國立交通大學

資訊工程學系

博士論文

三維多邊形處理、檢索及驗證方法

3-D Mesh Processing, Retrieval, and Authentication

研究生：林學億

指導教授：廖弘源　教授

林志青　教授

中華民國九十五年七月

三維多邊形處理、檢索及驗證方法
3-D Mesh Processing, Retrieval, and Authentication

研究生：林學億　　Student：Hsueh-Yi Sean Lin

指導教授：廖弘源　　Advisors：Hong-Yuan Mark Liao

林志青　　Ja-Chen Lin

國立交通大學
資訊工程系
博士論文

A Dissertation

Submitted to Department of Computer Science

College of Computer Science

National Chiao Tung University

in Partial Fulfillment of the Requirements

for the Degree of

Doctor of Philosophy

in

Computer Science

July 2006

Hsinchu, Taiwan, Republic of China

中華民國九十五年七月

國立交通大學

博碩士論文全文電子檔著作權授權書

（提供授權人裝訂於紙本論文書名頁之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學 資訊工程 系，九十四 學年度第 二 學期取得博士學位之論文。

論文題目：三維多邊形處理、檢索及驗證方法

指導教授：廖弘源及林志青

■ 同意　□不同意

本人茲將本著作，以非專屬、無償授權國立交通大學與台灣聯合大學系統圖書館：基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學及台灣聯合大學系統圖書館得不限地域、時間與次數，以紙本、光碟或數位化等各種方法收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行線上檢索、閱覽、下載或列印。

論文全文上載網路公開之範圍及時間：	
本校及台灣聯合大學系統區域網路	■中華民國 95 年 7 月 18 日公開
校外網際網路	■中華民國 95 年 7 月 18 日公開

授 權 人：林學億

親筆簽名：林學億

中華民國　九十五 年 七 月 十八 日

國立交通大學

博碩士紙本論文著作權授權書

（提供授權人裝訂於全文電子檔授權書之次頁用）

本授權書所授權之學位論文，為本人於國立交通大學 資訊工程 系，九十四 學年度第 二 學期取得博士學位之論文。

論文題目：三維多邊形處理、檢索及驗證方法

指導教授：廖弘源及林志青

■ 同意

本人茲將本著作，以非專屬、無償授權國立交通大學，基於推動讀者間「資源共享、互惠合作」之理念，與回饋社會與學術研究之目的，國立交通大學圖書館得以紙本收錄、重製與利用；於著作權法合理使用範圍內，讀者得進行閱覽或列印。

本論文為本人向經濟部智慧局申請專利(未申請者本條款請不予理會)的附件之一，申請文號為：____________________，請將論文延至____年____月____日再公開。

授 權 人：林學億

親筆簽名：林學億

中華民國 九十五 年 七 月 十八 日

國家圖書館博碩士論文電子檔案上網授權書

ID:GT008823818

本授權書所授權之論文為授權人在國立交通大學 資訊 學院 資訊工程 系 九十四 學年度第 二 學期取得博士學位之論文。

論文題目：三維多邊形處理、檢索及驗證方法

指導教授：廖弘源及林志青

茲同意將授權人擁有著作權之上列論文全文（含摘要），非專屬、無償授權國家圖書館，不限地域、時間與次數，以微縮、光碟或其他各種數位化方式將上列論文重製，並得將數位化之上列論文及論文電子檔以上載網路方式，提供讀者基於個人非營利性質之線上檢索、閱覽、下載或列印。

※ 讀者基於非營利性質之線上檢索、閱覽、下載或列印上列論文，應依著作權法相關規定辦理。

授權人：林學億

親筆簽名： 林學億

民國　九十五 年 七 月 十八 日

國立交通大學
研究所博士班

論文口試委員會審定書

本校 資訊工程 系 林學億 君

所提論文 三維多邊形處理、檢索及驗證方法
3-D Mesh Processing, Retrieval, and Authentication

合於博士資格水準、業經本委員會評審認可。

口試委員：

陳永昌

莊永裕 施仁忠

孫永年 莊仁輝

柳金章 廖弘源

指導教授： 廖弘源

系主任： 曾憲雄 教授

中華民國 95 年 7 月 17 日

Department of Computer Science
College of Computer Science
National Chiao Tung University
Hsinchu, Taiwan, R.O.C.

As members of the Final Examination Committee, we certify that we have read the dissertation prepared by Hsueh-Yi Sean Lin entitled 3-D Mesh Processing, Retrieval, and Authentication and recommend that it be accepted as fulfilling the dissertation requirement for the Degree of Doctor of Philosophy.

Yung-Yu Chuang

Jin-Jang Leou

Zen-Chung Shih

Mark Liao

Jen-Hui Chuang

Dissertation Advisor: Mark Liao

Chairman:

Date: 2006/7/17

三維多邊形處理、檢索及驗證方法

學生： 林學億　　　　　　　　指導教授：廖弘源　博士

林志青　博士

國立交通大學資訊工程學系(研究所)博士班

中文摘要

本論文提出了三種適於計算機圖學之相關應用的多邊形處理技術。首先，我們提出一種利用視覺感知特徵為引導的三維多邊形物件分解方法。所提方法利用視覺感知相關的特徵引導三維多邊形物件的分解過程。由於特徵擷取法係根基於認知心理學中的視覺特點理論，因此所提分解方法能夠適切地仿傚人類視覺對物件之構成要素的感知能力。透過物件的拆解/分解，有許多計算機圖學為主的應用能夠達到效能的提升。這些應用包括：碰撞偵測、熱幅射能量成像模擬、強健式傳輸與連續式傳輸、材質貼圖、三維形變、多邊形化簡與壓縮法、以骨架為主的動畫製作以及三維物件檢索系統。透過研究考察，我們發現既有的三維物件檢索系統缺乏心理學理論的支

撐與連結。反觀心裡學領域，許多認知心裡學專家已提出了許多重要的理論(或規則)闡述人類視覺對物件或型式的感知過程。有鑑於此，在第二項研究中，我們提出了一套以認知心理學為基礎的三維物件檢索系統。我們將具體化源自於認知心理學領域的概念性理論，用以設計特徵擷取及比對方法，並進而實現一套能仿傚人類對物件辨識與認知處理的三維物件檢索系統。在第三項研究中，為了讓使用者能驗證與偵測所擷取之三維物件的篡改與否，我們提出一種利用新的脆弱型浮水印技術以達到篡改偵測與驗證的目的。所提方法所嵌入之浮水印除了能抵抗一些非惡意的幾何處理(如頂點座標量化及頂點順序重組)，尚能利用視覺檢視三維多邊形物件的篡改處。

3-D MESH PROCESSING, RETRIEVAL, AND AUTHENTICATION

Student: Hsueh-Yi Sean Lin Advisors: Hong-Yuan Mark Liao and Ja-Chen Lin

Department of Computer Science

College of Computer Science

National Chiao Tung University

Abstract in English

In this dissertation, we propose three mesh processing techniques for 3-D graphics-related applications. First, we propose a novel mesh decomposition scheme called “visual salience-guided mesh decomposition,” which uses visually salient features to guide the mesh decomposition process. Since the features adopted are closely related to the psychology-based theory of visual salience, the decomposition process can appropriately mimic the function of a human visual system. There are a variety of applications that benefit from breaking up a 3-D object into components. These applications include collision detection, radiosity simulations, robust transmission and streaming, texture mapping, metamorphosis, 3-D shape retrieval, simplification and compression, watermarking, and control skeleton extraction for key-frame animation. In our investigation, the existing 3-D shape analysis and retrieval algorithms are lack of explicit link to psychology-based principles, while cognitive psychologists have found a set of principles (or properties) that are fairly important in perception of a form or a shape. This motivates us to conduct our second work called "a cognitive psychology-based approach for 3-D shape retrieval." In this work, we incorporate a set of principles, which is originated from cognitive psychology, into the design of 3-D shape retrieval system. The proposed approach is intended for mimicking human visual perception and recognition based on the psychology-based rules proposed by Hoffman and Singh. In addition, our system realizes a "recognition-by-components" and "recognition-by-visually-salient-components" search strategies. Moreover, use of combined search strategies can perform a coarse-to-fine retrieval task. In our third work, to provide a user with an ability to verify the integrity of the content he/she received (or retrieved), we propose a new fragile watermarking scheme for authenticating 3-D polygonal meshes. The proposed scheme can not only achieve localization of malicious modifications in visual inspection, but also is immune to certain incidental data processings, such as quantization of vertex coordinates and vertex reordering.

Dedication

To my fiancée, Hsiao-Shan Sharon Huang, who has been my faithful companion, unwavering supporter, sincere comforter, friend and lover through all it takes to be the person God wants me to be. I love you next to God.

And to my parents, Song-Fa and Pao-Lien Chou Lin, and my sister, May-Hsiang Carrie Lin, for their love, patience, and support in my time of studying and working.

Acknowledgements

This thesis represents seven years of work in the National Chiao Tung University. This work would not have been possible without the boundless guidance and assistance of mentors, colleagues, and friends.

First and foremost, I thank my advisor, Dr. Mark Liao, from Academia Sinica. He allowed me the freedom and the encouragement to explore, while at the same time provided excellent suggestions and invaluable insight. Any words of appreciation would be an understatement of my gratitude to him. I also thank another advisor, Dr. Ja-Chen Lin, from the National Chiao Tung University for giving me the freedome to pursue the research that led to this thesis.

