been used [49], and the main idea is to analyze the human language acquisition by representing the set of terms as vertices and edges representing the relationship between them. Also, statistical measures have been recently studied to determine how well the internet topologies are connected in order to reflect many practical and theoretical network characteristics such as robustness of the network under attack.

The rest of this chapter is organized as follows. In Section 3.2, we briefly recall some basic concepts of 3D mesh data, and we introduce the Laplacian matrix of a triangle mesh and its spectral decomposition. In Section 3.3, we propose some statistical distributions to study the topological properties of 3D mesh data and we explain in more details the fundamental aspects of each distribution. In Section 3.4, we provide some experimental results to show the usefulness of the proposed statistical measures in 3D graphics. Finally, we conclude in Section 3.5.

3.2 Problem Formulation

The Laplacian matrix of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by $L = D - A$, where $A = (a_{ij})$ is the adjacency matrix between the vertices, that is $a_{ii} = 0$ and $a_{ij} = 1$ if

$\boldsymbol{v}_i \sim \boldsymbol{v}_j$; and $D = \mathrm{diag}\{d_i : i = 1, \ldots, m\}$ is the degree matrix (diagonal matrix whose (i, i) entry is d_i). The Laplacian matrix may be written as [25]

$$L = (\ell_{ij}) = \begin{cases} d_i & \text{if } \boldsymbol{v}_i = \boldsymbol{v}_j \\ -1 & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{o.w.} \end{cases}$$

The Laplacian matrix is symmetric and positive semidefinite [25]. Hence, The eigenvalues of L are nonnegative $0 = \lambda_1 < \lambda_2 \leq \ldots \lambda_m$. The first eigenvector is $\mathbf{1} = (1, 1, \ldots, 1)^T$ (an m-vector of ones), and all the remaining eigenvectors are orthogonal to $\mathbf{1}$. The second eigenvalue λ_2 is called the *algebraic connectivity* of the mesh, that is $\lambda_2 > 0$ if and only if the mesh is connected. In general, the spectral analysis of a triangle mesh is useful for discovering clusters of highly interconnected vertices [47, 48].

3.3 Proposed Statistical Measures

3.3.1 Average vertex degree

The average vertex degree of a triangle mesh $\mathbb{M}$ is the coarsest connectivity characteristic, and it given by

$$\overline{d} = \frac{1}{m} \sum_{i=1}^{m} d_i = \frac{2|\mathcal{E}|}{m}. \tag{1}$$

The average vertex degree may be used as a measure of connectivity of a triangle mesh. We say that a triangle mesh is "better-connected" if it has a higher average vertex degree. Fig. 3.1 shows an example of two triangle meshes with different average vertex degrees. Although the number of edges in the 3D bunny model is about 14.2 times larger than the

3D camel model, we found out, however, that the 3D camel has a much higher average vertex degree and hence it is much better-connected than the 3D bunny.

The average vertex degree, however, has limited utility since triangle meshes with the same average vertex degree may have vastly different topological structures.

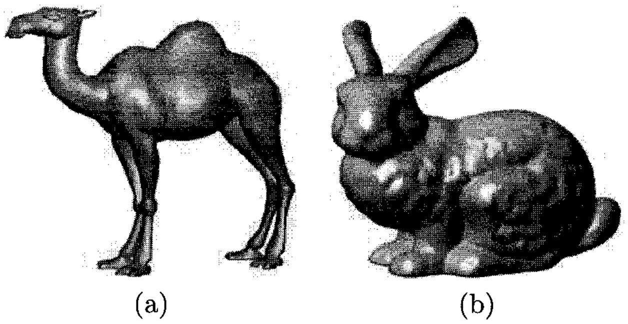


Figure 3.1: (a) 3D camel model, (b) 3D bunny model.

3.3.2 Mesh degree distribution

A more informative characteristic of graph connectivity is the vertex degree distribution, which provides information on the number of vertices in the triangle mesh having a certain degree. The vertex degree distribution is the probability that a randomly selected vertex has a degree equal to k, and it is given by

$$p_k = \frac{m_k}{m},$$

where m_k is the number of vertices of degree k, that is m_k is the cardinality of the set $\mathcal{V}_k = \{\boldsymbol{v}_i \in \mathcal{V} : d_i = k\}$.

The degree distribution contains more information about the mesh connectivity than the average vertex degree since given a specific form of p_k we can always restore the average vertex degree by $\bar{k} = \sum_{k=1}^{k_{\max}} kp_k$, where $k_{\max}$ is the maximum vertex degree in the set $\mathcal{V}_k$.

3.3.3 Mesh assortativity distribution

An important concept that provides important information about the correlations of the degree of neighboring vertices is the average neighbor connectivity of each vertex, and it is given by

$$K_{nn}(\boldsymbol{v}_i) = \frac{1}{d_i} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^{\star}} d_j, \quad \forall \boldsymbol{v}_i \in \mathcal{V}. \tag{2}$$

The assortativity function of a mesh is defined as

$$K_{nn}(k) = \frac{1}{m_k} \sum_{\boldsymbol{v}_i \in \mathcal{V}_k} K_{nn}(\boldsymbol{v}_i) \tag{3}$$

If degrees of neighboring vertices are uncorrelated, then the assortativity is a constant. When correlations are present, two main classes of possible correlations can been identified: assortative and disassortative. The triangle mesh has an assortative behavior if $K_{nn}(k)$ increases with k, indicating that large degree vertices are connected with other large degree vertices, and has an disassortative behavior if $K_{nn}(k)$ is a decreasing function of k. The mesh assortativity distribution is defined as the histogram of the assortativity function.

3.3.4 Mesh clustering coefficient distribution

Clustering refers to the the process of organizing objects into groups whose members are similar in some way. In the triangle mesh framework, clustering may be used to measure how close the neighbors of any vertex are to form a group within the mesh. The clustering coefficient of each vertex is defined as

$$C(\boldsymbol{v}_i) = \frac{2}{d_i - 1}, \quad \forall \boldsymbol{v}_i \in \mathcal{V}, \tag{4}$$

and the higher clustering coefficient of the vertex is, the more interconnected its neighbors are.

The average clustering coefficient of all vertices of degree k is given by

$$C(k) = \frac{1}{m_k} \sum_{\boldsymbol{v}_i \in \mathcal{V}_k} C(\boldsymbol{v}_i), \tag{5}$$

and the clustering coefficient distribution is defined as the histogram of $C(k)$.

3.3.5 Mesh geodesic distance distribution

The geodesic distance $g(\boldsymbol{v}_i, \boldsymbol{v}_j)$ between two mesh vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ is the shortest length $L(\gamma) = \int_a^b \|\gamma'(t)\| dt$ of a smooth curve $\gamma : [a, b] \to \mathcal{M}$ such that $\gamma(a) = \boldsymbol{v}_i$ and $\gamma(b) = \boldsymbol{v}_j$. The primary motivation behind the geodesic distance is of overcoming the limitations of the Euclidean distance which by virtue of its linearity in nature cannot account for nonlinear structures in a 3D object. Unlike the Euclidean distance which is basically a straight line between two points in 3D space, the geodesic distance captures the global nonlinear structure and the intrinsic geometry of the data. The geodesic distance calculation is based on a similar approach used for computing the isometric feature mapping (Isomap) for multidimensional scaling [50] on nonlinear manifolds [51]. The algorithm has two main steps:

(i) Construct a neighborhood graph by connecting a given vertex to its k-nearest neighbors, and link these neighboring vertices by edges with weights equal to the Euclidean distances.

(ii) Compute the geodesic distances (shortest paths) between all pairs of n points in the constructed graph using Dijkstra's or Floyd's algorithm.

Note that Step (i) may be alleviated by choosing a random subset of $\mathcal{V}$ in order to reduce the computational complexity of the geodesic calculation.

3.4 Experimental Results

This section presents experimental results where the proposed statistical distributions are applied to 3D mesh models with different topologies and varying sizes as depicted in Fig. 3.2(a)-3.7(a). For example, the 3D camel model contains 2443 vertices and 7326 edges, whereas the 3D bunny model consists of 34834 vertices and 104288 edges.

The Laplacian spectra of each model are depicted in Fig. 3.2(b)-3.7(b). Spectral analysis is a powerful tool for investigating 3D structures such as discovering clusters of highly connected vertices. By performing the spectrum analysis, a wide range of critical graph characteristic can be discovered. The largest values are particularly important. Most graphs with high values for these largest eigenvalues have small diameter and contain strong clusters. In our experiments, we discovered using spectral analysis of the Laplacian matrix that the eigenvalues start at most for the lower normalized ranks which explains that all the 3D models are highly clustered on those lower normalized ranks.

By analyzing the mesh degree distributions of the 3D models as shown in Fig. 3.2(c)-3.7(c), we can easily see that the probability of the degree increases as the vertex degree increases until it reaches the maximum with vertices of degree 6, then it starts decreasing as the vertex degree increases.

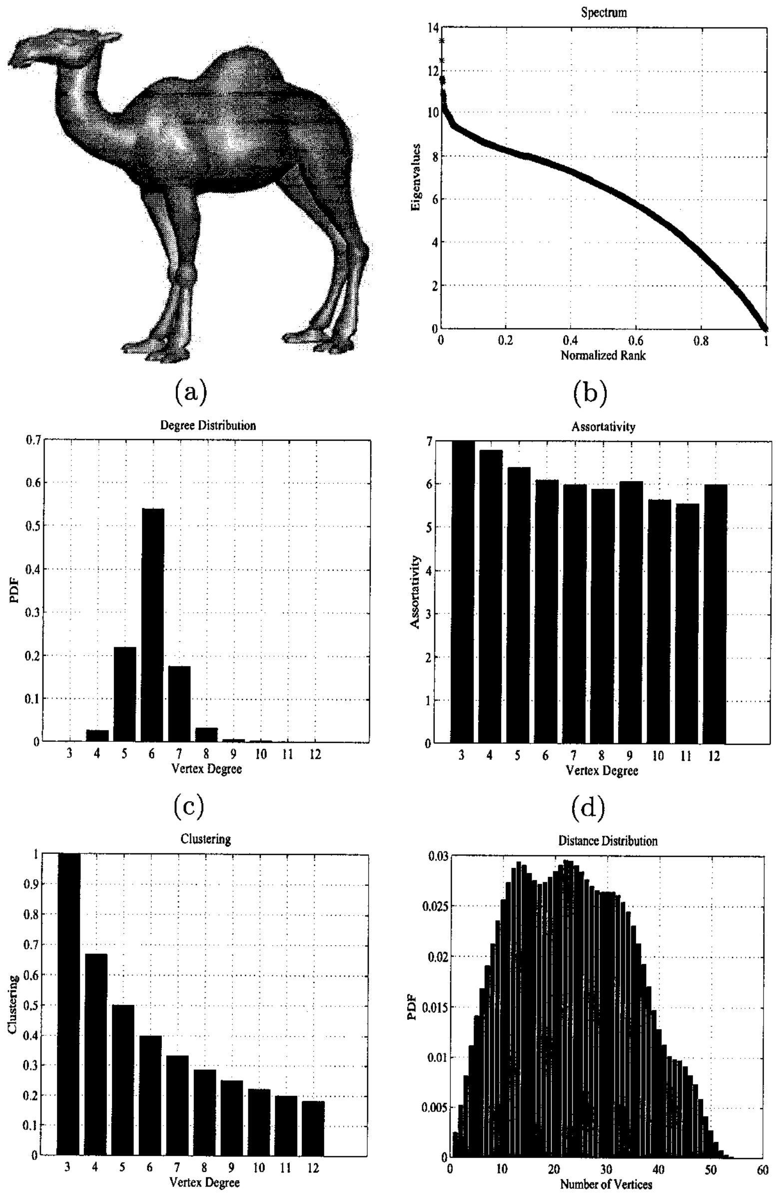
From the mesh assortativity plots shown in Fig. 3.2(d)-3.7(d), we observe that the distribution of assortativity depends on the distribution of vertices on a 3D mesh. For example, the 3D camel cow, and feto models have a strong dissortative behavior as illustrated in Fig. 3.2(d), Fig. 3.4(d) and Fig. 3.7(d), but all the remaining models show an assortative behavior behavior for small vertex degrees, and a dissortative for large vertex degrees.

For the mesh clustering coefficient distributions shown in Fig. 3.2(e)-3.7(e), the vertices of degree 3 have the highest coefficient value which is equal to 1, meaning that every neighbor connected to a vertex v_i of degree 3 is also connected to every other vertex within the neighborhood. Moreover, the clustering coefficient starts decreasing as the vertex degree increases until it reaches a lower clustering value for the highest vertices degree.

The mesh geodesic distance distributions as depicted in Fig. 3.2(f)-3.7(f) clearly show that the nonlinear structure of each 3D object. For example, the geodesic distributions of the 3D camel and dragon models have a multimodal shape which clearly capture the humps in these models.

3.5 Conclusions

In this chapter, we proposed several statistical measures to analyze the topological properties of 3D models. The proposed statistical measures include the mesh degree, the mesh assortativity, the mesh clustering coefficient, and the mesh geodesic distance distributions. The experimental results clearly showed the effectiveness of the proposed measures in quantifying the topological features of 3D objects.



(a) (b)

(c) (d)

(e) (f)

Figure 3.2: (a) 3D camel model, (b) spectrum, (c) vertex degree distribution, (d) assortativity distribution, (e) clustering distribution, (f) geodesic distance distribution.