I thank the members of my committee, Dr. Yung-Chang Chen, Dr. Jen-Hui Chuang, Dr. Yung-Yu Chuang, Dr. Jin-Jang Leou, Dr. Zen-Chung Shih, and Dr. Yung-Nien Sun, for their insightful comments and suggestions which improve the quality of this thesis. I also thank Dr. Jung-Hong Chuang and Dr. Hsueh-Ming Hang for their comments when I was taking my thesis prelim.

In my time at Academia Sinica, I have had the opportunity to work in collaboration with a number of different peopole without whose help this thesis could not have been completed. I would like to thank the many friends that have offered help and encouragement. In particular, I thank Dr. Chun-Shien Lu and Dr. Kuo-Young Cheng for their advices on technical issues in digital watermarking and computer graphics respectively. I also thank Jessica Su and and Jack Hsieh from the company Digibits in San Francisco for their financial support and encouragement during the long nights I had been working on the project of mesh watermarking. Additionally, I thank my colleagues, Jung-Shiong Norris Chang, Dr. Liang-Hua Chen, Dr. Duan-Yu Andrew Chen, Chun-Yen Chen, Dr. Jan-Ru Jerry Chen, Dr. Li-Fen Chen, Yi-Yun Grace Hong, Dr. Wei-Leih Hsu, Jien-Ming Lai, Yu-Ming Benson Liang, Hsin-Yuan Lo, Hsiu-Man Amy Shih, Dr. Chih-Wen

Lucas Su, Dr. Chwen-Zye Sze, and Dr. Gwo-Jong Yu. These are the people that made my years at Academia Sinica a joy.

In the last year, I have had the opportunity to conduct graphics-related research in collaboration with the colleagues at the University of Illinois at Urbana-Champaign, USA. I would like to express my gratitude to my host advisor, Dr. Yizhou Yu, who always pushes me for more, but never for more than I could. I also would like to thank my colleagues: Dr. Byoungwon Choe, Wei-Wen Feng, Dr. Lin Shih, Dr. Hongcheng Wang, Qing Wu, and Tian Xia. They have offered tireless guidance, assistance, and encouragement. Particular thanks go to the National Science Council of Taiwan, who generously sponsored me to spend the one-year research abroad. This greatly broadens my research training and pave the way for the future international collaborations.

Furthermore, I would like to thank my lovely freinds who have always brought me to joy and happiness when I could have a few days off at work. Thank you to my friends: Liang-Chao Chang, Richard Chien, Vincent Hsu, Christo Lin, Jun-You Lin, Curtis Wen, and Shih-Tung Wu. Thank you to my tennis mates at Urbana-Champaign in the US: Sandy Chang, Keng Hsu, Henry Kwong, Juno Lee, and Shu-Jung Tang. Additionally thank you to my colleages at the National Chiao Tung University: Shin-Ming Chang, Yu-Jay Chang, Dr. Shang-Kuan Chen, Dr. Chih-Ching Thien, and Dr. Ran-Zan Wang.

Finally, I sincerely offer my thanks to my brothers and sisters in Christ whose ardent support has been indispensable to me over the last year. Thank you to my spiritual mentors and counsellors: Jim Egli, Chia-Fon and Yu-Yu Lee, Steve and Ruth Wegman, and Gary Zinn. Thank you to my prayer team: Don and Jennifer Follis, Daniel Goulet, Brian Harvey, Kevin Koontz, Penny Moisson, and Rosalie Zachary. Thank you to the members in the Bible study group: Yuan-Chung Lin, Yong Liu and Yuancheng Tu, Wuheng Luo and Qi Zhou, Ya-Ting Teng, Mei-Song Tong and Wei Liu, and Liying Zhao. Thank you to the members in the men's mini-group: Niu Jin, Min Li, Yanjun Tang, and Hai Wu. Thank you to the friends that I made along the way to the Kingdom of God: Jeff and Kim Kacmarynski, Jonathan and Susan Kim, Rose Ann Black and Sue Lawson, Joshua and Shawna Martell, Sean and Hsin-Yi Marzano, Mike Moutray, Jennifer Shen, Jennie Tang, Zhimou Wen, and Mike Wilson.

Table of Contents

List of Tables

List of Figures

Chapter 1 Introduction

In this chapter, the motivation for this dissertation is first introduced. We then provide an introduction to the 3-D mesh representation, which is used throughout this dissertation. Next, we briefly introduce the proposed mesh processing techniques. Finally, the organization of this dissertation is described.

1.1 Motivation

With the fast development of hardware and software for computer graphics, graphics-based applications among a number of fields, including cartography, geographical information system (GIS), virtual reality, scientific visualization, computer-aided design (CAD), and computer-aided manufacture (CAM), widely appear in recent years. As a result of the emergence of the new multimedia type, 3-D graphical objects (or 3-D geometry), a variety of digital geometry processing (DGP) algorithms, including data creation, acquisition, storage, transmission, editing, animation, and simulation, has come out. A detailed survey can be found in [116]. Multimedia retrieval, including the retrieval of texts [2, 16, 129], images [42, 126], audios [32], videos [122, 126], and/or 3-D graphics [21, 86, 101], has been a very hot research area in the past decade. On the other hand, copyright protection and authentication of 3-D graphical objects arise from the fact that transferring digitized media via the Internet has become very popular in recent years. Most of the people believe that the invention of the Internet in early 1990's is one of the major driving forces for the prosperity of this important area. In this dissertation, we, therefor, put our emphasis on 3-D mesh processing, retrieval, and authentication.

In this dissertation, we propose three mesh processing techniques for 3-D graphics-related applications. First, we propose a novel mesh decomposition scheme called "visual salience-guided mesh decomposition," which uses visually salient features to guide the mesh decomposition process. Since the features adopted are closely related to the psychology-based theory of visual salience, the decomposition process can appropriately mimic the function of a human visual system. In addition, the proposed decomposition algorithm is robust againt randomization of vertex coordinates. There are a variety of applications that benefit from breaking up a 3-D object into components. These applications include collision detection, radiosity simulations, robust transmission and streaming, texture mapping, metamorphosis, 3-D shape retrieval, simplification and compression, watermarking, and control skeleton extraction for key-frame animation. In our investigation, the existing 3-D shape analysis and retrieval algorithms are lack of explicit link to psychology-based principles, while cognitive psychologists have found a set of principles (or properties) that are fairly important in perception of a form or a shape. This motivates us to conduct our second work called "a cognitive psychology-based approach for 3-D shape retrieval." In this work, we incorporate a set of principles, which is originated from cognitive psychology, into the design of 3-D shape retrieval system. The proposed approach is intended for mimicking human visual perception and recognition based on the psychology-based rules proposed by Hoffman and Singh. In addition, our system realizes a "recognition-by-components" and "recognition-by-visually-salient-components" search strategies. Moreover, use of combined search strategies can perform a coarse-to-fine retrieval task. In our third work, to provide a user with an ability to verify the integrity of the content he/she received (or retrieved), we propose a new fragile watermarking scheme for authenticating 3-D polygonal meshes. The proposed scheme can not only achieve localization of malicious modifications in visual inspection, but also is immune to certain incidental data processings, such as quantization of vertex coordinates and vertex reordering.

1.2 Basic Concepts about Mesh-based Representation for 3-D Objects

In order to speed up rendering and manipulation on 3-D models, a 3-D model is commonly represented by means of vertices, edges, and faces, which is so-called "polygonal meshes." In particular, triangulated meshes, comprised of triangle faces, have been widely used in representation of 3-D graphical objects. A triangulated mesh, in which every edge is shared by exactly two triangles and every triangle possesses exactly three neighboring triangles, is termed as *manifold* mesh. When every boundary edge on a triangulated mesh is adjacent to only one triangle, the triangulated mesh is so-called *manifold with boundary*. Since manifold meshes possess well-behaved topology, they are favorable for a number of applications, such as curvature analysis and radiosity. On the contrary, non-manifold meshes possess problematic topology such as cracks and T-junctions, which are not suitable particularly for CAD/CAM applications. Examples of manifold and non-manifold meshes are illustrated in Figs. 1.1-1.2, respectively. In addition to manifold/non-manifold types, triangulated meshes can be classified into *regular*, *semi-irregular*, and *irregular* meshes according to regularity of mesh connectivity. A mesh with regular connectivity means that every vertex has exactly six neighboring edges (valence of 6). On the mesh boundary, regular vertices have valence of 4. In an irregular mesh, vertices can have any degree. A semi-regular mesh is formed by starting with a coarse irregular mesh and then quadrisecting all triangles. Fig. 1.3 illustrates the regular, semi-regular, and irregular connectivity in 2-D.

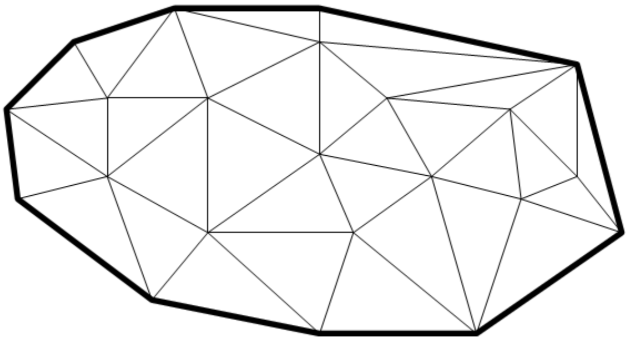
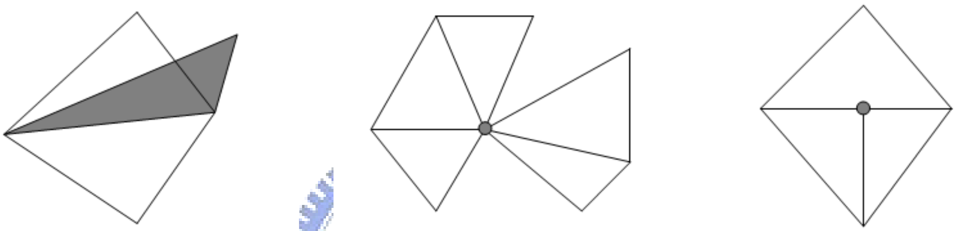
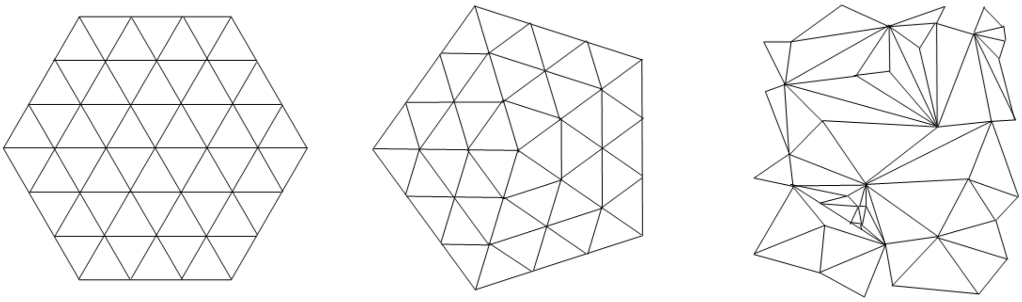


Figure 1.1: A 2-manifold with a boundary (boundary edges in bold) [82].



(a) Singular edge (b) Singular vertex (c) T-junction

Figure 1.2: Example of non-manifold meshes [82]: (a) a singular edge is an edge shared by more than three triangles; (b) a vertex shared by two otherwise unconnected sets of triangles is called a singular vertex; (c) a T-vertex.



(a) (b) (c)

Figure 1.3: Two-dimensional illustration of (a) regular, (b) semi-regular, and (c) irregular connectivity [25].

1.3 Overview of the Proposed Methods

In this dissertation, we propose three mesh processing techniques for 3-D graphics-related applications. To properly mimic human visual perception of 3-D shapes, we propose a visual salience-guided mesh decomposition scheme to decompose a 3-D mesh into parts. Using the proposed decomposition scheme as a preprocessing step, we propose a cognitive psychology-based approach for 3-D shape analysis and retrieval. For authentication of 3-D polygonal meshes, we propose a novel fragile watermarking method for localization of malicious modifications. An overview of the proposed methods is given as follows:

Visual Salience-Guided Mesh Decomposition

In this work, we propose a novel mesh decomposition scheme called "visual salience-guided mesh decomposition." The concept of "part salience," which originated in cognitive psychology, asserts that the salience of a part can be determined by (at least) three factors: the protrusion, the boundary strength, and the relative size of the part. We try to convert these "conceptual" rules into "real" computational processes, and use them to "guide" a 3-D mesh decomposition process in such a way that the *significant components* can be precisely identified and efficiently extracted from a given 3-D mesh. The proposed decomposition scheme not only identifies the part's boundaries defined by the minima rule, but also labels each part with a "quantitative" degree of visual salience during the mesh decomposition process.

A Cognitive Psychology-based Approach for 3-D Shape Retrieval

In this work, we incorporate a set of principles that originated in cognitive psychology into the design of 3-D shape analysis and retrieval systems. Based on the "visual salience-guided mesh decomposition" scheme we proposed previously, a 3-D shape represented in mesh form is first broken into parts such that human visual perception of the parts can be appropriately mimicked. Next, the decomposed parts are individually analyzed and quantified according to the properties of visual salience. To establish the indices of 3-D meshes for the subsequent retrieval process, spherical parameterization is adopted to map the decomposed parts onto the surface of a unit sphere. In this way, the degree of similarity between a query provided by a user and models in the database can be calculated.

Fragile Watermarking for Authenticating 3-D Polygonal Meshes

Designing a powerful fragile watermarking technique for authenticating 3-D polygonal meshes is a very difficult task. Yeo and Yeung [131] were first to propose a fragile watermarking method to perform authentication of 3-D polygonal meshes. Although their method can authenticate the integrity of 3-D polygonal meshes, it cannot be used for localization of changes. In addition, it is unable to distinguish malicious attacks from incidental data processings. In this work, we trade off the causality problem in Yeo and Yeung's method for a new fragile watermarking scheme. The proposed scheme can not only achieve localization of malicious modifications in visual inspection, but also is immune to certain incidental data processings (such as quantization of vertex coordinates and vertex reordering). During the process of watermark embedding, a local mesh parameterization approach is employed to perturb the coordinates of invalid vertices while cautiously maintaining the visual appearance of the original model. Since the proposed embedding method is independent of the order of vertices, the hidden watermark is immune to some attacks, such as vertex reordering. In addition, the proposed method can be used to perform region-based tampering detection.

1.4 Dissertation Organization

The remainder of this dissertation is organized as follows. In Chapter 4, the proposed visual salience-guided mesh decomposition is described in detail. In Chapter 3, we propose the computational process for realizing the qualitative salient features, and describe in detail how to incorporate visual salience into the representation scheme for 3-D shape retrieval. In Chapter 2, the proposed fragile watermarking method for authenticating 3-D polygonal meshes is described in detail. Finally, in Chapter 5, we present our conclusions and future work.

Chapter 2 Visual Salience-Guided Mesh Decomposition

Decomposition is a leverage to obtain the componential representation of a whole object. After the decomposition step is executed, the decomposed components can be individually selected, grouped, and analyzed based on the properties of interest. In recent years, a variety of applications have benefited from decomposing a 3-D object into its component parts. These applications include collision detection [74], radiosity simulations [40], robust transmission and streaming [14, 15], texture mapping [72, 105], metamorphosis [112], simplification and compression [138], 3-D shape retrieval [37, 138], and control-skeleton extraction for key-frame animation [62]. The requirements that an effective object decomposition method has to satisfy usually depend on the application. In this study, we emphasize high-level abstraction and organization for human object understanding.

High-level organization imposed on perceived data has been explored extensively in both human visual processes and computer vision systems. Related studies can be found in [11, 50, 51, 106, 114]. In order to account for human visual processes, cognitive psychologists have identified a set of properties (or rules) that are fairly important in the perception of a form or a shape. In the field of computer vision, perceptual organization has shown that computational resources can be effectively applied to extract structural and meaningful organization from perceived data. Furthermore, perceptual organization can be regarded as an intelligent process that can perform high-level abstraction for image understanding. In the literature, perceptual organization has been applied to the segmentation of range images [28], 2-D images [110], textures [102], patterns [134], and contours [136] respectively. However, for some reason, existing 3-D object decomposition techniques lack a direct link to perceptual organization. Below, we briefly review some existing 3-D object decomposition techniques and specify their possible links to perceptual organization. Furthermore, we point out their common shortcomings in terms of perceptual organization.

In [127], Wu and Levine introduced a simulated electrical charge distribution scheme to perform surface characterization. Using their representation, the concave boundaries can be located at the minima of the local charge density. In [84], Mangan and Whitaker extended the watershed algorithm, which was originally designed for image segmentation [108], to partition both volumetric and mesh surfaces. In [74], Li *et al.* employed skeletonization and space sweeping procedures to extract organic parts. In [115], Svensson and Baja introduced the concept of distance transform to decompose 3-D volumetric objects into kernels and elongated parts. Shlafman *et al.* [112] proposed a k-means based clustering algorithm that separates distant faces, but clusters close faces together. In [62], Katz and Tal proposed a general framework for mesh decomposition, in which a maximum-flow (minimum-cut) algorithm is applied to construct boundaries without jagged effects. In [75], Lien and Amato identified and resolved the non-convex features in order of importance to achieve approximate convex decomposition. In [80], Liu and Zhang introduced a spectral clustering method that favors segmentation along concave regions. In [63], Katz *et al.* introduced multi-dimensional scaling representation and spherical mirroring operation to extract prominent feature points and core component respectively. The boundaries between the extracted features are then constructed and refined using Katz and Tal's algorithm [62].

From the perspective of perceptual organization, the underlying assumption of the above methods is based on psychologists' definitions of a part, which are regularized by "a uniformity of nature" [50]. In cognitive psychology, the principle of transversality[1] is regarded as one of the guidelines for finding a part's shapes. Among existing methods, those that trace concave regions [62, 84, 93, 127] adopt the minima rule[2] to construct a part's boundaries. With regard to clustering-

[1] Transversality regularity: When two arbitrarily shaped surfaces are made to interpenetrate, they always meet in a contour of concave discontinuity of their tangent planes. For a detailed discussion of transversality, please refer to [45].

[2] Minima rule: All negative minima of the principal curvatures (along their associated lines of curvature) form boundaries between parts. For a detailed discussion of the minima rule, please refer to [51].

based methods [62, 112] and merging processes [84, 115], the Gestalt law of similarity and proximity is frequently adopted to capture the homogeneous characteristics of parts. Obviously, some existing 3-D object decomposition methods do attempt to extract a part's shapes and boundaries by mimicking human visual perception of 3-D shapes. However, none of the existing methods takes the salience of parts into account. According to the findings of cognitive psychologists [51], the salience of parts usually plays an important role in the 2-D silhouette and 3-D shape partitioning processes.