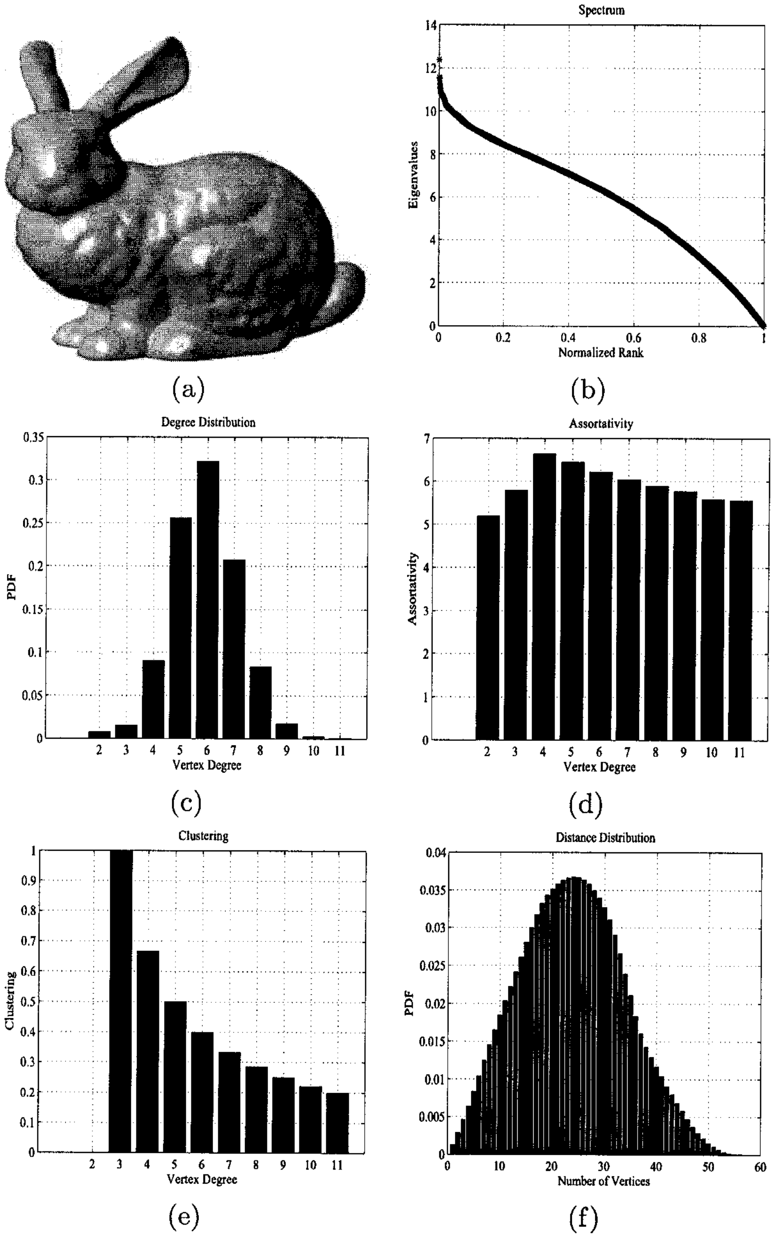
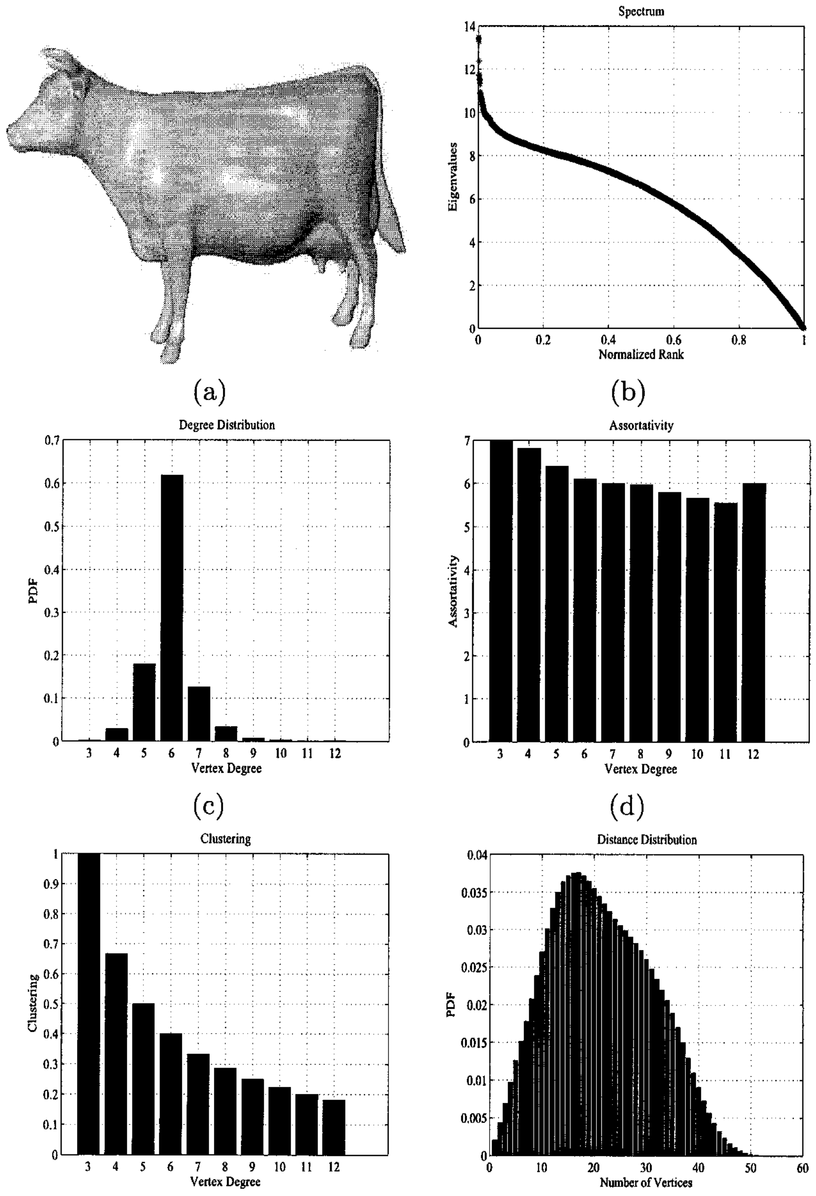
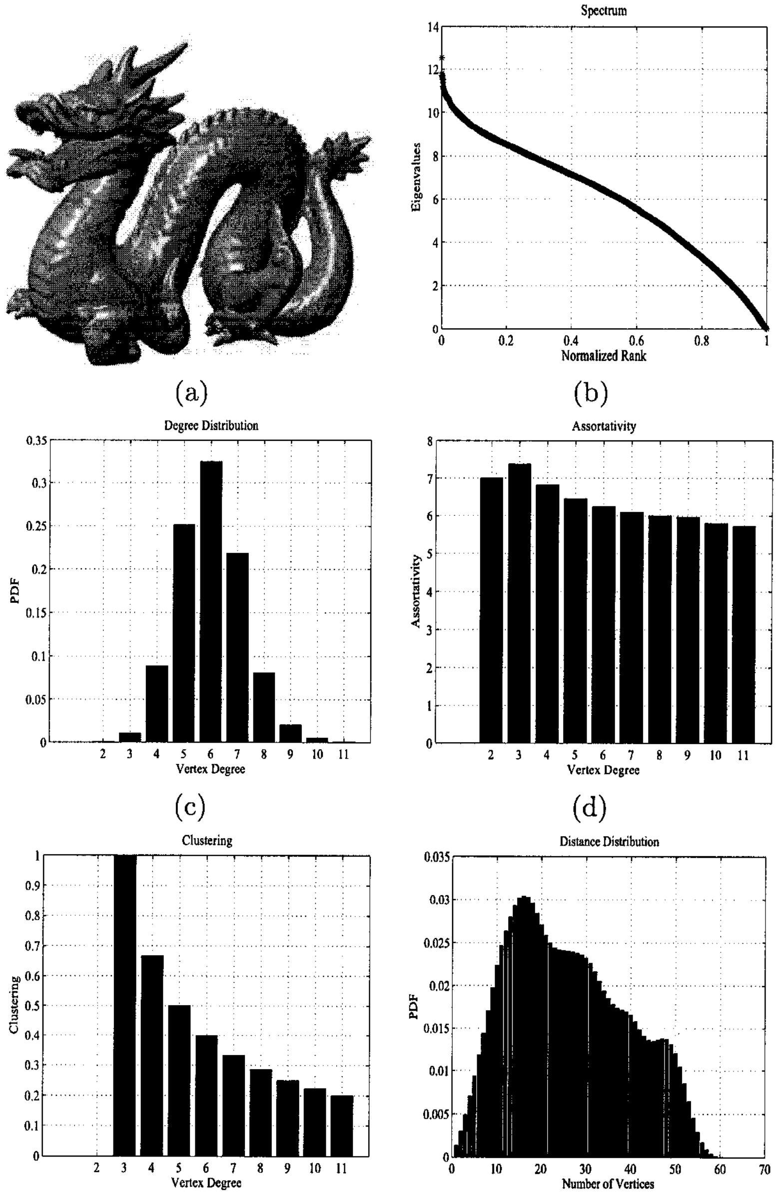


Figure 3.3: (a) 3D bunny model, (b) spectrum, (c) vertex degree distribution, (d) assortativity distribution, (e) clustering distribution, (f) geodesic distance distribution.



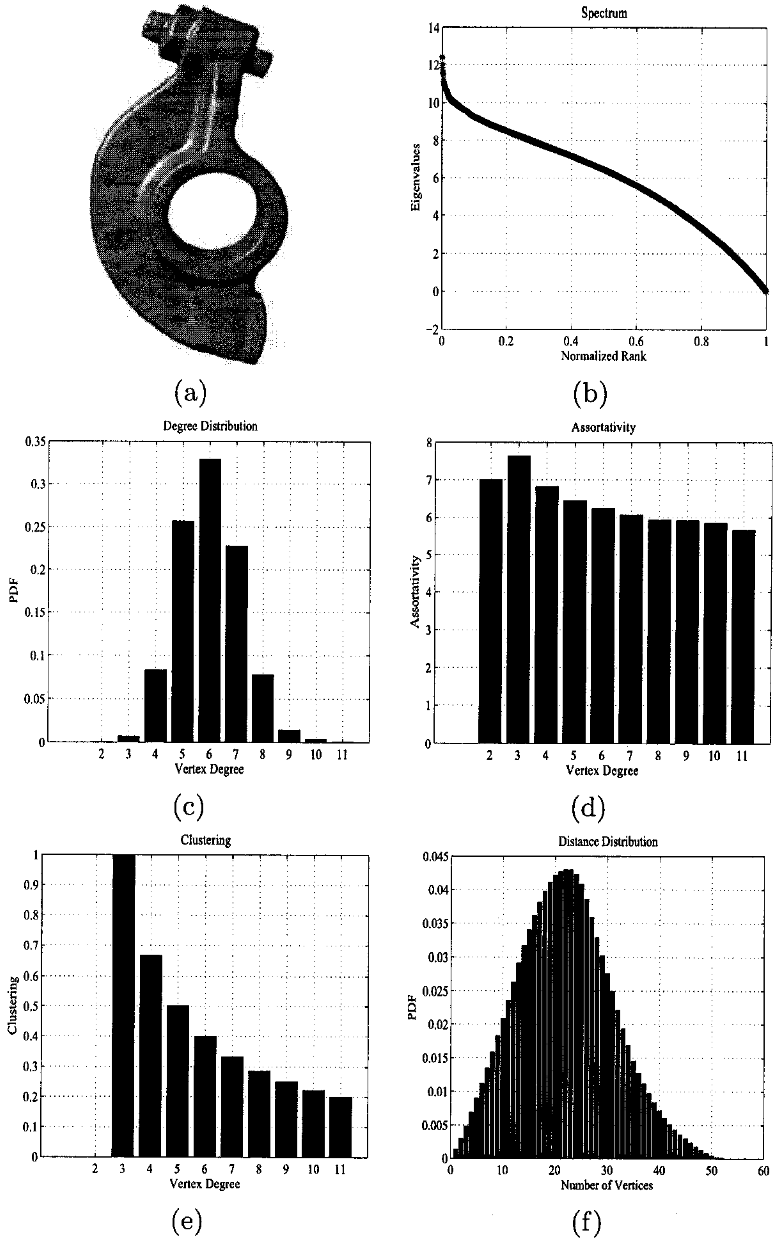
(a) (b) (c) (d) (e) (f)

Figure 3.4: (a) 3D cow model, (b) spectrum, (c) vertex degree distribution, (d) assortativity distribution, (e) clustering distribution, (f) geodesic distance distribution.



(a) (b) (c) (d) (e) (f)

Figure 3.5: (a) 3D dragon model, (b) spectrum, (c) vertex degree distribution, (d) assortativity distribution, (e) clustering distribution, (f) geodesic distance distribution.

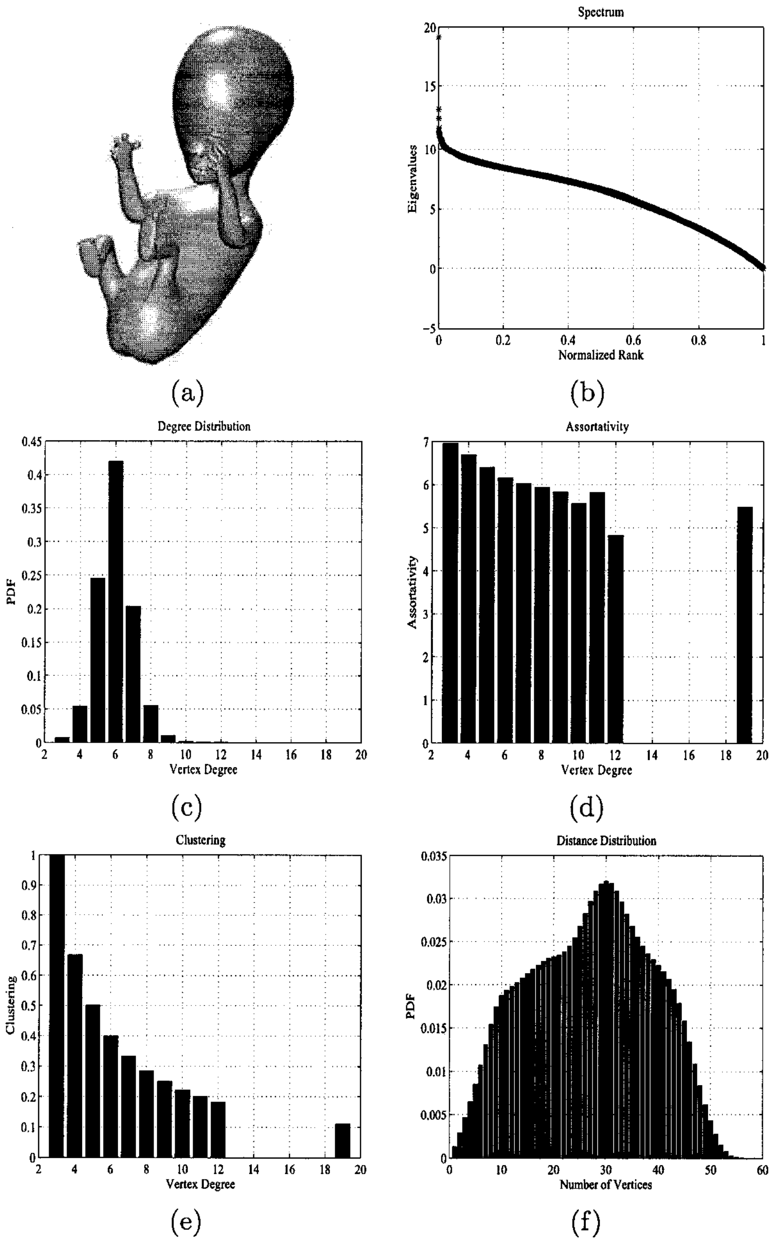


(a) (b)

(c) (d)

(e) (f)

Figure 3.6: (a) 3D mechanical model, (b) spectrum, (c) vertex degree distribution, (d) assortativity distribution, (e) clustering distribution, (f) geodesic distance distribution.



(a) (b)

(c) (d)

(e) (f)

Figure 3.7: (a) 3D feto model, (b) spectrum, (c) vertex degree distribution, (d) assortativity distribution, (e) clustering distribution, (f) geodesic distance distribution.

CHAPTER 4

Euclidean Distance-Based Skeletal Graph

We present an invariant shape descriptor for topological coding of 3D objects. The proposed approach encodes a 3D object into a topological graph using a normalized distance function. Unlike the height function which has been traditionally used to model topology, the proposed distance function-based graph is invariant to rigid motion transformations. Simulations results demonstrate the potential of the proposed topological graph which may be used as a shape signature for object matching and reconstruction.

4.1 Introduction

Shape recognition of three-dimensional objects has become an active research field with the recent developments in solid modeling and visualization [44]. Today, vast amounts of 3D models are being developed and are distributed freely or commercially on the internet. For purposes of object matching and 3D database search, a reliable and efficient shape representation or descriptor, which will encode the shape information of objects, is essential.

Our proposed approach aims to represent objects with topological coding. Topology represents the connectedness of a shape and enables parts of shapes, which are connected, to be mapped and drawn equivalently. Morse theory is a branch of topology that studies the properties of a Morse function, that is a function that has only nondegenerate singular points [44], and it describes the topology changes of the level sets of this function at those singularities. Regular or noncritical points do not affect the number or genus of the components of the level sets. It can be shown that Morse functions are dense and stable in the set of all smooth functions, that is the structure of nondegenerate singularities does not change under small perturbations. A Morse theory-based representation that captures topological properties of objects is the so-called Reeb graph representation proposed by Shinagawa *et al.* [28, 29]. The vertices of the Reeb graph are the singular points of a function defined on the underlying object [28, 29, 44].

In this chapter, we propose an invariant Reeb graph for topology coding using the Morse distance function [30]. The key idea is to identify and encode regions of topological interest of a 3D object in the Morse-theoretic framework [30–32]. The main motivations behind using the distance function are: (a) rotational invariance which makes it more adapted to object recognition than the Morse height function, and (b) it can be shown that a surface may be reconstructed from its intersections with concentric spheres centered at the barycenter of the underlying surface [44]. The surface topological changes occur as we change the values of the radii of these concentric spheres. At singular points, the level curves of the distance function may split or merge which indicate topological changes.