In this chapter, we propose a novel mesh decomposition scheme called "visual salience-guided mesh decomposition," which bases decomposition on the theory of part salience borrowed from cognitive psychology [51]. The theory asserts that the salience of a part is usually determined by three factors: the protrusion, the boundary strength, and the relative size of the part. The computational processes designed for realizing two of these salient features are presented in Section 2.2. We use these features to guide the decomposition process so that the visually significant components can be extracted from a given 3-D mesh. This new approach has a number of potential applications. For example, in 3-D shape databases, the organization of each object should be in accordance with our visual perception. Specifically, the organization from the parts to the whole would allow us to conduct a "part-in-whole" search process (as in [37]). In addition, extracting significant components based on different salient features would allow us to construct an efficient and valid set of visual parts from a 3-D model. In this way, one could realize "*query-by-significant-components*" in a 3-D shape retrieval system.

The remainder of this chapter is organized as follows. In Section 2.1, we introduce the theory of part salience and illustrate its importance in the perception of parts. In Section 2.2, we propose the computational processes for realizing the qualitative salient features, and describe in detail how to incorporate visual salience into the mesh decomposition process. The experiment results are presented in Section 2.3. Finally, in Section 2.4, we present our conclusions.

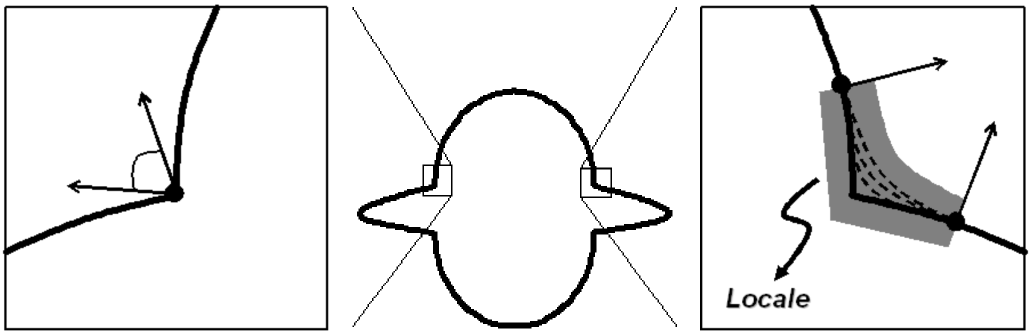
2.1 Review of Hoffman and Singh's Theory of Part Salience

In [51], Hoffman and Singh proposed the theory of part salience, which states that at least three factors determine the salience of a part: the protrusion, the boundary strength, and the relative size of the part. We now give the quantitative definitions of these salient factors and then describe their importance in visual processes.

Protrusion of A Part This factor is the degree to which a part protrudes from its main body. For 2-D silhouettes, it can be quantified as the ratio of the perimeter of the part (excluding its bases) to the sum of its base lengths. For 3-D shapes, the base of a part is referred to as the minimal surface formed by the boundary curve of the part. Hence, the protrusion of a 3-D part can be quantified as the ratio of the area of the part's surface to the area of its base surface.

Strength of A Part's Boundary According to the principle of transversality, a part's boundaries are usually located at the concave creases, as shown in Fig. 2.1(b). In [51], Hoffman and Singh proposed that possible quantitative definitions of the boundary strength include the turning normals and locale turning, as shown in Figs. 2.1(a) and 2.1(c) respectively. Obviously, the indication of the normal direction must have a global orientation consistency so that the boundary strength can be captured precisely. The discriminating capability of turning normals and locale turning is shown by the following examples. For 2-D silhouettes, two sides of a crease boundary usually have two normals, and the angle between them can, in one sense, represent the strength of that boundary. On the other hand, for potentially smooth boundaries, which are represented by the dotted lines in Fig. 2.1(c), there is one normal at every point along a curve. To tackle this problem, Hoffman and Singh [51] proposed obtaining the measure of turning in an appropriate region near the boundary. As shown in Fig. 2.1(c), the gray region is the so-called *locale*[3] and the normals on its two sides (i.e., the so-called *locale turning*) are used to characterize the strength of the smooth boundary. For 3-D shapes, the principal curvatures can be used to measure the strength of a

[3] By definition [51], a locale is an appropriate region near (but not just infinitesimally near) a negative minimum of the curvature, in which we can explore how the curve evolves.



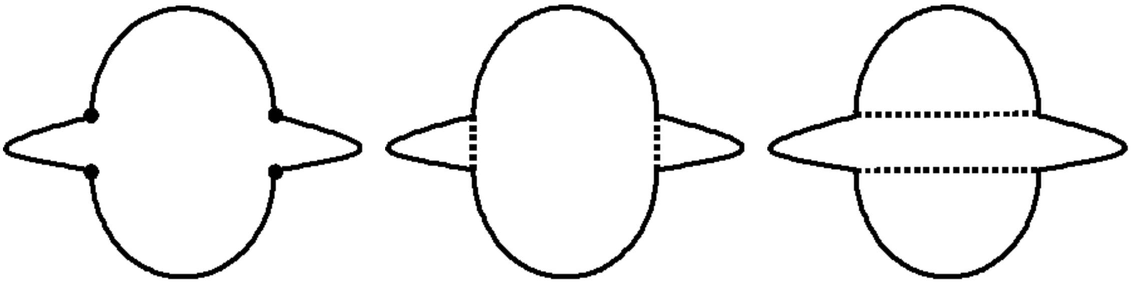
(a) Turning normals (b) 2-D Silhouette (c) Locale turning

Figure 2.1: Illustration of turning normals and locale turning at the boundary of a 2-D silhouette.

part's boundary.

Relative Size of A Part This factor indicates the size of a part relative to the whole object. For 2-D silhouettes, it can be defined as the ratio of the area of a part to the area of the whole object. For 3-D shapes, the relative volume can be used to measure a part's relative size.

Having reviewed the factors that may be used to determine the salience of a part, we now discuss their effects on both visual and decomposition processes. For simplicity, the following discussion is based on 2-D silhouettes; however, the concept can be easily extended to 3-D models. Fig. 2.2 shows the boundaries and cuts of parts of a 2-D silhouette, indicated by isolated points and dotted lines respectively. Note that, in Fig. 2.2(a), the four boundaries are used to form possible cuts; and, in Figs. 2.2(b) and 2.2(c), each part is generated by exactly one cut. According to the visually salient properties of interest, a 2-D silhouette may have different interpretations. For example, the 2-D silhouette might be interpreted as an alien's head with a pair of protrusive ears when the salience of the part is determined primarily by its protrusion (i.e., the part's cuts in Fig. 2.2(b)). On the other hand, the 2-D silhouette might be interpreted as an unidentified flying object when the part salience is determined primarily by its relative size (i.e., the part's cuts in Fig. 2.2(c)). As a result, the part salience would affect not only the high-level visual processes that determine the interpretation of a shape, but also the low-level visual processes that determine how the shape is really decomposed. In order to precisely determine a part's cuts, another independent theory that incorporates a priori knowledge about the shape is usually required. In the early 1980's, 3-D object recognition was a popular research topic [28]. Also, among the large number of research issues, 2-D perceptual organization [106] and recognition-by-components (or parts) [11, 50, 114] were two important directions. However, their development was hindered by some ill-posed early vision problems, such as edge detection and image segmentation. Since these problems could not be solved, 2-D perceptual organization and recognition by 2-D components (or parts) could not be converted into "complete" computational processes, so they both failed. Nowadays, there are large numbers of 3-D models distributed worldwide. Since 3-D models (or meshes) are not restricted by the limitations of 2-D images, perceptual organization is now possible in 3-D cases. Furthermore, Hoffman and Singh's theory of part salience means that a priori knowledge may not



(a) A part's boundaries (b) A part's cuts (c) A part's cuts

Figure 2.2: A part's boundaries and cuts in a 2-D silhouette (Re-sketched from [51]).

be necessary in 3-D shape decomposition processes. However, the quantitative definitions for part salience proposed by Hoffman and Singh [51] were made under the assumption that a part and its boundary are found in advance. In terms of perceptual organization, this is a drawback that, to some extent, limits the power of Hoffman and Singh's theory. In this chapter, we propose a new mesh decomposition scheme that incorporates the cognitive psychology theory into the mesh decomposition process such that the visually significant components can be extracted from a given 3-D mesh.

2.2 Visual Salience-Guided Mesh Decomposition

We now discuss the computational processes for realizing two of the visually salient features, namely, the protrusion and the boundary strength. We also describe how to incorporate each visually salient feature into the mesh decomposition process. As to the third salient feature, the relative size of components, we can easily calculate it once the protrusion and the boundary strength are known. We shall use the relative size feature in the mesh retrieval process. This section is organized as follows. In Section 2.2.1, we present the computational process for characterizing the protrusion of an arbitrary surface mesh. Based on protrusion characterization, a local maximum approach for choosing the salient representatives of parts is proposed and described in Section 2.2.2. In Section 2.2.3, we describe in detail the proposed computational process for modeling the boundary strength. The proposed measure of boundary strength is used as the guideline to find the locale of a part's boundary. In Section 2.2.4, a coarse-to-fine approach is proposed for finding the locale of a part's boundary. In Section 2.2.5, Katz and Tal's algorithm for determining the boundary of a part (presented in [62]) is described for the purpose of completeness.