The rest of this chapter is organized as follows. The next section briefly describes the concepts of Morse theory, followed by a mathematical description of the Reeb graph representation for 3D topological modeling. In Section 4.3, we propose a novel distance function based approach to construct invariant Reeb graphs of 3D objects. And finally in Section 4.4, we provide simulation results for topological coding using the distance function based Reeb graph, and we discuss its most attractive properties as well as its low computational complexity.

4.2 Morse theory and Topological Modeling

4.2.1 Morse theory

Morse theory explains the presence and the stability of singular points in terms of the topology of the underlying smooth manifold. Topology is the property that determines which parts of the shape of objects are connected to which other parts [33, 34], while geometry determines where, in a given coordinate system, each part is located [34]. The basic principle is that the topology of a manifold is very closely related to the singular points of a smooth function defined on that manifold [30]. A smooth function $f : \mathbb{M} \rightarrow \mathbb{R}$ on a smooth manifold $\mathbb{M}$ is called a *Morse function* if all its singular points are nondegenerate, i.e. the Hessian matrix is nonsingular at every singular point. The only nondegenerate singularities are the minimum, maximum and saddle points as depicted in Figure 4.1. Nondegenerate singularities are isolated, that is, there cannot be a sequence of nondegenerate singularities converging to a nondegenerate singularity $\boldsymbol{p} \in \mathbb{M}$. Another important result is Morse theorem which says that a small, smooth perturbation of a Morse function yields another

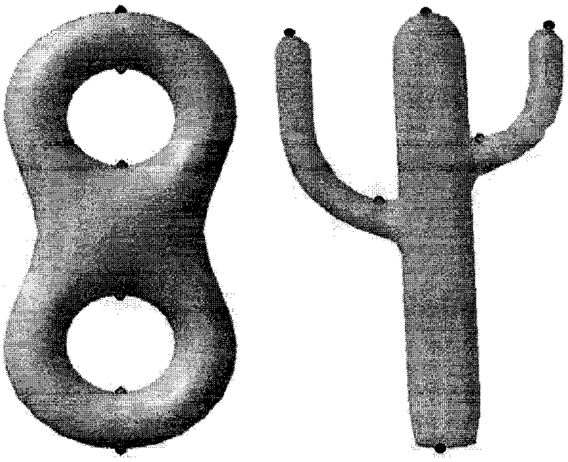


Figure 4.1: Nondegenerate singular points of a Morse height function: minimum (blue), saddle (green), maximum (red).

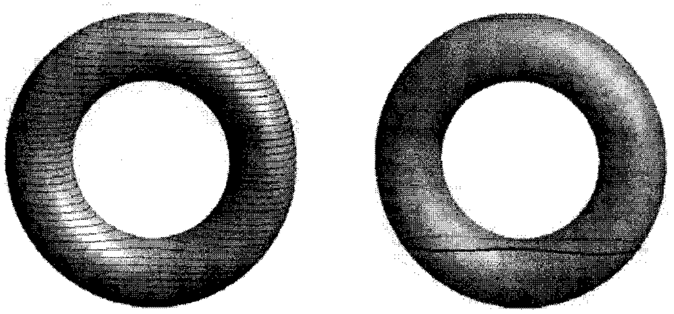
Morse function. The density means that there is a Morse function arbitrarily close to any non-Morse function. A *level set* $f^{-1}(a)$ of f at a value a may be composed of one or many connected components, such as closed curves on the surface. Morse deformation lemma states that if no critical points exist between two level sets of f, then the two level sets are topologically equivalent and can be deformed onto one another [4]. In particular, they consist of the same number of connected components [1]. Furthermore, Morse theory implies that topological changes on the level sets occur only at critical points [3]. This property is illustrated next.

4.3 Level sets around Morse points

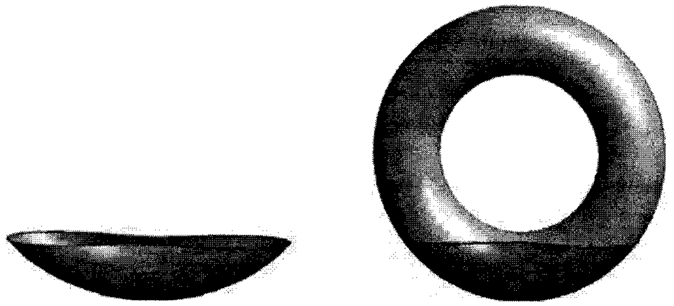
Let $f : \mathbb{M} \to \mathbb{R}$ be a Morse function defined on a compact surface $\mathbb{M}$, and denote by $\mathbb{M}_a$ the sub-surface of $\mathbb{M}$ consisting of all points at which f takes values less than or equal to a real number a

$$\mathbb{M}_a = \{\boldsymbol{p} \in \mathbb{M} : f(\boldsymbol{p}) \leq a\}.$$

Let L_a be the set of points where the value of f is exactly a, that is $L_a = f^{-1}(a)$. Note that when a is a regular value, the set L_a is a smooth curve of $\mathbb{M}$ and it is the boundary of $\mathbb{M}_a$.



(a) Level curves (b) Level curve L_a



(c) Subsurface $\mathbb{M}_a$ (d) Subsurface and Level curve

Figure 4.2: Illustration of $\mathbb{M}_a$ and L_a.

Proposition 4.1 *Let $a < b$ be real numbers such that the function $f : \mathbb{M} \to \mathbb{R}$ has no critical value in the interval $[a, b]$, then*

(a) The level curves $L_a = f^{-1}(a)$ and $L_b = f^{-1}(b)$ are diffeomorphic.

(b) The subsurfaces $\mathbb{M}_a$ *and* $\mathbb{M}_b$ *are diffeomorphic, with boundaries* L_a *and* L_b *respectively.*

Figure 5.1 shows the evolution of the subsurface $\mathbb{M}_a$ as the parameter a changes. If $a < \min_{\boldsymbol{p}\in\mathbb{M}}\{f(\boldsymbol{p})\}$, then $\mathbb{M}_a = \emptyset$. And as we increase the parameter a, the subsurface $\mathbb{M}_a$ changes until it covers the entire surface $\mathbb{M}$. We may think of the height function $f : \mathbb{M} \to \mathbb{R}$ as dipping a doughnut into a cup of chocolate cream.

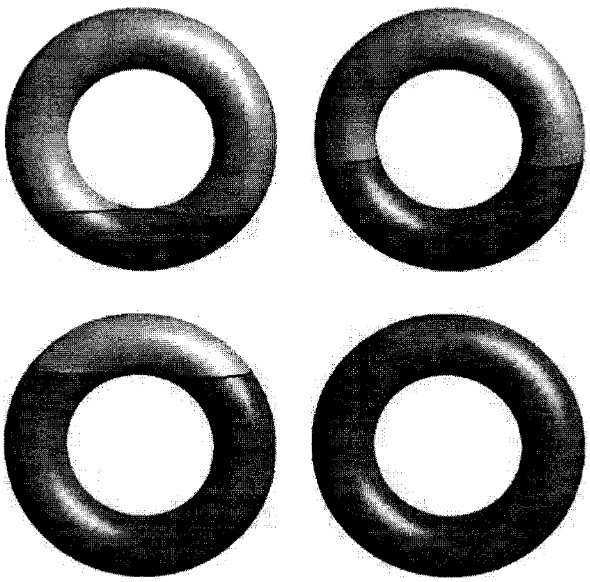


Figure 4.3: Evolution of $\mathbb{M}_a$ as a changes.

4.3.1 Reeb graph

An interesting concept related to Morse theory and very useful to analyze a surface topology is the Reeb graph. The latter is defined as a quotient space $\mathbb{M}/\!\sim$ with the equivalence relation given by $\boldsymbol{p} \sim \boldsymbol{q}$ if and only if $f(\boldsymbol{p}) = f(\boldsymbol{q})$ and $\boldsymbol{p}, \boldsymbol{q}$ belong to the same connected component of $f^{-1}(f(\boldsymbol{p}))$. In particular, each connected component is represented by a point in the Reeb graph as illustrated in Figure 4.4 in the case f is a height function h given by $h(x, y, z) = z$ for all $\boldsymbol{p} = (x, y, z) \in \mathbb{M}$. Mathematically, a quotient space $\mathbb{M}/\!\sim\, = \{[\boldsymbol{p}] : \boldsymbol{p} \in \mathbb{M}\}$ is the set of equivalence classes of the relation $\sim$, where $[\boldsymbol{p}] = \{\boldsymbol{q} \in \mathbb{M} : \boldsymbol{q} \sim \boldsymbol{p}\}$ is the equivalence class of $\boldsymbol{p} \in \mathbb{M}$. Intuitively, $\mathbb{M}/\!\sim$ is a space created by taking the space $\mathbb{M}$ and gluing $\boldsymbol{p}$ to any $\boldsymbol{q}$ that satisfies $\boldsymbol{q} \sim \boldsymbol{p}$. The classes $[\boldsymbol{p}]$ are the connected components for the Reeb graph, and being in the same component is an equivalence relation:

$$\boldsymbol{q} \sim \boldsymbol{p} \Longleftrightarrow f(\boldsymbol{q}) = f(\boldsymbol{p}) \text{ and } \boldsymbol{p}, \boldsymbol{q} \in \mathcal{C}\{f^{-1}(f(\boldsymbol{p}))\},$$

where $\mathcal{C}\{\cdot\}$ denotes the connected component.

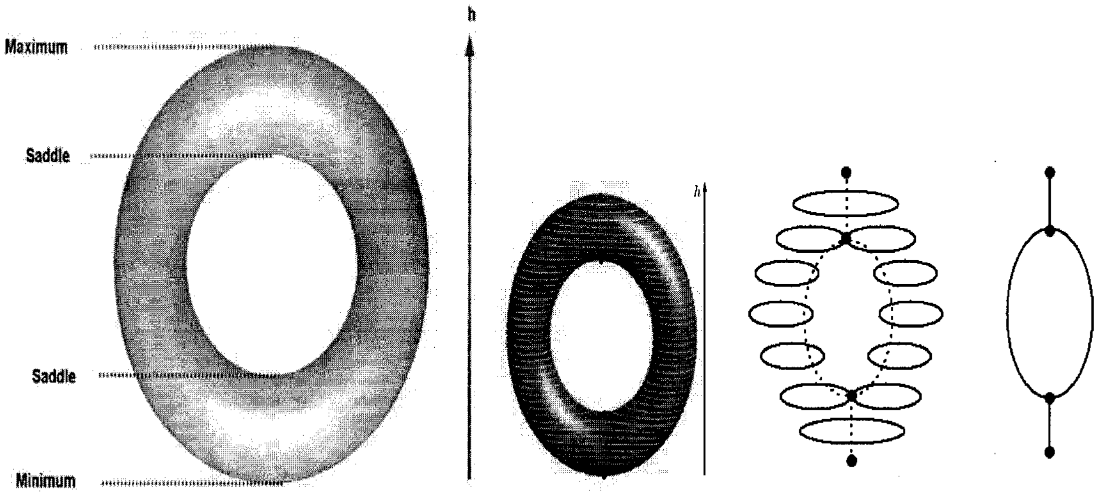


Figure 4.4: Reeb graph representation of a torus using the Morse height function.

Shinagawa *et al.* [35] have studied Reeb graphs using the height function. Since the level sets of the height function are horizontal planes perpendicular to the height axis, the weakness of such Reeb graphs is that they are not invariant to rotation. Lazarus *et al.* [36]

used the geodesic distance from a manually chosen source point as a Morse function to compute their extracted graphs which they referred to as *level set diagrams.* Hilaga *et al.* [35] used the geodesic distance from point to point on a surface to overcome the problem of automatic extraction of the source point. The computation of the geodesic distance, however, involves the Dijkstra's algorithm [35, 36] which takes approximately 90% of the overall computation time [35].

In the next section, we propose a normalized distance function based approach for topology coding to overcome the problems mentioned above.

4.4 Proposed Approach

4.4.1 Distance function

The concept of distance is of paramount importance to topology, with the actual numeric values being of less importance. In fact, topologists often use a distance function, but the attributed numerical values have only secondary meaning. To illustrate this, suppose we are given an object in the ordinary three-dimensional space, and a point outside the object, and the question is: does the object come arbitrarily close to this reference point? this may be stated as: is the point a boundary point of the object? “Arbitrarily close” means that if one imagines a ball around the reference point, then the ball contains some points belonging to the object no matter how small the ball is. The actual distances between the points belonging to the object and the reference point do not matter, and there just have to be arbitrarily small values among them.

The distance function is a function which has nondegenerate critical points, and it can be shown that almost all distance functions are Morse functions. In fact, for fixed $\boldsymbol{c} \in \mathbb{R}^3$, we may define a distance function of $\mathbb{M}$ to $\boldsymbol{c}$ as $d_{\boldsymbol{c}} : \mathbb{M} \to \mathbb{R}$ such that $d_{\boldsymbol{c}}(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{c}\|^2$.

Proposition 4.2 *The distance function* $d_{\boldsymbol{c}} : \mathbb{M} \to \mathbb{R}$ *defined by* $d_{\boldsymbol{c}}(\boldsymbol{p}) = \|\boldsymbol{p}-\boldsymbol{c}\|^2$ *is rotation and translation invariant.*

Proof: Let R be a rotation matrix and $\boldsymbol{t}$ a translation parameter, then we have

$$\begin{aligned} d_{R\boldsymbol{c}+\boldsymbol{t}}(R\boldsymbol{p}+\boldsymbol{t}) &= \|R\boldsymbol{p}+\boldsymbol{t}-(R\boldsymbol{c}+\boldsymbol{t})\|^2 \\ &= \|R(\boldsymbol{p}-\boldsymbol{c})\|^2 \\ &= (\boldsymbol{p}-\boldsymbol{c})^T R^T R(\boldsymbol{p}-\boldsymbol{c}) \\ &= \|\boldsymbol{p}-\boldsymbol{c}\|^2 \\ &= d_{\boldsymbol{c}}(\boldsymbol{p}). \end{aligned}$$

4.4.2 Morse-theoretic analysis of the distance function

A surface $\mathbb{M}$ may be defined locally in parametric form by a Monge patch $\boldsymbol{r} : \mathbb{M} \to \mathbb{R}$ such that the neighborhood of each point $\boldsymbol{p} = \boldsymbol{r}(x, y) \in \mathbb{M}$ may be defined as a graph of a function $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$. In other words, there exists $(x, y) \in \Omega$ such that $\boldsymbol{p} = \boldsymbol{r}(x, y) = (x, y, u(x, y))$. Hence the distance function may be expressed as

$$d_{\boldsymbol{c}}(\boldsymbol{p}) = d_{\boldsymbol{c}}(\boldsymbol{r}(x, y)) = \|\boldsymbol{r}(x, y) - \boldsymbol{c}\|^2.$$

Taking the derivatives with respect to x and y, the first partial derivatives are given by $d_x = 2\boldsymbol{r}_x \cdot (\boldsymbol{r}(x, y) - \boldsymbol{c})$ and $d_y = 2\boldsymbol{r}_y \cdot (\boldsymbol{r}(x, y) - \boldsymbol{c})$. Hence d has a critical point at

$\boldsymbol{p} = \boldsymbol{r}(x, y)$ if and only if $\boldsymbol{c} - \boldsymbol{p}$ is orthogonal to $\mathbb{M}$ at $\boldsymbol{p}$, i.e. $\boldsymbol{c} - \boldsymbol{r}(x, y)$ is parallel to the surface normal $\boldsymbol{N}$. Thus $\boldsymbol{c} = \boldsymbol{r}(x, y) + a\,\boldsymbol{N}$, where a is constant.

The distance function from the origin of a coordinate system (i.e. $\boldsymbol{c} = (0, 0, 0)$) is given by

$$d(x, y) = \|\boldsymbol{r}(x, y)\|^2 = x^2 + y^2 + u(x, y)^2.$$

The gradient and Hessian matrix of d are given by

$$\nabla d(x, y) = 2[(x, y) + u(x, y)\nabla u(x, y)]$$

and

$$\nabla^2 d(x, y) = 2[(1 + \|\nabla u(x, y)\|^2)I_2 + u(x, y)\,\nabla^2 u(x, y)],$$

where I_2 is the 2×2 identity matrix. The second partial derivatives at a critical point can be easily derived as

$$d_{xx} = 2(\boldsymbol{r}_x \cdot \boldsymbol{r}_x + \boldsymbol{r}_{xx} \cdot (\boldsymbol{r}(x, y) - \boldsymbol{c})) = 2(\boldsymbol{r}_x \cdot \boldsymbol{r}_x - a\,\boldsymbol{r}_{xx} \cdot \boldsymbol{N}).$$

Hence, the Hessian matrix may be expressed in matrix form as follows

$$\nabla^2 d = 2(\boldsymbol{I} - a\,\boldsymbol{II}),$$

where $\boldsymbol{I}$ and $\boldsymbol{II}$ are the first and second fundamental forms respectively [34]. A degenerate critical point of the distance function satisfies $\det(\nabla^2 d) = 0$ if and only if

$$\det(\nabla^2 d) = 1/a^2 = \kappa_1 \kappa_2,$$

where κ_1 and κ_2 are the principal curvatures [34]. A point $\boldsymbol{p} \in \mathbb{M}$ is therefore a degenerate critical point of the distance function $d_{\boldsymbol{c}}$ if and only if $\boldsymbol{c}$ is a focal point of $(\mathbb{M}, \boldsymbol{p})$. In addition, the Morse index of a nondegenerate critical point of the distance function $d_{\boldsymbol{c}}$ is equal to the number of focal points of $(\mathbb{M}, \boldsymbol{p})$ which lie on the segment from $\boldsymbol{p}$ to $\boldsymbol{c}$. This can be shown using the Hessian matrix $\nabla^2 d$ since the number of its negative eigenvalues is equal to the number of eigenvalues of the $\boldsymbol{I\!I}$ (assuming that $\boldsymbol{I}$ is the identity matrix) which are $\geq 1/a$.

4.4.3 Properties of the distance function

Without loss of generality we choose $\boldsymbol{c}$ to be the centroid of the surface $\mathbb{M}$, and for simplicity we consider the centroid to be the origin of the Euclidean coordinate system as pictured in Figure 4.5. Hence the distance function becomes

$$d(\boldsymbol{p}) = \|\boldsymbol{p}\|^2 = x^2 + y^2 + z^2,$$

where $\boldsymbol{p} = (x, y, z)$. Note that for $r > 0$, the level sets $\{\boldsymbol{p} \in \mathbb{M} : d(\boldsymbol{p}) = r^2\}$ of the distance function are concentric spheres of radii r. The key idea behind using the distance function is to track the changes in topology as we cross a surface singularity. In the first step, we start with a sphere having a sufficiently small radius, and centered as the barycenter of the underlying surface, then we evolve this sphere by increasing its radius so that we will have a set of concentric spheres covering the entire surface as illustrated in Figure 4.5.

The most important properties of the distance function are:

(i) the level sets $\{\boldsymbol{p} \in \mathbb{M} : d(\boldsymbol{p}) = r^2\}$ of the distance function are concentric spheres of radii r,

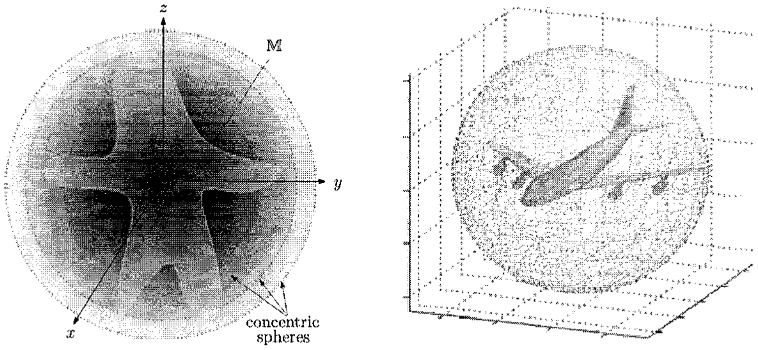


Figure 4.5: Illustration of the distance function.

(ii) a 3D object can be reconstructed if we know its intersections with these concentric spheres as depicted in Figure 4.5.

(iii) the distance function is rotation and translation invariant and can be easily normalized to achieve scale invariance.

4.4.4 Normalized distance function

3D objects are usually represented as polygonal or triangle meshes in computer graphics and geometric-aided design. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, and $\mathcal{T} = \{T_1, \ldots, T_n\}$ is the set of triangles.

Given a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, we define the normalized distance function as

$$\bar{d}(\boldsymbol{p}) = \frac{d(\boldsymbol{p}) - d_{\min}}{d_{\max} - d_{\min}}, \quad \forall \boldsymbol{p} \in \mathcal{V},$$

where $d_{\min} = \min d(\boldsymbol{p})$ and $d_{\max} = \max d(\boldsymbol{p})$, $\forall \boldsymbol{p} \in \mathcal{V}$.

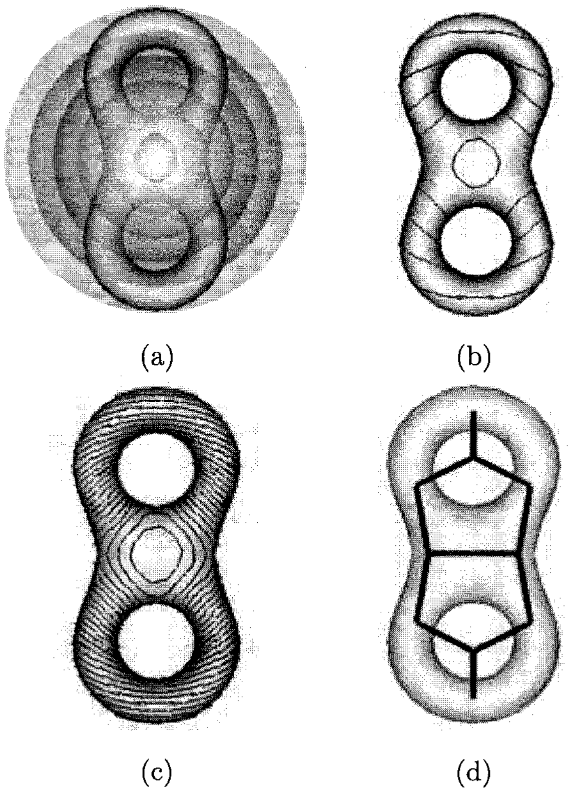
It can easily shown that the normalized distance function is scale-invariant, that is

$\overline{d}(s\boldsymbol{p}) = \overline{d}(\boldsymbol{p}), \forall s \in \mathbb{R}$. Even with the normalized form, calculating the distance function for a given surface point is simple and computationally inexpensive. In addition, it follows from Sard's theorem [44] that almost all distance functions are Morse functions. It is important to understand how the distance function relates to topology and Morse theory. The intersection of an object with a growing sphere results in closed curves, or connected components, on the object surface (see Figure 5.3(b-c)). The critical points of the surface, which define its topology, correspond to distance levels, at which there is a change in number of connected components. This change in number of connected components reflects the changes in shape, particularly, branching and merging of the object or holes within. The Reeb graph is depicted in Figure 5.3(d).

4.5 Simulations Results and Discussions

Using the normalized distance function, we constructed the Reeb graphs of two object models (horse and frog) which are shown in Figure 4.7 and Figure 4.8. The horse model in Figure 4.7(a) is decimated 51% and 78% to obtain the models depicted in Figure 4.7(b) and Figure 4.7(c). Finally, the 3D frog in Figure 4.8(a) is rotated -45 degrees to obtain the model shown in Figure 4.8(b). The model illustrated in Figure 4.8(c) was obtained by rotating the original model 90 degrees and scaling it by a factor of 2.

Next, we discuss the results of our proposed technique based on the following criteria:


(a) (b)

(c) (d)

Figure 4.6: (a) Concentric spheres evolving on a double torus, (b) intersection of the double torus and spheres, (c) intersection with spheres at 25 levels of the distance function, (h) Reeb graph.

Decimation of 3D mesh models:

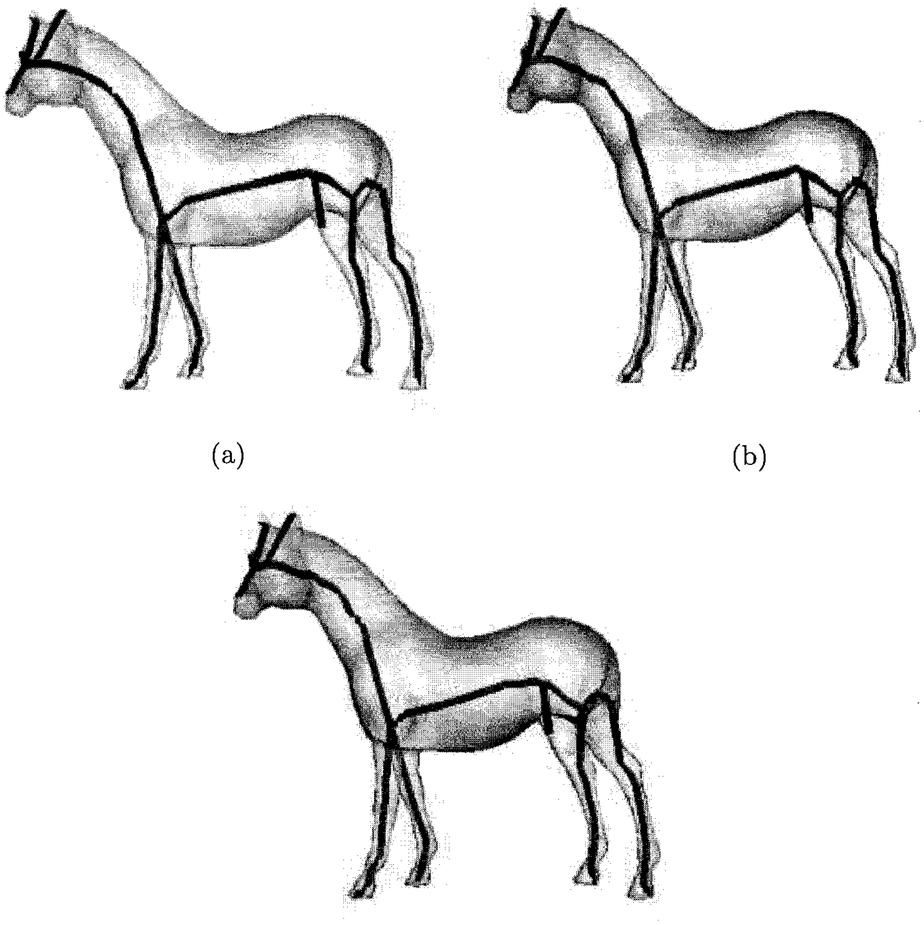
The goal of decimation is to reduce the total number of faces in a triangle mesh, while preserving topology. The three horse meshes shown in Fig. 4.7(a)-4.7(c) contain 39698, 19316, and 8556 triangular faces. As can be seen, the decimation has not altered the shape of the horse, and the Reeb graphs of the decimated meshes are good approximations of the original mesh graph. In summary, as long as decimation does not lead to changes in topology, the corresponding distance function based Reeb graph will still preserve the features of the object.

Invariance to rotation, translation, and scaling:

As a result of the distance function being rotationally invariant, distance function based Reeb graphs also demonstrate the same property. Figure 4.8(a) and Figure 4.8(b) clearly illustrate that rotating an object, prior to the construction of the Reeb graph, does not change the critical points of the distance function. Their corresponding Reeb graphs are identical, and this property does not hold when using the height function. Moreover, by introducing the normalized distance function, we guarantee that different sizes of the same object are modeled with the same Reeb graph. Figure 4.8(c) shows the Reeb graph of the 3D frog model scaled by a factor of 2 but at the same resolution as Figure 4.8(a).

Computational complexity:

Computing the centroid and the distance function for a 3D triangle mesh with n vertices takes $\mathcal{O}(n)$ time. Constructing the nodes and edges of the Reeb graph requires calculating the connected component of triangles and hence also takes $\mathcal{O}(n)$ time. The overall complexity is, therefore, $\mathcal{O}(n)$, which shows an improvement over geodesic function based Reeb graphs with complexity $\mathcal{O}(n \log n)$ [6].



(a)

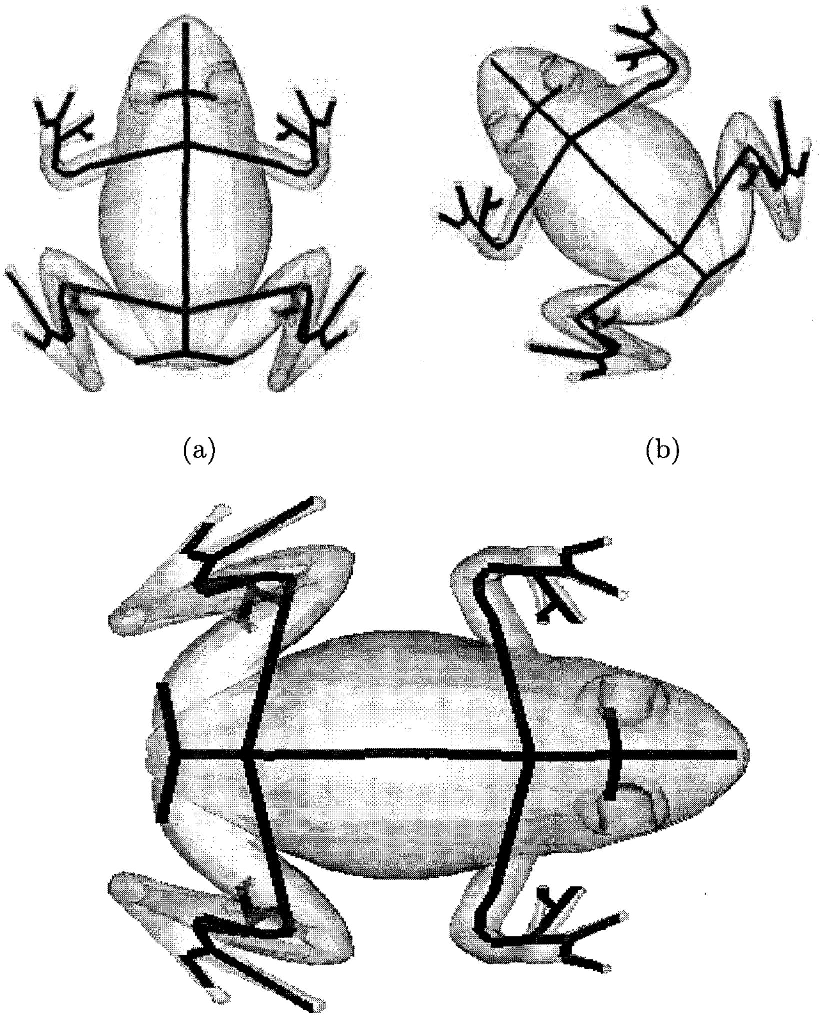
(b)

(c)

Figure 4.7: Reeb graphs of a 3D horse model (a) with 39698 triangle faces, (b) with 19316 triangle faces, (c) with 8556 triangle faces.

4.6 Conclusions

In this chapter, we proposed a distance function-based approach to topological coding of 3D objects in the Morse-theoretic framework. Using a distance function, we constructed invariant Reeb graphs of three dimensional objects. The main attractive properties of



(a) (b)

(c)

Figure 4.8: Reeb graphs of (a) a frog model; model rotated (b) -45 degrees and (c) 90 degrees around y-axis and scaled by 2 prior to Reeb graph construction.

the proposed approach are: invariance to rotation, translation, and scaling; robustness to decimation, and low computational complexity. Future work will be focused on using the distance function based Reeb graph as a shape signature for 3D object matching, as well as the reconstruction of objects from the level sets of the distance function.