2.2.1 Modeling the Protrusion as the Degree of Center

In this section, we propose a suitable way to characterize the protrusion of a shape. It is intuitive that a protrusion is closely related to the skeletal structure of a shape. As a result, some existing skeletonization methods [22, 70, 74, 111] may be useful for characterizing the protrusion. In our investigations, however, we found that the integral function proposed by Hilaga *et al.* [49] is more suitable for protrusion characterization. The main reasons are as follows. First, the integral function can be constructed on any type of polygonal meshes, including

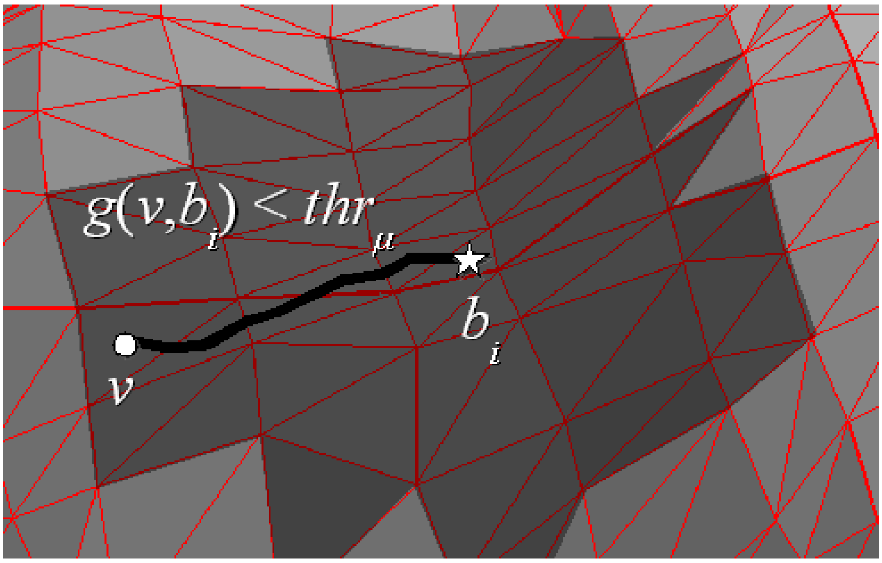


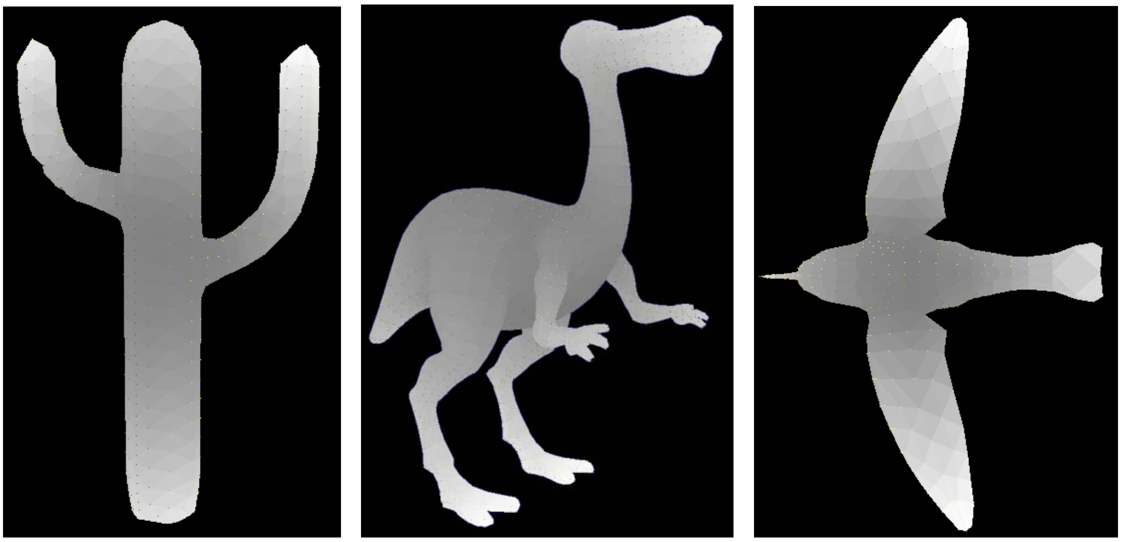
Figure 2.3: Illustration of the base patch construction for protrusion characterization: The darker region is the base patch occupied by b_i.

non-orientable, non-closed, and non-manifold surfaces. Second, the function is very stable so that there is no initial point selection problem. Third, the integral can be calculated over the entire surface. As a result, the protrusion of every vertex is accessible to any salience-guided process. Finally, the function is not only invariant to geometrical transformations (such as rotation, translation, and scaling), but is also resistant to noise added to vertex coordinates. Therefore, we adopt the integral function described in [49] to characterize the protrusion of a part.

In [49], the degree of center at the point v on the surface S is defined as follows [49]:

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

where $g(v,p)$ represents the geodesic distance between v and p on S. The contin-



(a) Cactus (b) Dinopet (c) Hummingbird

Figure 2.4: The protrusion degree calculated on different 3-D meshes.

uous integral function $\mu(v)$ is defined as the total sum of geodesic distance from the point v to all points on S. In other words, the value of $\mu(v)$ can be interpreted as a distance from the point v to arbitrary points on S. More precisely, a smaller value of $\mu(v)$ indicates that the point v is closer to the center of the surface S. On the other hand, a larger value of $\mu(v)$ means that the point v is farther from the center of the surface. It can be seen from Eq. (2.1) that calculating the integral based on geodesic distance is computationally prohibitive. To trade off accuracy for computational efficiency, Hilaga *et al.* employed Dijkstra's algorithm to approximate geodesic distance based on edge length of a 3-D mesh.

Here, in contrast to [49], the integral function is constructed on the dual graph of a given 3-D mesh, $G = (V, E)$, where V and E represent the set of dual vertices and the set of dual edges respectively. A dual vertex $v \in V$ is referred to as the center of mass of a face in the original mesh, while a dual edge $(u, v) \in E$ links the center-of-mass of two adjacent faces and intersects at the midpoint of the edge shared by the two faces. For computational efficiency, we segment the mesh into small patches of approximately equal size, which we called base patches. Each base patch is represented by a single dual vertex, b_i, located at its approximate center. Such a base patch is constructed by a modified version of Dijkstra's algorithm such that the shortest distance between the base vertex and any vertex within the base patch is less than a radius value. As shown in Fig. 2.3, the darker region is the base patch of the radius thr_μ with the base dual-vertex b_i in its center. Obviously, by increasing the number of base patches, a more accurate integral can be obtained; however, the drawback is an increase of computation time. Let $area(v)$ denote the area of the mesh face corresponding to a dual vertex v and $area(V)$ denote the total area of the object surface. The protrusion degree at a dual vertex v can be defined as in [49]:

$$\mu(v) = \sum_i g(v, b_i) \cdot area(P_i), \tag{2.2}$$

where $\{b_0, b_1, \ldots\}$ are the base dual-vertices representing the base patches $\{P_0, P_1, \ldots\}$. In addition, $area(P_i)$ is the area of the entire base patch $area(P_i) = \sum_{v_j \in P_i} area(v_j)$ and $\sum_i area(P_i) = area(V)$, while $g(v, b_i)$ returns the geodesic distance between the dual vertex v and the base vertex b_i. Since the function $\mu(v)$ defined in Eq. (2.2) is not invariant to scaling transformation, a normalized version of $\mu(v)$ is defined as in [49]:

$$\text{Protrusion}(v) = \frac{\mu(v) - \min_{u \in V} \mu(u)}{\max_{u \in V} \mu(u)}. \tag{2.3}$$

The calculation of the integral function has the complexity $O(|V|\log|V|)$, where $|V|$ is the number of faces on the mesh. Using the normalized protrusion degree defined in Eq. (2.3), we can calculate a numeric value (ranging from 0 to 1) for each dual vertex located on a 3-D mesh. The farther a dual vertex is from the center of a 3-D mesh, the larger the protrusion degree will be. Fig. 2.4 illustrates the protrusion degree calculated on different 3-D meshes. Note that a darker color represents a protrusion degree close to 0, while a lighter color means the protrusion degree is close to 1.

2.2.2 Choosing the Salient Representatives of Parts

Here, we describe how to select a set of salient representatives from a given 3-D mesh. The local maxima of protrusion degrees is the criterion used to select salient features. After the selection process is completed, each identified local maximum can be regarded as a salient representative of a part. Given a dual vertex $r \in V$, the dual vertex is chosen as a salient representative of a part if the following condition is satisfied:

$$\text{Protrusion}(r) = \max_{W_r} \{\text{Protrusion}(v)\} \tag{2.4}$$

where $W_r = \{v \in V | g(r, v) < thr_p\}$ is an observation window for finding a local maximum of protrusion degrees; and thr_p represents the size of the observation