CHAPTER 5

Multiresolution Mixture Skeletal Graph

We present a multiresolution skeletal graph for topological coding of 3D shapes. The proposed skeletonization algorithm encodes a 3D object into a topological graph using a normalized mixture distance function. The approach is accurate, invariant to Euclidean transformations, robust to mesh decimation, computationally efficient, and preserves topology. Experimental results demonstrate the potential of the proposed topological graph which may be used as a shape signature for 3D object matching and retrieval, and also for skeletal animation.

5.1 Introduction

The importance of 3D shape recognition is increasing rapidly in the field of computer graphics and multimedia communication due to the difficulty in processing information efficiently without its recognition. With the increasing use of scanners to create 3D models which are usually represented as triangle meshes in computer graphics and geometric-aided design, shape recognition of 3D objects has become an active research field with the recent developments in solid modeling and visualization [44]. Today, vast amounts of 3D models are being developed and are distributed freely or commercially on the internet. 3D objects consist of geometric and topological information, and their compact representation is an important step towards a variety of computer graphics applications including indexing, retrieval, animation, and matching in a database of 3D models. The first step in 3D object matching usually involves finding a reliable shape descriptor or skeletal graph which will encode efficiently the 3D shape information. Skeletonization aims at reducing the dimensionality of a 3D shape while preserving its topology [44, 57], and has a variety of computer graphics applications including object recognition and matching [35, 41], collision detection [54], mesh editing [52], and animation [53].

Most 3D shape representation techniques proposed in the literature of computer graphics, computer vision and computer-aided design are based on geometric/topological representations which represent the features of an object [28, 35, 36, 38–42, 42, 43]. An alternative to feature-based representations is global methods which represent a 3D object by a global measure or shape distribution defined on the surface of the object [58, 59].

Our proposed approach aims to represent 3D objects with topological coding. Topology represents the connectedness of a shape and enables parts of shapes, which are connected, to be mapped and drawn equivalently. One of the key mathematical tools used to study the topology of spaces is Morse theory which is the study of the relationship between functions on a space and the shape of the space. Morse studies the properties of a Morse function which has only nondegenerate singular points [44], and it describes the topology changes of the level sets of this function at those singularities. Regular or noncritical points do not affect the number or genus of the components of the level sets. It can be shown that Morse functions are dense and stable in the set of all smooth functions, that is the structure of nondegenerate singularities does not change under small perturbations. A Morse theory-based representation that captures topological properties of objects is the so-called Reeb graph representation proposed in [28, 29]. The vertices of the Reeb graph are the singular points of the height function defined on the surface of 3D object [28, 29, 44]. The height function-based approach may lead to the extraction of an unbounded number of critical points. This limitation has been addressed in [43] by introducing a fair Morse function that produces the least possible number of critical points. Since the level sets of the height function are horizontal planes perpendicular to the height axis, the weakness of such Reeb graphs is that they are not invariant to rotation. Lazarus *et al.* [36] used the geodesic distance from a manually chosen source point as a Morse function to compute their extracted graphs which they referred to as *level set diagrams*. Hilaga *et al.* [35] used the geodesic distance from point to point on a surface to overcome the problem of automatic extraction of the source point. The computation of the geodesic distance, however, involves the Dijkstra's algorithm [35,36] which takes almost 90% of the overall computation time [35].

In this chapter, we propose an invariant multiresolution skeletal graph for topology coding using a normalized mixture distance function. The key idea is to identify and encode regions of topological interest of a 3D object in the Morse-theoretic framework [30, 55, 56].

The main motivations behind using the distance function are: (a) rotational invariance which makes it more adapted to object recognition than the Morse height function, and (b) it can be shown that a surface may be reconstructed from its intersections with concentric superellipsoids centered at the barycenter of the underlying surface. The topological changes occur as we change the level values of these concentric superellipsoids. At singular points, the level sets of the mixture distance function may split or merge which indicate topological changes.

The remainder of this chapter is organized as follows. The next section briefly describes the basic concepts of Morse theory, followed by a mathematical description of the Reeb graph representation for 3D topological modeling. In Section 5.3, we propose a normalized mixture distance function-based approach to construct invariant skeletal graphs of 3D objects. Section 5.4 is devoted to the algorithmic steps of the proposed approach. In Section 5.5, we present experimental results for topological coding using the mixture distance function-based skeletal graph, and we discuss its most attractive properties which include robustness to mesh decimation, invariance to rigid motion transformations, and computational efficiency. And finally, in Section 5.6, we summarize our results and point out future directions.

5.2 Morse theory for Topological Modeling

In computer graphics and geometric-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} =$

$\{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges, and $\mathcal{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ is the set of triangles.

5.2.1 Morse theory

Morse theory explains the presence and the stability of singular points in terms of the topology of the underlying smooth manifold. Topology is the property that determines which parts of the shape of objects are connected to which other parts [33, 34], while geometry determines where, in a given coordinate system, each part is located [34]. The basic principle is that the topology of a manifold is very closely related to the singular points of a smooth function defined on that manifold [30]. A smooth function $f : \mathbb{M} \rightarrow \mathbb{R}$ on a smooth manifold $\mathbb{M}$ is called a *Morse function* if all its singular points are nondegenerate, i.e. the Hessian matrix is nonsingular at every singular point.The only nondegenerate singularities are the minimum, maximum and saddle points. Nondegenerate singularities are isolated, that is, there cannot be a sequence of nondegenerate singularities converging to a nondegenerate singularity $\boldsymbol{p} \in \mathbb{M}$. A *level set* $f^{-1}(a)$ of f at a value a may be composed of one or many connected components, such as closed curves on the surface. Morse deformation lemma states that if no critical points exist between two level sets of f, then the two level sets are topologically equivalent and can be deformed onto one another [4]. In particular, they consist of the same number of connected components [1]. Furthermore, Morse theory implies that topological changes on the level sets occur only at critical points [3]. This property is illustrated next.

5.2.2 Level sets around Morse critical points

Let $f : \mathbb{M} \to \mathbb{R}$ be a Morse function defined on a compact surface $\mathbb{M}$, and denote by $\mathbb{M}_a$ the sub-surface of $\mathbb{M}$ consisting of all points at which f takes values less than or equal to a real number a

$$\mathbb{M}_a = \{\boldsymbol{p} \in \mathbb{M} : f(\boldsymbol{p}) \leq a\}.$$

Figure 5.1 shows the evolution of the subsurface $\mathbb{M}_a$ as a changes, when f is a height function. If $a < \min_{\boldsymbol{p} \in \mathbb{M}}\{f(\boldsymbol{p})\}$, then $\mathbb{M}_a = \emptyset$. And as we increase the parameter a, the subsurface $\mathbb{M}_a$ changes until it covers the entire surface $\mathbb{M}$. We may think of the height function $f : \mathbb{M} \to \mathbb{R}$ as dipping a doughnut into a cup of chocolate cream.

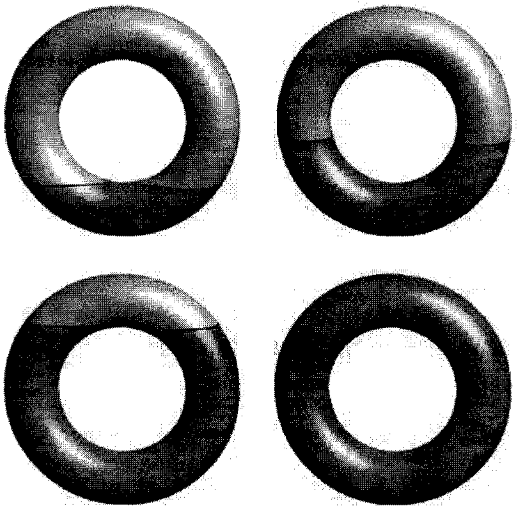


Figure 5.1: Evolution of $\mathbb{M}_a$ as a changes.

5.2.3 Reeb graph

An interesting concept related to Morse theory and very useful to analyze a surface topology is the Reeb graph. The latter is defined as a quotient space $\mathbb{M}/\sim$ with the equivalence relation given by $\boldsymbol{p} \sim \boldsymbol{q}$ if and only if $f(\boldsymbol{p}) = f(\boldsymbol{q})$ and $\boldsymbol{p}, \boldsymbol{q}$ belong to the same connected component of $f^{-1}(f(\boldsymbol{p}))$. Intuitively, $\mathbb{M}/\sim$ is a space created by taking the space $\mathbb{M}$ and gluing $\boldsymbol{p}$ to any $\boldsymbol{q}$ that satisfies $\boldsymbol{q} \sim \boldsymbol{p}$. The classes $[\boldsymbol{p}]$ are the connected components for the Reeb graph, and being in the same component is an equivalence relation:

$$\boldsymbol{q} \sim \boldsymbol{p} \Longleftrightarrow f(\boldsymbol{q}) = f(\boldsymbol{p}) \text{ and } \boldsymbol{p}, \boldsymbol{q} \in \mathcal{C}\{f^{-1}(f(\boldsymbol{p}))\},$$

where $\mathcal{C}\{\cdot\}$ denotes the connected component.

In the next section, we propose a normalized mixture distance function based approach for topology coding of 3D shapes.

5.3 Proposed Approach

The concept of distance is of paramount importance to topology, with the actual numeric values being of less importance. In fact, topologists often use a distance function, but the attributed numerical values have only secondary meaning. To illustrate this, suppose we are given an object in the ordinary three-dimensional space, and a point outside the object, and the question is: does the object come arbitrarily close to this reference point? this may be stated as: is the point a boundary point of the object? "Arbitrarily close" means that if one imagines a ball around the reference point, then the ball contains some points belonging to the object no matter how small the ball is.

Given a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, and let $\boldsymbol{p}_i = (x_i, y_i, z_i)^T \in \mathcal{V} = \{\boldsymbol{p}_1, \ldots, \boldsymbol{p}_m\}$. Denote by $\boldsymbol{V} = (\boldsymbol{p}_1\ \boldsymbol{p}_2\ \ldots \boldsymbol{p}_m)^T$ the $m \times 3$ mesh vertex matrix having as rows the coordinates of the mesh vertices, that is

$$\boldsymbol{V} = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{pmatrix} \in \mathbb{R}^{m\times 3}.$$

Let $\boldsymbol{c} = (\bar{x}, \bar{y}, \bar{z})^T$ be the centroid of the triangle mesh, that is $\boldsymbol{c} = (\sum_{i=1}^{m} \boldsymbol{p}_i)/m$. We define the $m \times 3$ centered vertex matrix as

$$\begin{aligned} V_{\boldsymbol{c}} &= (\boldsymbol{p}_1 - \boldsymbol{c}\ \boldsymbol{p}_2 - \boldsymbol{c}\ \ldots\ \boldsymbol{p}_m - \boldsymbol{c})^T \\ &= \begin{pmatrix} x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\ x_2 - \bar{x} & y_2 - \bar{y} & z_2 - \bar{z} \\ \vdots & \vdots & \vdots \\ x_m - \bar{x} & y_m - \bar{y} & z_m - \bar{z} \end{pmatrix}. \end{aligned}$$

The Euclidean distance function of $\mathbb{M}$ to $\boldsymbol{c}$ is defined as

$$d_{\boldsymbol{c}}^{euc} : \mathbb{M} \to \mathbb{R} \quad \text{such that} \quad d_{\boldsymbol{c}}^{euc}(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{c}\|^2,$$

and it can be easily shown that it is rotation and translation invariant [39].

Let $A = m(V_c^T V_c)^{-1}$, we define the affine distance function as follows

$$d_{\boldsymbol{c}}^{aff} : \mathbb{M} \to \mathbb{R} \quad \text{such that} \quad d_{\boldsymbol{c}}^{aff}(\boldsymbol{p}) = \|\boldsymbol{p} - \boldsymbol{c}\|_A^2 = (\boldsymbol{p} - \boldsymbol{c})^T A (\boldsymbol{p} - \boldsymbol{c}),$$

and it can be shown that it is invariant to affine transformations [37].

5.3.1 Mixture distance function

For fixed $\boldsymbol{c} \in \mathbb{R}^3$, we define the mixture distance function as a convex combination of the Euclidean and the affine distance functions:

$$d_{\boldsymbol{c}} : \mathbb{M} \to \mathbb{R} \quad \text{such that} \quad d_{\boldsymbol{c}}(\boldsymbol{p}) = \lambda\|\boldsymbol{p}-\boldsymbol{c}\|^2 + (1-\lambda)\|\boldsymbol{p}-\boldsymbol{c}\|_A^2,$$

where $\lambda \in (0,1)$ is a mixture parameter that needs to be estimated or chosen *a priori.* From the invariance properties of the Euclidean and affine distance functions, it is easy to verify that the mixture distance function is invariant to orthogonal and translation transformations.

5.3.2 Morse-theoretic analysis of the mixture distance function

A surface $\mathbb{M}$ may be defined locally in parametric form by a Monge patch $\boldsymbol{r} : \mathbb{M} \to \mathbb{R}$ such that the neighborhood of each point $\boldsymbol{p} = \boldsymbol{r}(x,y) \in \mathbb{M}$ may be defined as a graph of a function $u : \Omega \subset \mathbb{R}^2 \to \mathbb{R}$. In other words, there exists $(x,y) \in \Omega$ such that $\boldsymbol{p} = \boldsymbol{r}(x,y) = (x,y,u(x,y))$. Hence the mixture distance function may be expressed as

$$d_{\boldsymbol{c}}(\boldsymbol{p}) = d_{\boldsymbol{c}}(\boldsymbol{r}(x,y)) = \lambda\|\boldsymbol{r}(x,y)-\boldsymbol{c}\|^2 + (1-\lambda)\|\boldsymbol{r}(x,y)-\boldsymbol{c}\|_A^2.$$

Taking the derivatives of $d_{\boldsymbol{c}}$ with respect to x and y, the first partial derivatives are given by $d_x = \boldsymbol{r}_x^T B_\lambda(\boldsymbol{r}-\boldsymbol{c})$ and $d_y = \boldsymbol{r}_y^T B_\lambda(\boldsymbol{r}-\boldsymbol{c})$, where $B_\lambda = \lambda I_3 + (1-\lambda)(A + A^T)$, and I_3 is the 3×3 identity matrix. Thus $d_{\boldsymbol{c}}$ has a critical point at $\boldsymbol{p} = \boldsymbol{r}(x,y)$ if and only if $B_\lambda(\boldsymbol{p}-\boldsymbol{c})$ is orthogonal to $\mathbb{M}$ at $\boldsymbol{p}$, or equivalently $\boldsymbol{c} - \boldsymbol{r}(x,y)$ is parallel to the surface normal $\boldsymbol{N}$. Therefore

$$\boldsymbol{c} = \boldsymbol{p} + \alpha B_\lambda^{-1}\boldsymbol{N} = \boldsymbol{r}(x,y) + \alpha B_\lambda^{-1}\boldsymbol{N},$$

where α is a constant.

On the other hand, the second order partial derivatives of $d_{\boldsymbol{c}}$ at a critical point are given by

$$d_{xx} = \boldsymbol{r}_x^T B_\lambda \boldsymbol{r}_x + \boldsymbol{r}_{xx}^T B_\lambda (\boldsymbol{r} - \boldsymbol{c}) = \boldsymbol{r}_x^T B_\lambda \boldsymbol{r}_x - \alpha \boldsymbol{r}_{xx}^T \boldsymbol{N}$$

$$d_{yy} = \boldsymbol{r}_y^T B_\lambda \boldsymbol{r}_y + \boldsymbol{r}_{yy}^T B_\lambda (\boldsymbol{r} - \boldsymbol{c}) = \boldsymbol{r}_y^T B_\lambda \boldsymbol{r}_y - \alpha \boldsymbol{r}_{yy}^T \boldsymbol{N}$$

$$d_{xy} = \boldsymbol{r}_x^T B_\lambda \boldsymbol{r}_y + \boldsymbol{r}_{xy}^T B_\lambda (\boldsymbol{r} - \boldsymbol{c})) = \boldsymbol{r}_x^T B_\lambda \boldsymbol{r}_y - \alpha \boldsymbol{r}_{xy}^T \boldsymbol{N}.$$

Hence, the Hessian matrix is given by $\nabla^2 d = \boldsymbol{I} - \alpha \boldsymbol{II}$, where $\boldsymbol{I}$ and $\boldsymbol{II}$ are the first and second fundamental forms [34] with respect to the orthogonal basis $\{\sqrt{B_\lambda}\,\boldsymbol{r}_x, \sqrt{B_\lambda}\,\boldsymbol{r}_y\}$.