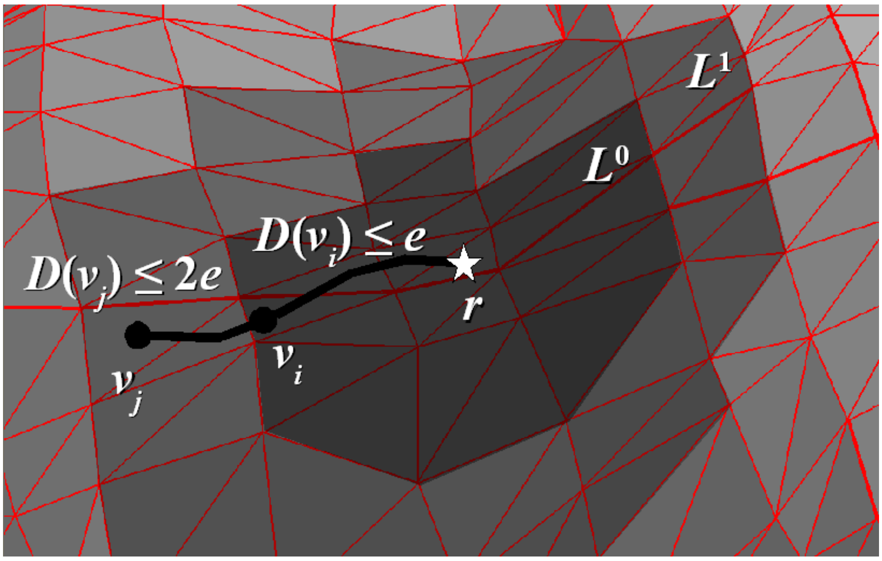


Figure 2.5: Illustration of the candidate locales construction: The two darker regions are the first two candidate locales of the salient representative of a part.

window, with which we can control the range of influence of a protrusive stimulus. The observation window can be constructed using the modified version of Dijkstra's algorithm mentioned in the previous section. By replacing b_i and thr_μ in Fig. 2.3 with r and thr_p respectively, the darker region shown in Fig. 2.3 can be interpreted as the observation window for choosing the salient representative. Moreover, if the protrusion degree of the vertex r (i.e., the star shown in Fig. 2.3) is the largest value within the local window, the vertex is chosen as the salient representative. Note that since the observation windows of local maxima are subject to overlap, only one of them is chosen as a salient representative.

2.2.3 Modeling the Boundary Strength based on the Border Area Change

In this section, we describe how to convert the concept of boundary strength into a computational process. Since the boundary of a part is completely unknown, we start from the surface mesh and the salient representatives obtained in the previous section. Motivated by the concept of locale turning (described in Section 2.1), we use Dijkstra's algorithm [23] to explore how the surface evolves in the locale of a boundary. For clarity, we split the computational process for modeling the boundary strength into two steps:

Step 1. Establishing the Candidate Locales

Given a salient representative of a part, r, a set of candidate locales, $\{L_r^x\} = \{L_r^0, L_r^1, \cdots\}$, is established. For simplicity and later use, we drop the subscript r in subsequent descriptions and denote the xth candidate locale as:

$$L^x = \{v | \forall v \in V, x \cdot e \leq D(v) < (x+1) \cdot e\} \qquad \text{for } x \in \{0, \ldots, l-1\}, \quad (2.5)$$

where e represents the extent of a candidate locale, in which the boundary evolution is explored; and $l = \lfloor \max_{v \in V} D(v)/e \rfloor$ is the number of candidate locales established. $D(v)$ returns the shortest distance from the source, r, to a dual vertex, v, in terms of geodesic distance and protrusive difference. Fig. 2.5 illustrates that based on the new distance measure $D(\cdot)$, the first two candidate locales, L^0 and L^1, are established using the modified version of Dijkstra's algorithm (as in Section 2.2.1). To compute the shortest distance $D(\cdot)$, the weight for each edge $(u, v) \in E$ in the dual graph is defined as follows:

$$Weight(u, v) = \delta \cdot \frac{Len(u, v)}{avg(Len)} + (1 - \delta) \cdot \frac{Prot(u, v)}{avg(Prot)}, \tag{2.6}$$

where $Len(u, v)$ is the length of a dual edge between u and v. Here, $Prot(u, v)$ represents the absolute protrusion degree of difference between two dual vertices, u and v. Also, $avg(Len)$ and $avg(Prot)$ represent the average length and the average protrusion degree difference respectively. In order to fulfill the proximity and similarity requirement of the Gestalt laws, the first term on the right-hand side of Eq. (2.6) is usually considered in Dijkstra's algorithm to determine the shortest distance (or path) on a graph in terms of the geodesic proximity. The purpose of the second term is to balance the effect caused by the geodesic proximity, while creating the candiate locales containing similar protrusion degrees; δ is the weighting between the two constraints. Moreover, including the protrusive similarity is helpful in maintaining a locale's boundaries approximately parallel to a part's boundaries.

Since each salient representative produces a set of candidate locales and eventually grows into the whole 3-D mesh, certain of candidate locales must "march" by the potential region of its corresponding part boundary. However, the sets of candidate locales will overlap one another. To prevent candidate locales from marching into the regions occupied by other parts, a constrained set of candidate locales is constructed such that the region-growing process always ends whenever a termination base is touched. To do so, we first define the termination base, K, as follows:

$$K = \{v | \forall v \in V, \text{Protrusion}(v) \leq thr_b\}, \tag{2.7}$$

where thr_b is the parameter used to collect the set of faces that forms the termination base. Next, the constrained set of candidate locales, $\mathbf{L}$, is defined as the union of $(m+1)$ *consecutive* locales,

$$\mathbf{L} = \bigcup_{x=0}^{m<l-1} L^x, \tag{2.8}$$

satisfying the following constraint:

$$L^x \cap K \neq \emptyset \qquad \text{for } x \in \{m - \Delta_b, \ldots, m\}, \tag{2.9}$$

where Δ_b is used to specify that the last $(\Delta_b$+1) locales in $\mathbf{L}$ overlap with the termination base, K. By the above construction, the overlap between a constrained set of locales and the termination base provides a potential region in which to find the correct boundary of a part.

Step 2. Modeling the Boundary Strength

With the constrained set of candidate locales established in Step 1, we now consider two adjacent locales in $\mathbf{L}$ to determine how the surface evolves in candidate locales. Let V_{L^x} denote the set of dual vertices in $L^{x+1} \in \mathbf{L}$ that has a dual edge joining $L^x \in \mathbf{L}$ in the graph G. We then associate the following geometric property to the xth candidate locale in $\mathbf{L}$:

$$f(x) = \sum_{v \in V_{L^x}} area(v). \tag{2.10}$$

Since V_{L^x} is a set of dual vertices that collects the direct neighbors between L^x and L^{x+1}, $f(x)$ can be regarded as the total-area-of-border between two adjacent candidate locales. Based on the geometric property defined in Eq. (2.10), we model the boundary strength as the total-area-of-border change in response to the boundary's evolution. The modeling is reasonable because, at the border of two adjacent parts, the total-area-of-border defined above will usually undergo a significant change. Therefore, to judge whether a locale contains a boundary using the total-area-of-border change is a justifiable choice. As a consequence, the boundary strength at the xth candidate locale can be defined as follows:

$$\text{Boundary\_Strength}(x) = \left| f(x+1) - f(x) \right|. \tag{2.11}$$

By obtaining the measure of boundary evolution for the boundary strength, we can explore how the surface evolves in the locale of a part's boundary. Moreover, by treating $f(x)$ as a one-dimensional function defined in $\mathbf{L}$, we can make the process for finding the locale of a part's boundary analytic.

2.2.4 Finding the Locale of A Part's Boundary

In this section, we describe how to use the previously defined boundary strength to locate the locale of a part's boundary. As mentioned in the previous section, the boundary strength is quantified in response to the boundary's evolutionary process. Hence, the locale of a part's boundary should possess the maximum boundary strength. However, the function $f(x)$ is very jagged (or noisy), since the faces in the immediate neighbor of the xth candidate locale can never have a regular area due to the nature of a mesh-based object. This makes finding the locale of a part's boundary very difficult. To overcome this, based on Haar wavelet representation [83, 96], the function $f(x)$ is transformed into w different scales, $f_1(x), f_2(x), \cdots, f_w(x)$. Then, the candidate locale that possesses the most significant boundary strength is traced from a coarser scale $f_j(x)$ to a finer scale $f_{j-1}(x)$ until a predefined finer scale is reached. In this way, one can conduct a coarse-to-fine search to identify the locale that contains a part's boundary. Let k_{j-1} denote the index of the candidate locale that possesses the maximum boundary strength in $f_{j-1}(x)$. Then, the index k_{j-1} is determined by the following recursion:

$$\begin{cases} k_w = \arg\max_x \mathrm{Boundary\_Strength}_w(x) & \text{initially} \\ k_{j-1} = \arg\max_{x \in I_j} \mathrm{Boundary\_Strength}_{j-1}(x) & \text{if } 1 \leq j < w \end{cases}, \tag{2.12}$$

where $I_j = \{2k_j, \ldots, 2k_j + 2\}$ is the search range derived from the index k_j (i.e., the k_jth candidate locale in $f_j(x)$). $\mathrm{Boundary\_Strength}_{j-1}(x)$ is equal to

$|c_{j-1,x+1} - c_{j-1,x}|$, which can be regarded as the boundary strength characterized in the finer scale $f_{j-1}(x)$. Note that the above recursion is stopped at the finer scale, $f_1(x)$, since the original function, $f(x)$, is too noisy to correctly find the locale of the part's boundary. Furthermore, since the extent of a candidate locale may be too small to contain a part's boundary, we can extend it as follows:

$$\widetilde{\mathbf{L}} = \bigcup_{x=2k_1-\Delta_+}^{2k_1+\Delta_-} L^x, \qquad L^x \in \mathbf{L}, \tag{2.13}$$

where Δ_- and Δ_+ are used to respectively control the left and right extent of the candidate locale found above. After the region containing a part's boundary is found, the next step is to construct the boundary. In [62], Katz and Tal have shown that by defining an appropriate capacity function, the boundary can be found by solving a maximum-flow (minimum-cut) problem [23,41]. We, therefore, apply Katz and Tal's method [62] to construct the boundary.

2.2.5 Determining the Boundary of A part

In this section, for the purpose of completeness, we describe Katz and Tal's method [62] for determining a part's boundary. As described in [62], the problem of how to construct a part's boundary within the region containing the boundary is formulated as a maximum-flow (minimum-cut) in an undirected constrained flow network graph problem. To construct the flow network graph, we denote the locale, $\widetilde{\mathbf{L}}$, found in the previous section by the dual graph $G_{\widetilde{\mathbf{L}}} = (V_{\widetilde{\mathbf{L}}}, E_{\widetilde{\mathbf{L}}})$ and the remaining two regions A and B, which are separated by $\widetilde{\mathbf{L}}$, by $G_A = (V_A, E_A)$

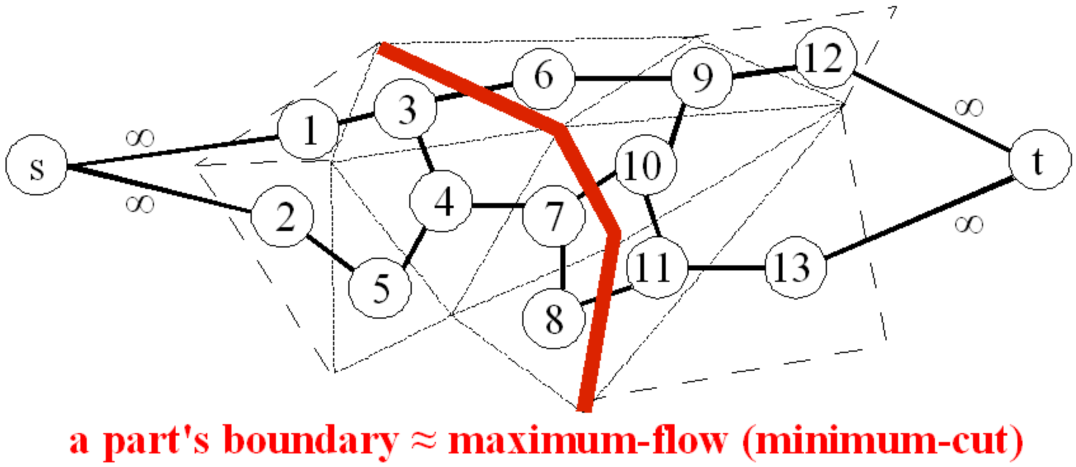


Figure 2.6: Illustration of constructing the boundary by solving the maximum-flow (minimum-cut) problem.

and $G_B = (V_B, E_B)$ respectively (i.e., $G_A \cup G_{\widetilde{\mathbf{L}}} \cup G_B = G$). In addition, the set of all dual-vertices in V_A whose corresponding faces in A share an edge with faces in $\widetilde{\mathbf{L}}$ is denoted by $V_{\widetilde{\mathbf{L}}A}$ (resp. $V_{\widetilde{\mathbf{L}}B}$). Next, we construct an undirected flow network graph $G' = (V', E')$ by adding two new vertices, s (source) and t (sink), as in [62]:

$$V' = V_{\widetilde{\mathbf{L}}} \cup V_{\widetilde{\mathbf{L}}A} \cup V_{\widetilde{\mathbf{L}}B} \cup \{s, t\}\,,$$

$$E' = E_{\widetilde{\mathbf{L}}} \cup \left\{(s,v), \forall v \in V_{\widetilde{\mathbf{L}}A}\right\} \cup \left\{(t,v), \forall v \in V_{\widetilde{\mathbf{L}}B}\right\}$$
$$\cup \left\{e_{uv} \in E | u \in V_{\widetilde{\mathbf{L}}}, v \in \left\{V_{\widetilde{\mathbf{L}}A} \cup V_{\widetilde{\mathbf{L}}B}\right\}\right\}. \quad (2.14)$$

As illustrated in Fig. 2.6, the dotted mesh is the region $\widetilde{\mathbf{L}}$ (i.e., the nodes 3-11) while the solid lines and circle nodes together form the constrained flow network graph. In addition, the two nodes 1 and 2 shown in Fig. 2.6 belong to the region A while the two nodes 12 and 13 belong to the region B. Note that in Eq. (2.14), the goal of adding the two sets of vertices, $V_{\widetilde{\mathbf{L}}A}$ (i.e., the nodes 1 and 2 in Fig. 2.6) and $V_{\widetilde{\mathbf{L}}B}$ (i.e., the nodes 12 and 13 in Fig. 2.6), is to consider the case that the boundary is on either side of the locale, that is either the boundary between $V_{\widetilde{\mathbf{L}}A}$ and $\widetilde{\mathbf{L}}$ or the boundary between $V_{\widetilde{\mathbf{L}}B}$ and $\widetilde{\mathbf{L}}$.

After the flow network graph is constructed, the capacity function Cap on an edge $(u, v) \in E'$ is defined as in [62]:

$$Cap(u,v) = \begin{cases} \frac{1}{1+\frac{Ang\_Dist(\alpha_{uv})}{avg(Ang\_Dist)}} & \text{if } \{u, v \neq s, t\} \\ \\ \infty & \text{else} \end{cases}, \qquad (2.15)$$

where α_{uv} represents the angle between the two faces sharing the same edge (u, v) and Ang_Dist is a conversion function $Ang\_Dist(\alpha_{uv}) = \eta(1 - \cos(\alpha_{uv}))$. The conversion function is used to map the dihedral angle α_{uv} to a positive value; $avg(Agn\_Dist)$ represents the average angle distance over the entire mesh. In addition, a small positive value for η is used for convex angles while $\eta = 1$ is used for concave angles. In [62], Katz and Tal have shown that by the definition of $Cap(u, v)$, the minimum cut found in the flow network graph tends to pass through edges having highly concave dihedral angles. As shown in Fig. 2.6, the thicker line represents the boundary corresponding to the maximum-flow (minimum-cut) in the flow network graph.

With the proposed measures for the protrusion and the boundary strength, the significant components of an arbitrary 3-D mesh can be identified and extracted according to their visual salience (i.e., visual significance). In terms of efficiency, the total complexity of the proposed method is $O(|V|^2\log|V|)$. The protrusion characterization can be performed in $O(|V|\log|V|)$ [49]. In addition, the process for choosing a part's salient representative costs $O(|V|^2\log|V|)$. The process for modeling the boundary strength can be performed in $O(R|V|\log|V|)$, where R denotes the number of salient representatives.

2.3 Experiment Results

A series of experiments were conducted to test the effectiveness of the proposed method. We used the set of triangulated meshes listed in Table 2.1 as the data set in our experiments. Moreover, all the parameters used in our experiments were the same and set as follows. To choose the salient representatives of parts, we found that most of the salient representatives could be properly chosen by assigning $thr_p = 5 \cdot thr_\mu$, where thr_μ was used to generate the base patches for protrusion characterization (as mentioned in Section 2.2.1). In the implementation process, we adopted the radius value $thr_\mu = \sqrt{0.005 \cdot area(V)}$ described in [49] to generate the set of base patches. Note that since the parameter, thr_p, is related to the total area of a mesh surface, the number of salient representatives depends on the mesh itself. As shown in Fig. 2.10, different meshes have different numbers of salient representatives. (i.e., the balls on the surfaces). To generate the constrained set of candidate locales, the threshold value $thr_b = 0.05$ was adopted to generate the termination base. To perform Dijkstra's algorithm,

Table 2.1: The triangulated meshes used in our experiments

Model name	**Number of vertices**	**Number of faces**	**Running time (sec.)**
Bunny	3752	7500	5.880
Cactus	620	1236	0.306
Cat	2779	5544	3.355
Cheetah	4704	9404	8.077
Cheetah2	5027	10050	9.036
Dinopet	2039	3999	2.303
Duck	716	1428	0.462
Female	1792	3572	1.573
Hand	1577	3110	1.472
Horse	2502	5000	3.284
Hummingbird	830	1640	0.611
Manatee	1977	3940	2.043
Santa	2502	5000	3.385
Stingray	997	1990	1.142
Tiger	956	1908	0.727

the parameter $\delta = 0.5$ was used to balance the weighting between the geodesic proximity and the protrusive similarity. To obtain the wavelet transform of $f(x)$, a fixed number of candidate locales $l = 32$ was adopted. As a result, we assigned the extent of a locale as $e = \lfloor \max_{v \in V} D(v)/32 \rfloor$ during the process of collecting candidate locales. In addition, for $x \in \{m+1, \cdots, l-1\}$ the value of $f(x)$ was padded with $f(m)$.