A degenerate critical point of the mixture distance function satisfies $\det(\nabla^2 d_{\boldsymbol{c}}) = 0$ if and only if

$$\det(\nabla^2 d_{\boldsymbol{c}}) = 1/\alpha^2 = \kappa_1 \kappa_2,$$

where κ_1 and κ_2 are the principal curvatures [34]. A point $\boldsymbol{p} \in \mathbb{M}$ is therefore a degenerate critical point of the mixture distance function $d_{\boldsymbol{c}}$ if and only if $\boldsymbol{c}$ is a focal point of $(\mathbb{M}, \boldsymbol{p})$, that is $\boldsymbol{c} = \boldsymbol{p} + \kappa_1^{-1} \boldsymbol{N}$ or $\boldsymbol{c} = \boldsymbol{p} + \kappa_2^{-1} \boldsymbol{N}$. If $\boldsymbol{c}$ is the origin of the coordinate system, then it is clear that $\boldsymbol{c}$ is not a focal point and hence $d_{\boldsymbol{c}}$ has no degenerate critical points. In addition, the Morse index of a nondegenerate critical point of the mixture distance function $d_{\boldsymbol{c}}$ is equal to the number of focal points of $(\mathbb{M}, \boldsymbol{p})$ which lie on the segment from $\boldsymbol{p}$ to $\boldsymbol{c}$. This can be shown using the Hessian matrix $\nabla^2 d_{\boldsymbol{c}}$ since the number of its negative eigenvalues is equal to the number of eigenvalues of the second fundamental form matrix $\boldsymbol{II}$ (assuming that the first fundamental form $\boldsymbol{I}$ is the identity matrix) which are $\geq 1/\alpha$.

5.3.3 Properties of the mixture distance function

Without loss of generality we choose $\boldsymbol{c}$ to be the centroid of the surface $\mathbb{M}$, and for simplicity we consider the centroid to be the origin of the Euclidean coordinate system. Hence the mixture distance function becomes

$$d(\boldsymbol{p}) = \lambda\|\boldsymbol{p}\|^2 + (1-\lambda)\|\boldsymbol{p}\|_A^2.$$

Note that for $\ell > 0$, the level sets $\{\boldsymbol{p} \in \mathbb{M} : d(\boldsymbol{p}) = \ell\}$ of the mixture distance function are concentric superellipsoids as illustrated in Figure 5.2. The key idea behind using the mixture distance function is to track the changes in topology as we cross a surface singularity. In the first step, we start with a superellipsoid having a sufficiently small level value ℓ, and centered as the barycenter of the underlying surface, then we evolve this superellipsoid by increasing the level value so that we will have a set of concentric superellipsoids covering the entire surface.

The most important properties of the mixture distance function are:

(i) the level sets $\{\boldsymbol{p} \in \mathbb{M} : d(\boldsymbol{p}) = \ell\}$ of the distance function are concentric superellipsoids.

(ii) a 3D object can be reconstructed if we know its intersections with these concentric superellipsoids.

(iii) the mixture distance function is rotation and translation invariant and can be easily normalized to achieve scale invariance as shown next.

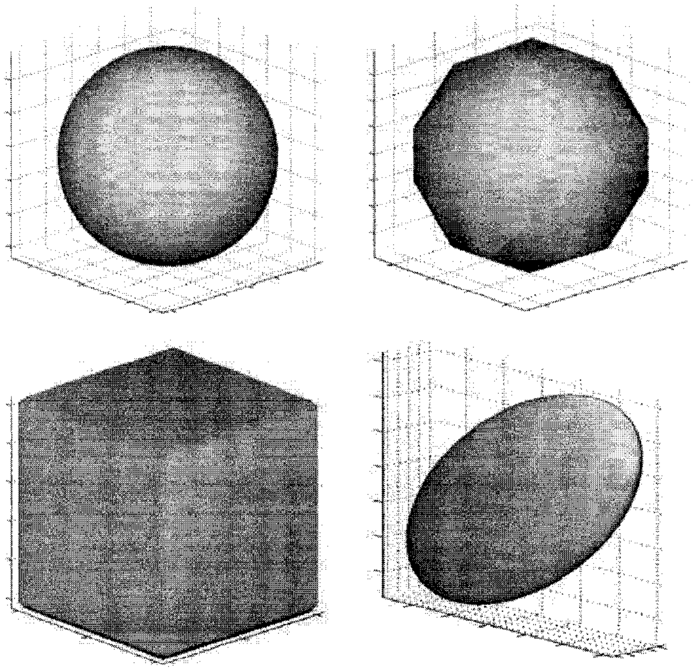


Figure 5.2: Level sets of the mixture distance function.

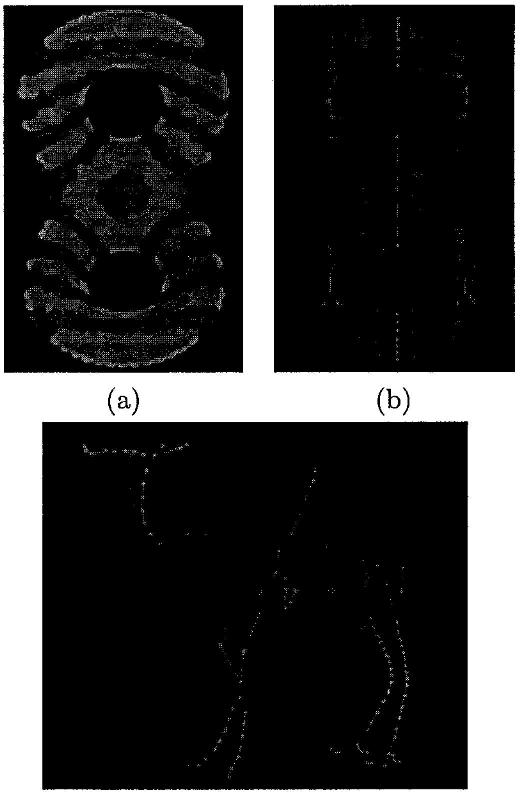
5.3.4 Normalized mixture distance function

Given a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, we define the normalized mixture distance function as

$$\widetilde{d}(\boldsymbol{p}) = \frac{d(\boldsymbol{p}) - d_{\min}}{d_{\max} - d_{\min}}, \quad \forall \boldsymbol{p} \in \mathcal{V},$$

where $d_{\min} = \min d(\boldsymbol{p})$ and $d_{\max} = \max d(\boldsymbol{p})$, $\forall \boldsymbol{p} \in \mathcal{V}$.

It can easily be shown that the normalized mixture distance function is scale-invariant, that is $\widetilde{d}(s\boldsymbol{p}) = \widetilde{d}(\boldsymbol{p}), \forall s \in \mathbb{R}$. Even with the normalized form, calculating the mixture distance function for a given surface point is simple and computationally inexpensive. In addition, it follows from Sard's theorem [44] that almost all distance functions are Morse functions. It is important to understand how the mixture distance function relates to topology and Morse theory. The intersection of a 3D object with a growing superellipsoid results in connected components on the object surface (see Figure 5.3(a)). The critical points of the surface, which define its topology, correspond to distance levels, at which there is a change in number of connected components (see Figure 5.3(b)). This change in number of connected components reflects the changes in topology of a 3D shape, particularly, branching and merging or holes within as shown in Figure 5.3(b). The skeletal graphs of a double torus and a camel model are depicted in Figures 5.3(b)-(c). These topological skeletons were extracted using the algorithm described in the next section.


(a) (b)

(c)

Figure 5.3: (a)-(b) Connected components and skeletal graph of a double torus, (c) skeletal graph of camel model.

Algorithm: Skeletonization of 3D shapes

```
1. $c$ = centroid($\mathbb{M}$); $\Leftarrow$ Find the centroid of the triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ as the arithmetic mean of the mesh vertices.
2. Find the maximum distance $D_{\max} = \max dist(c, \mathbb{M})$ from the centroid.
3. For ($k = 1$ to $R$)
     $d(k) = k * D_{\max}/R$; $\Leftarrow$ where $R$ is a predefined integer called the resolution parameter of the skeletal graph.
4. VerticesSet$_p$[0,1] = setIntersect($\mathbb{M}$,1); $\Leftarrow$ Find the vertices subset of $\mathbb{M}$ from $c$ to distance $d(1)$
5. NodeSet$_p$ = centroid(VerticesSet$_p$[0,1](n)); $\Leftarrow$ Assign a node to each connected component at its centroid.
6. Connect $c$ and NodeSet$_p$
7. For $k = 2$ to $R$
     VerticesSet$_c$[$k-1, k$] = setIntersect($\mathbb{M}, k-1, k$);$\Leftarrow$ Find the intersection of $\mathbb{M}$ from distance $d(k-1)$ to $d(k)$
     For each connected component VerticesSet$_c$ [$k-1, k$]($n$)
          NodeSet$_c$ = centroid(VerticesSet$_c$ [$k-1, k$]($n$))
          For each connected portion
               Connect NodeSet$_c$ and NodeSet$_p$
          End For
     End For
     NodeSet$_p$ = NodeSet$_c$
     VerticesSet$_p$ = VerticesSet$_c$
8. End For
```

Table 5.1: Algorithmic steps of the proposed approach.

5.4 Skeletonization algorithm

Let $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ be a triangle mesh, and for simplicity we consider the centroid c of $\mathbb{M}$ to be the origin of the Euclidean coordinate system. The main algorithmic steps of the mixture distance-based skeletal graph are described in Table 5.1. In Figure 5.4 the work flow of the proposed skeletonization algorithm is shown step by step to illustrate the skeletal graph extraction, where the VerticesSet and NodeSet are marked as green and red points respectively.

The complexity of the proposed algorithm can be determined as follows. Computing the centroid and the normalized mixture distance function for a 3D triangle mesh with m vertices takes $\mathcal{O}(m)$ time. Constructing the nodes and edges of the skeletal graph requires calculating the connected component of triangles and hence also takes $\mathcal{O}(m)$ time. The overall complexity is, therefore, $\mathcal{O}(m)$, which shows an improvement over geodesic function based Reeb graphs with complexity $\mathcal{O}(m \log m)$ [35].

5.5 Experimental Results

We now present examples of skeletal graphs for 3D shapes using the proposed algorithm. In all the experiments, the parameter λ of the mixture distance function is set to $\lambda = ||\boldsymbol{p}||^2/(||\boldsymbol{p}||^2 + ||\boldsymbol{p}||_A^2)$. Using the Skeletonization algorithm described in the previous section, we constructed the skeletal graphs of several 3D models as shown in Figure 5.5. The results clearly indicate the accuracy of the extracted skeletons.

Next, we show the robustness of the proposed skeletal graph to mesh decimation and its invariance to Euclidean transformations. Also, we present comparison results with the Euclidean distance-based graph and we discuss the choice of the resolution parameter.

5.5.1 Decimation of 3D mesh models

The goal of mesh decimation is to reduce the total number of faces in a triangle mesh while preserving topology. The original heart model shown in Figure 5.6(d) contains 25600 triangles. The three decimated heart meshes shown in Figures 5.6(a)-5.6(c) contain 400, 1600, and 6400 triangular faces respectively. Note that the skeletal graphs of the decimated meshes are good approximations of the original mesh graph. Moreover, when the optimal resolution R is achieved, then increasing or decreasing the number of faces will not affect the skeletal graph significantly. In summary, as long as decimation does not lead to dramatic changes in topology, the corresponding skeletal graphs will accurately capture the topology of 3D objects. Moreover, it is also worth pointing out that the size of a 3D model is crucial in finding the optimal resolution R of a skeletal graph. The larger a 3D mesh is, the higher resolution can be obtained. The resolution parameter used in Figure 5.6 is $R = 5$.

5.5.2 Invariance to rotation, translation, and scaling

Figure 5.7(a) and Figure 5.7(b) clearly illustrate that rotating a 3D spider model, prior to the construction of the skeletal graph, does not change the critical points of the mixture distance function. Their corresponding skeletal graphs are identical. Moreover, by introducing the normalized mixture distance function, we guarantee that different dimensions of the same object are modeled with the same skeletal graph. Figure 5.7(c) shows the skeletal graph of the 3D spider model scaled by a factor of 2 but at the same resolution as Figure 5.7(a).

5.5.3 Comparison results

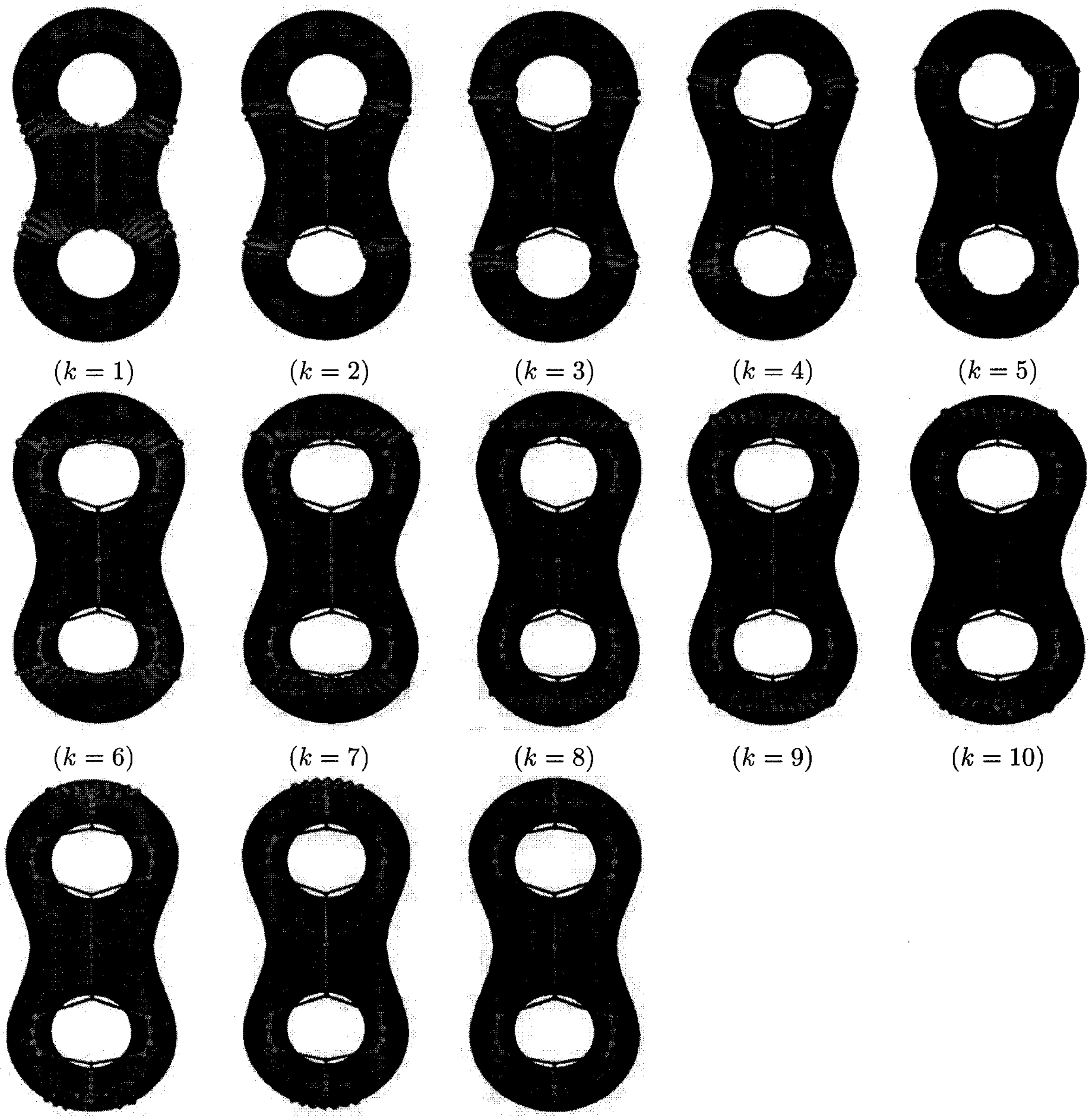
Figures 5.8-5.12 show the comparison results between the Euclidean and mixture distance-based approaches. As can be seen, the mixture distance function-based approach provides more accurate results while preserving topology.

5.5.4 Effect of the resolution parameter

Our current selection of the resolution parameter R is based on experimentation. The optimal choice for the user-defined parameter R still needs to be further investigated. However, we noticed through extensive experimentation with a variety of 3D models that for relatively large models the resolution parameter $R = 22$ provides satisfactory results in terms of skeleton extraction accuracy, computational complexity, and topology preservation. Figures 5.13-5.14-5.15 illustrate the skeletal graphs at different resolution levels.

5.6 Conclusions

In this chapter, we introduced a normalized mixture distance function-based approach to topological coding of 3D objects in the Morse-theoretic framework. The proposed algorithm preserves efficiently the topology of 3D shapes, and it is robust, accurate, and has a low computational complexity. The main attractive properties of the proposed multiresolution approach are: invariance to rotation, translation, and scaling; and robustness to mesh decimation. We illustrated the approach with several examples of skeletal graphs for a variety of 3D objects. Future work will be focused on using this skeletal graph as a shape signature for 3D object matching and retrieval, and also for skeletal animation. Theoretically we hope to develop more rigorous way of finding the optimal resolution parameter of the proposed algorithm.



($k = 1$) ($k = 2$) ($k = 3$) ($k = 4$) ($k = 5$)

($k = 6$) ($k = 7$) ($k = 8$) ($k = 9$) ($k = 10$)

($k = 11$) ($k = 12$) ($k = 13$)

Figure 5.4: Skeletal graph extraction at each step $k = 1, \dots, R$, where $R = 13$

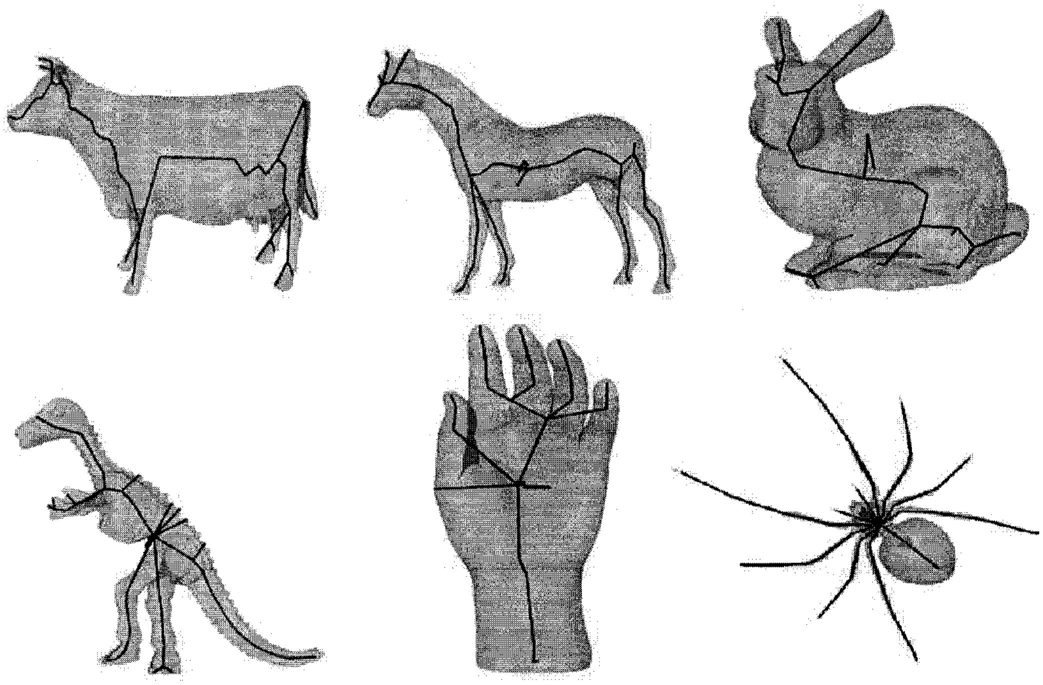
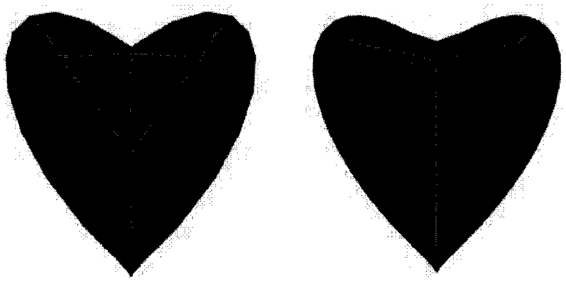
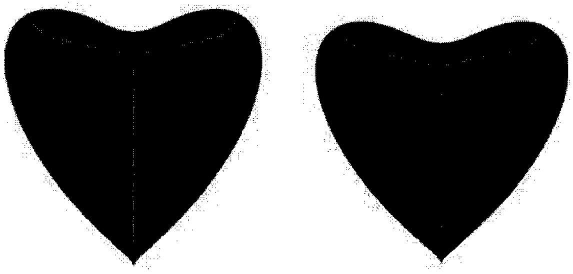


Figure 5.5: Skeletal graphs of different 3D models.

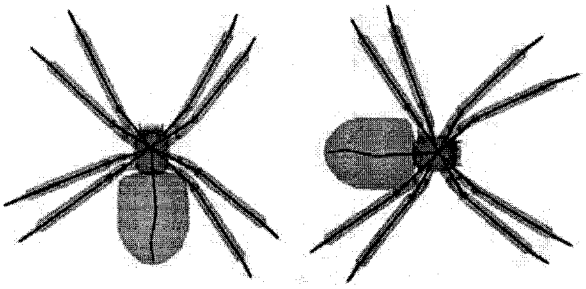


(a) 400 triangles (b) 1600 triangles

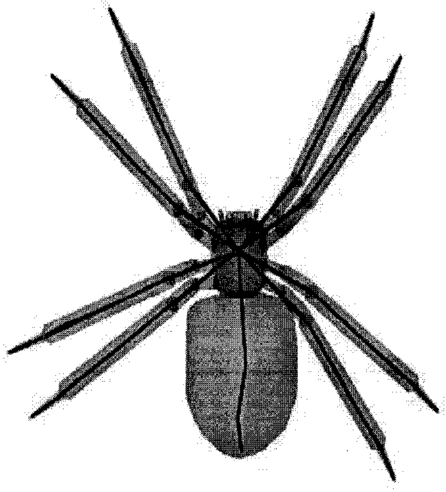


(c) 6400 triangles (d) 25600 triangles

Figure 5.6: Skeletal graph under mesh decimation.

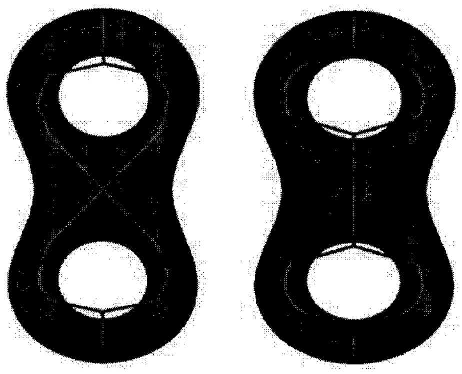


(a) original (b) rotated by 90°



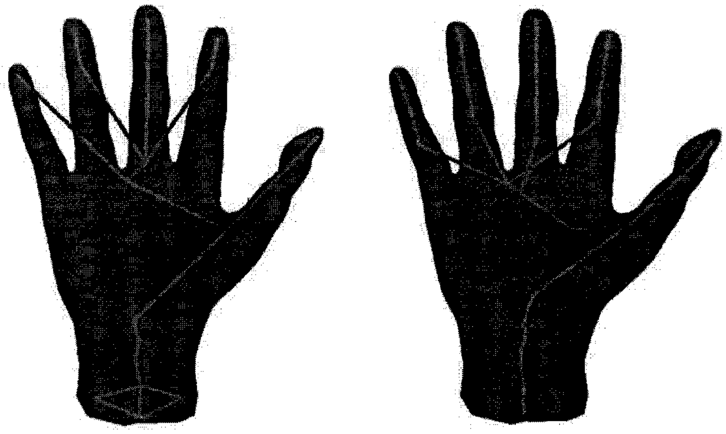
(a) scaled by a factor of 2

Figure 5.7: Illustration of skeletal graph invariance to rotation and scaling



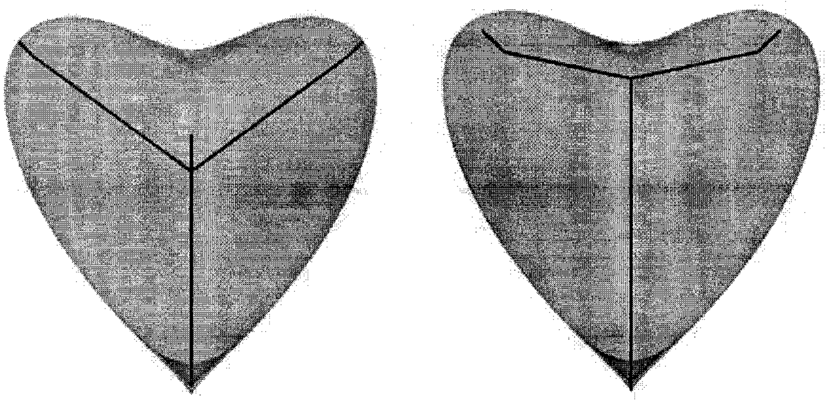
(a) Euclidean (b) mixture

Figure 5.8: Skeletal graphs using normalized Euclidean and mixture distance functions



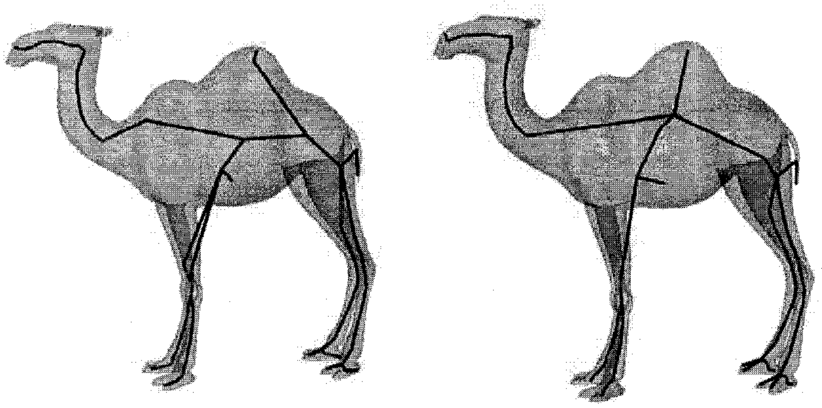
(a) Euclidean (b) mixture

Figure 5.9: Skeletal graphs using normalized Euclidean and mixture distance functions



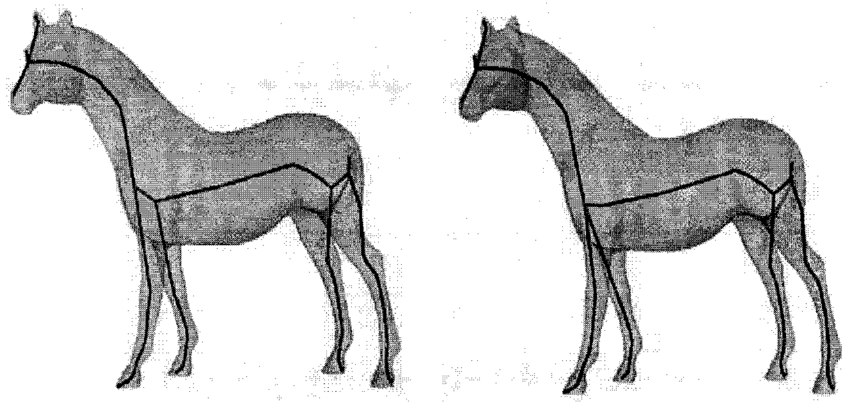
(a) Euclidean (b) mixture

Figure 5.10: Skeletal graphs using normalized Euclidean and mixture distance functions



(a) Euclidean (b) mixture

Figure 5.11: Skeletal graphs using normalized Euclidean and mixture distance functions



(a) Euclidean (b) mixture

Figure 5.12: Skeletal graphs using normalized Euclidean and mixture distance functions

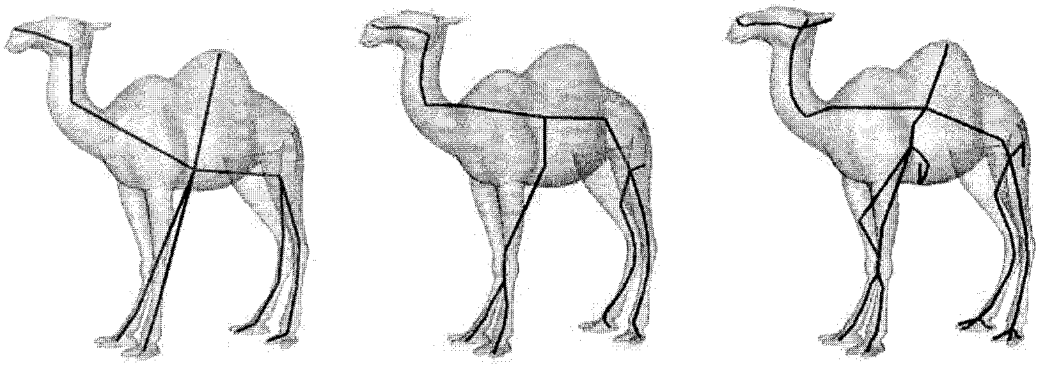

($R = 5$) ($R = 13$) ($R = 22$)

Figure 5.13: Skeletal graph at different resolution levels.

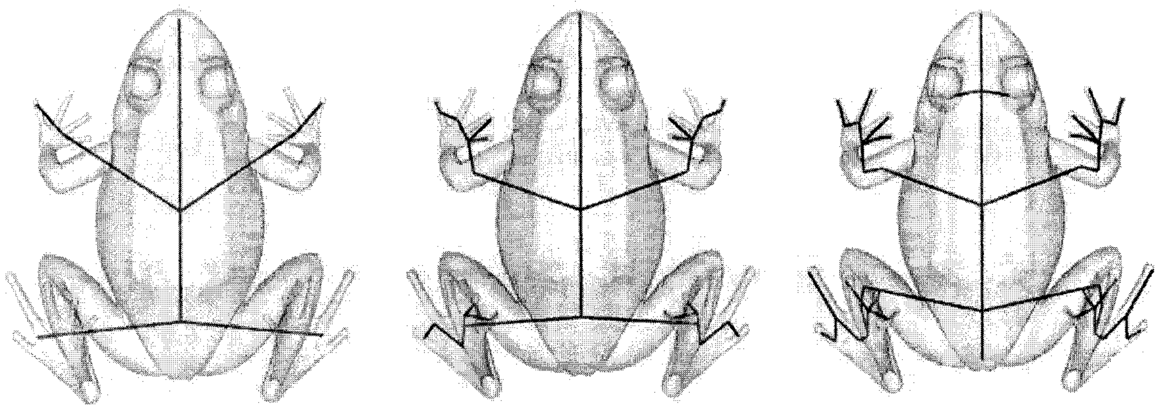
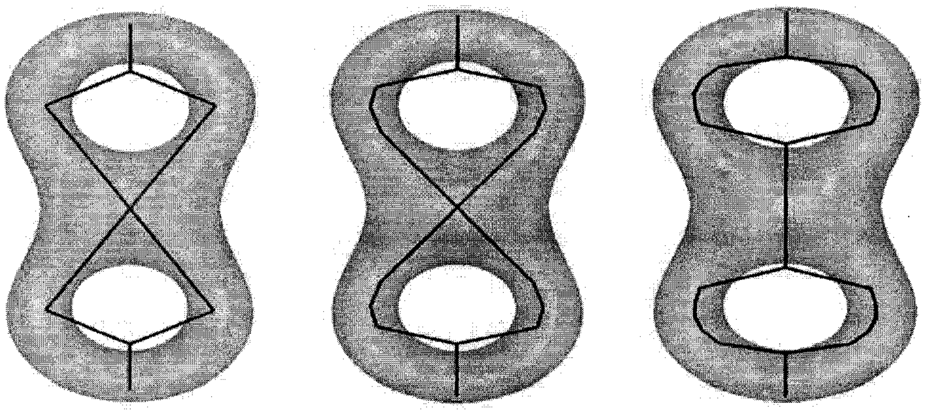

($R = 5$) ($R = 13$) ($R = 22$)

Figure 5.14: Skeletal graph at different resolution levels.



$(R = 5)$ $(R = 13)$ $(R = 22)$

Figure 5.15: Skeletal graph at different resolution levels.

CHAPTER 6

Conclusions and Future Work

This thesis has presented computational algorithms for variational 3D mesh denoising, and topological modeling of 3D shapes using Morse theory. We have demonstrated the use of these algorithms through a variety of computer graphics applications including mesh denoising, topological modeling of triangle meshes, and distance function-based object representation. The geometric/topological algorithms are tailored for the normalized mixture distance-based skeletal graph of 3D shapes. We have demonstrated the effectiveness of the proposed methods through numerical simulations with synthetic and real data in 3D computer graphics.

In the next Section, the contributions made in each of the previous chapters and the concluding results drawn from the associated research work are presented. Suggestions for future research directions related to this thesis are provided in Section 6.2.

6.1 Contributions of the thesis

6.1.1 Vertex-based anisotropic mesh denoising

We introduced a vertex-based anisotropic diffusion for 3D mesh denoising by solving a nonlinear discrete partial differential equation. The core idea behind our proposed technique is to use geometric insight in helping construct an efficient and fast 3D mesh smoothing strategy to fully preserve the geometric structure of the 3D mesh data. The experimental results clearly showed a much improved performance of the proposed approach in comparison with the current methods used in 3D mesh smoothing. For future work, we plan to incorporate the curvature information as well as additional regularization terms into the proposed model.

6.1.2 Statistical 3D mesh distributions

We proposed several statistical measures to analyze the topological properties of 3D models. The proposed statistical measures include the mesh degree, the mesh assortativity, the mesh clustering coefficient, and the mesh geodesic distance distributions. The experimental results clearly showed the effectiveness of the proposed measures in quantifying the topological features of 3D objects.

6.1.3 Euclidean distance-based skeletal graph

We proposed a distance function-based approach to topological coding of 3D objects in the Morse-theoretic framework. Using a distance function, we constructed invariant Reeb graphs of three dimensional objects. The main attractive properties of the proposed approach are: invariance to rotation, translation, and scaling; robustness to decimation, and low computational complexity. Future work will be focused on using the distance function based Reeb graph as a shape signature for 3D object matching, as well as the reconstruction of objects from the level sets of the distance function.

6.1.4 Multiresolution mixture skeletal graph

We introduced a mixture distance function-based approach to topological coding of 3D objects in the Morse-theoretic framework. The proposed algorithm preserves efficiently the topology of 3D shapes, and it is robust, accurate, and has a low computational complexity. The main attractive properties of the proposed multiresolution approach are: invariance to rotation, translation, and scaling; and robustness to mesh decimation. We illustrated the approach with several examples of skeletal graphs for a variety of 3D objects. Future work will be focused on using this skeletal graph as a shape signature for 3D object matching and retrieval, and also for skeletal animation. Theoretically we hope to develop more rigorous way of finding the optimal resolution parameter of the proposed algorithm.

6.2 Future research directions

Several interesting research directions motivated by this thesis are discussed next. In addition to designing new methodologies for geometric smoothing and modeling of 3D graphics, we intend to accomplish the following projects in the near future:

6.2.1 3D Object recognition

Recently we have been working on the representation, matching, indexing and retrieval in a database of 3D objects based on the topological and geometric information. Building a database requires collecting 3D models, computing their Reeb graph representations, and indexing in the database based on an abstracted information given by their skeletal graphs. An appropriate and efficient representation of the Reeb graph is the attributed Reeb graph that represents topology and geometry in a compact representation, where vertices and edges have geometric attributes. In other words, we associate to the graph as much geometric information as possible that will be attached to the graph for further tasks such as matching, indexing and retrieval.

6.2.2 Skeletal graph animation

Skeletal animation technology provides the illusion of natural motion for computer-generated characters in motion pictures and computer games. Many video games developers use skeletal graph animation techniques where a 3D model is represented by two parts. The first part is the mesh (also referred to as *skin*) of the model and the second part is the skeleton. The skin represents the surface of the model and may contain hundreds of thousands of vertices, and the skeleton is a set of bones that deform the skin as they move.

From a simplistic point of view, a skeleton may be thought of as a collection of rigid bones connected together by revolute joints (points of articulation that allow the rotation of connected bones about one or more axes). This structure is one that we, as humans, observe and manipulate constantly in the real world. Thus it would seem reasonable to suggest that skeletal motion, simulated adequately, would be a natural and realistic physical metaphor. A human body may be viewed a 3D model where the skin represents the mesh of the model and the bones would be the skeleton of the model. When the skeleton moves, the skin moves as well, and hence the mesh on the outside of a model moves with the bones. Skeletal graph animation allows 3D animators to animate models much faster. Rather than animating thousands of points to make a character walk, they just need to manipulate the leg bones, and the leg mesh will automatically deform.

List of References

[1] P. Perona and J. Malik, "Scale space and edge detection using anisotropic diffusion," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 12, no. 7, pp. 629-639, 1990.

[2] L. Rudin, S. Osher, and E. Fatemi, "Nonlinear total variation based noise removal algorithms," *Physica D*, vol. 60, pp. 259-268, 1992.

[3] Y.L. You, W. Xu, A. Tannenbaum, and M. Kaveh, "Behavioral Analysis of anisotropic diffusion in image processing," *IEEE Trans. Image Processing*, vol. 5, no. 11, pp. 1539-1553, 1996.

[4] P. Charbonnier, L. Blanc-Féraud, G. Aubert, and M. Barlaud, "Deterministic edge-preserving regularization in computed imaging," *IEEE Trans. Image Processing*, vol. 6, no. 2, pp. 298-311, 1997.

[5] J. Weickert, *Anisotropic diffusion in image processing*, Stuttgart, Germany: Teubner-Verlab, 1998.

[6] A. Yezzi, "Modified curvature motion for image smoothing and enhancement," *IEEE Trans. Image Processing*, vol. 7, no. 3, pp. 345-352, 1998.

[7] P. Kornprobst, R. Deriche, and G. Aubert, "Image sequence analysis via partial differential equations," *J. Math. Imag. Vision*, vol. 11, no. 1, pp. 5-26, 1999.

[8] C. Samson, L. Blanc-Féraud, G. Aubert, and J. Zerubia, "A variational model for image classification and restoration," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, pp. 460-472, 2000.

[9] M. Cetin and W.C. Karl, "Feature-enhanced synthetic aperture radar image formation based on nonquadratic regularization," *IEEE Trans. Image Processing*, vol 10, no. 4, pp. 623-631, 2001.

[10] Y. Bao and H. Krim, "Smart nonlinear diffusion: a probabilistic approach," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 63-72, 2004

[11] M. Giaquinta and S. Hildebrandt, *Calculus of Variations I: The Lagrangian Formalism*, Springer-Verlag, 1996.

[12] G. Taubin, "A signal processing approach to fair surface design," *Proc. SIGGRAPH*, pp. 351-358, 1995.

[13] S. Petitjean "A survey of methods for recovering quadrics in triangle meshes," *ACM Computing Surveys*, vol. 34, no. 2, pp. 211-262, 2002.

[14] H. Yagou, Y. Ohtake, and A. Belyaev, "Mesh smoothing via mean and median filtering applied to face normals," *Proc. Geometric Modeling and Processing*, pp. 124-131, 2002.

[15] S. Fleishman, I. Drori, and D. Cohen-Or, "Bilateral mesh denoising," *Proc. ACM SIGGRAPH*, pp. 950-953, 2003.

[16] T. Jones, F. Durand, and M. Desbrun, "Non-iterative, feature preserving mesh smoothing," *Proc. SIGGRAPH*, pp. 943-949, 2003.

[17] M. Desbrun, M. Meyer, P. Schröder, and A. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," *Proc. SIGGRAPH*, pp. 317324, 1999.

[18] M. Desbrun, M. Meyer, P. Schröder, and A. Bar, "Anisotropic feature-preserving denoising of height fields and bivariate data," *Graphics Interface*, pp. 145-152, 2000.

[19] U. Clarenz, U. Diewald, and M. Rumpf, "Anisotropic geometric diffusion in surface processing," *Proc. IEEE Visualization*, 2000.

[20] U. Clarenz, U. Diewald, and M. Rumpf, "Processing textured surfaces via anisotropic geometric diffusion," *IEEE Transactions on Image Processing*, vol. 13, no.2, pp. 248-261, 2004.

[21] C.L. Bajaj, and G. Xu, "Anisotropic diffusion of subdivision surfaces and functions on surfaces," *ACM Transactions on Graphics*, vol.22 no.1, pp.4-32, January 2003.

[22] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface smoothing via anisotropic diffusion of normals," *Proc. IEEE Visualization*, pp. 125-132, 2002.

[23] T. Tasdizen, R. Whitaker, P. Burchard, and S. Osher, "Geometric surface processing via normal maps," *ACM Transactions on Graphics*, vol. 22, no. 4, pp. 1012-1033, 2003.

[24] K. Hildebrandt and K. Polthier, "Anisotropic filtering of non-linear surface features," *Computer Graphics Forum*, vol. 23, no. 3, 2004.

[25] F.R. Chung, *Spectral Graph Theory*, American Mathematical Society, 1997.

[26] W.J. Rey, *Introduction to robust and quasi-robust statistical methods*, Springer, Berlin, Heidelberg, 1983.

[27] A.T. Fomenco and T.L. Kunii, *Topological modeling for visualization* (Tokyo: Springer-Verlag, 1997).

[28] Y. Shinagawa, T.L. Kunii, and Y.L. Kergosien, "Surface coding based on Morse theory," *IEEE Comp. Graph. and Appl.*, 11(5), 1991, 66-78.

[29] Y. Shinagawa and T.L. Kunii, "Constructing a Reeb graph automatically from cross sections," *IEEE Comp. Graph. and Appl.*, 11(6), 1991, 44-51.

[30] J. Milnor, *Morse theory* (New Jesey: Princeton University Press, 1963).

[31] J.C. Hart, "Morse theory for implicit surface modeling," *Proc. Visualization and Mathematics*, 1998, 257-268.

[32] J.C. Hart, "Computational topology for shape modeling," *Proc. Shape Modeling International*, 1999, 36-45.

[33] V. Guillemin and A. Pollack, *Differential topology* (New Jesey: Prentice-Hall, Inc., Englewood Cliffs, 1974).

[34] M. do Carmo, *Differential geometry of curves and surfaces* (New Jesey: Prentice-Hall, 1976).

[35] M. Hilaga, Y. Shinagawa, T. Komura, and T.L. Kunii, "Toplogical matching for fully automatic similarity estimation of 3D shapes," *Proc. ACM SIGGRAPH*, 2001, 203-212.

[36] F. Lazarus and A. Verroust, "Level set diagrams of polyhedral objects," *Proc. ACM Symposium on Solid Modeling and Applications*, 1999, 130-140.

[37] G.M. Nielson and T.A. Foley, "A survey of applications of an affine invariant norm," *Mathematical Methods in Computer Aided Geometric Design*, Academic Press, Boston, pages 445-467, 1989.

[38] M.S. Hassouna and A.A. Farag, "Robust centerline extraction framework using level sets," *Proc. IEEE CVPR*, pages 458-465, 2005.

[39] S. Baloch, H. Krim, I. Kogan, and D. Zenkov, "Rotation invariant topology coding of 2D and 3D objects using Morse theory," *Proc. IEEE ICIP*, pages 796-799, 2005.

[40] N.D. Cornea, M.F. Demirci, D. Silver, A. Shokoufandeh, S.J. Dickinson, and P.B. Kantor, "3D Object Retrieval using Many-to-many Matching of Curve Skeletons," *Proc. IEEE International Conference on Shape Modeling and Applications*, pages 366-371, 2005.

[41] P. Dimitrov, C. Phillips, and K. Siddiqi, "Robust and efficient skeletal graphs," *Proc. CVPR*, pages 417-423, 2005.

[42] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson, "Skeleton based shape matching and retrieval," *Proc. Shape Modeling International*, pages 130-139, 2003.

[43] X. Ni, M. Garland, and J.C. Hart, "Fair morse functions for extracting the topological structure of a surface mesh," *ACM Transactions on Graphics*, pages 613-622, 2004.

[44] A.T. Fomenco and T.L. Kunii, *Topological modeling for visualization* (Tokyo: Springer-Verlag, 1997).

[45] C. Bettstetter, "On the minimum node degree and connectivity of a wireless multihop network," *ACM Press*, pp. 80-91, 2002.

[46] M. E. J. Newman, "The structure and function of complex networks," *SIAM Review*, vol. 45, pp. 167-256, 2003.

[47] A. Barrat, M. Barthelemy, and A. Vespignani, "Modeling the evolution of weighted networks," *Physical Review E*, vol. 70, 2004.

[48] P. Mahadevan, D. Krioukov, M. Fomenkov, B. Huffaker, X. Dimitropoulos, K.C. Claffy, and A. Vahdat, "The internet AS-level topology: three data sources and one definitive metric," *ACM SIGCOMM Computer Communication Review*, vol. 36, no. 1, pp. 17-26, 2006.

[49] H. Fuks and C. Phipps, "Toward a model of language acquisition threshold," *Proc. IASTED Conference on Modelling and Simulation*, Montreal, Canada, May 2006.

[50] T.F. Cox and M.A. Cox, *Multidimensional scaling*, second edition, Monographs on Statistics and Applied Probability, vol. 88, 2001.

[51] J.B. Tenenbaum, V. de Silva, and J.C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, pp. 2319-2323, December 2000.

[52] X.T. Li, T.W. Woon, T.S. Tan, Z.Y. Huang, "Decomposing polygon meshes for interactive applications," *Proceedings of ACM Symposium on Interactive 3D Graphics*, pages 35-42, 2001.

[53] L. Wade, R.E. Parent, "Automated generation of control skeletons for use in animation," *The Visual Computer*, pages 97-110, 2002.

[54] V. Macagon and B. Wünsche, "Efficient collision detection for skeletally animated models in interactive environments," *Proc. Image and Vision Computing, NZ*, pages 378-383, 2003.

[55] H. Edelsbrunner, J. Harer, and A. Zomorodian, "Hierarchical Morse complexes for piecewise linear 2-manifolds," *Proc. 17th Sympos. Comput. Geom.*, pages 70-79, 2001

[56] D. Cohen-Steiner, H. Edelsbrunner, and J. Harer, "Stability of persistence diagrams," *Proc. 21st Sympos. Comput. Geom.*, pages 263-271, 2005.

[57] T. Grigorishin, G. Abdel-Hamid, Y.H. Yang, “Skeletonization: an electrostatic field-based approach,” *Pattern Analysis and Applications*, pages 163-177, 1998.

[58] M. Ankerst, G. Kastenmüller, H. Kriegel, and T. Seidl, “3D shape histograms for similarity search and classification in spatial databases,” *Advances in Spatial Databases, 6th International Symposium*, pages 207226, 1999.

[59] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, “Matching 3d models with shape distributions,” *Shape Modeling International*, pages 154166, 2001.