2.3.1 Results of Finding the Locale of A Part's Boundary

This experiment was comprised of two parts. The intent of the first part was to show how to construct the candidate locales, the termination base, and the con-

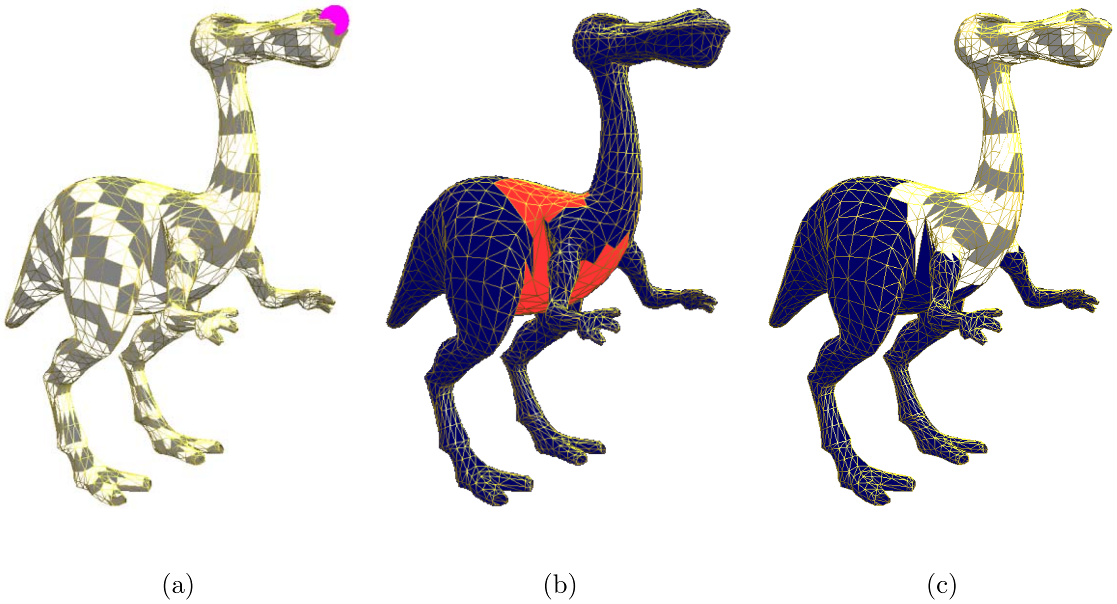
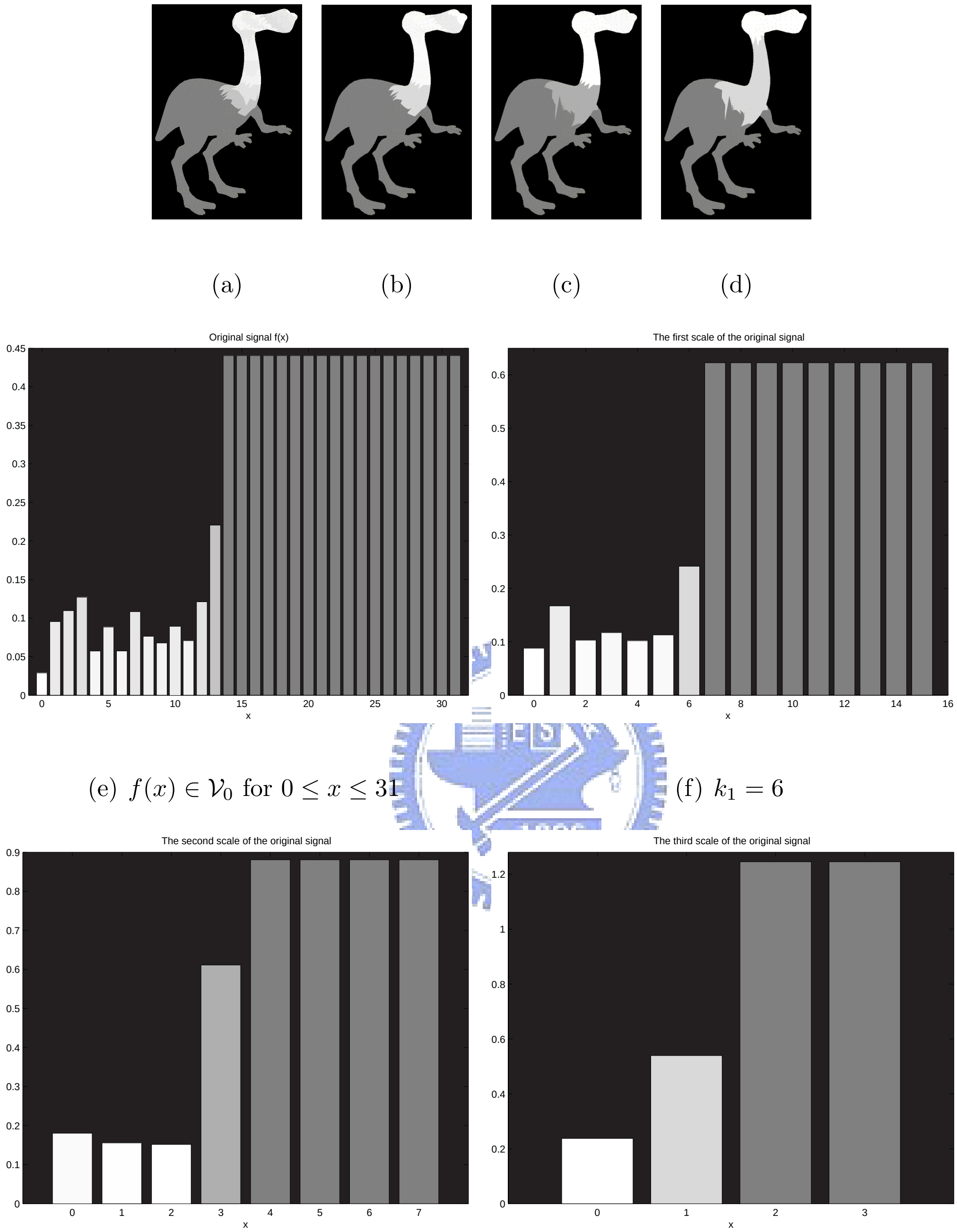


Figure 2.7: Illustration of (a) 32 candidate locales, (b) the termination base, and (c) the constrained candidate locales constructed from the dinopet model.

strained set of candidate locales. As shown in Fig. 2.7(a), thirty-two candidate locales were collected by applying Dijkstra's algorithm to the source (i.e., the ball near the dinopet's mouth). Meanwhile, Fig. 2.7(b) shows the termination base extracted from the dinopet model. Based on the results of Figs. 2.7(a) and 2.7(b), the constrained set of candidate locales was collected such that the consecutive candidate locales (starting with L^0) were gathered together and the last three candidate locales overlapped with the termination base. As shown in Fig. 2.7(c), these constrained candidate locales were effective in localizing the visual part. The intent of the second part of this experiment was to show the effectiveness of the proposed method in finding the locale of a boundary. As mentioned in Section 2.2.3, the change of total-area-of-border between two adjacent locales is utilized to judge how a part's boundary evolves in the candidate locales. For visualization purposes, the measure of total-area-of-border (i.e., $f(x)$) is converted into a gray scale; thus, the darker the gray scale, the larger the total-area-of-border. Fig. 2.8(a) shows the original function $f(x)$, shown in Fig. 2.8(e), plotted on the surface of the dinopet model. More precisely, the gray scale of the xth candidate locale on the surface mesh corresponds to the total-area-of-border $f(x)$. It can be seen from Fig. 2.8(e) that the shape of the original function is rough and uneven. Figs. 2.8(f)-2.8(h) show three different scales of the original function, $f_1(x), f_2(x)$, and $f_3(x)$ respectively. Their corresponding plots on the surface mesh are shown in Figs. 2.8(b)-2.8(d) respectively. To find the locale of a part's boundary, we started from the coarsest scale $f_3(x)$ and then found the most significant boundary strength within this scale (i.e., $k_3 = 1$). This shows that the locale of the boundary can be found within the 8th-16th candidate locales. Next, within the second scale, $f_2(x)$, the most significant boundary strength was located at $k_2 = 2$. The search range for the locale of the boundary was then shrunk (i.e., within the 8th-12th candidate locales). Finally, we stopped at the finer scale, $f_1(x)$, and $k_1 = 6$ was found within this scale. The locale of the boundary was then formed by the union of the 11th-13th candidate locales. As shown in Fig. 2.9, the proposed method described in Section 2.2.4 is effective in finding the locale of a part's boundary.



(a) (b) (c) (d)

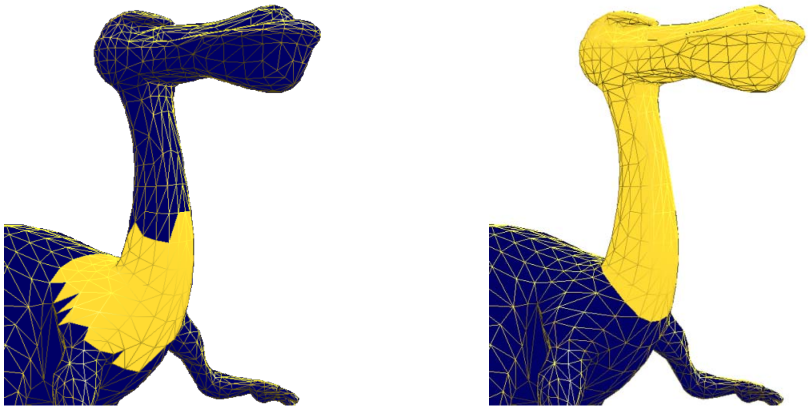
(e) $f(x) \in \mathcal{V}_0$ for $0 \leq x \leq 31$

(f) $k_1 = 6$

(g) $k_2 = 2$ and $I_3 = \{4, 5, 6\}$

(h) $k_3 = 1$ and $I_3 = \{2, 3, 4\}$

Figure 2.8: Each candidate locale is associated with its corresponding geometric property, which can be regarded as a one-dimensional function that defines an object surface: (a)–(d) the different scaled versions of the function $f(x)$ plotted on the object surface; (e)–(h) the different scaled versions of the function $f(x)$ corresponding to (a)–(d) respectively.



(a)

(b)

Figure 2.9: The locale of the part's boundary and its decomposition result: (a) the region that contains a part's boundary was found by applying the proposed method (i.e., $L^{11} \cup L^{12} \cup L^{13}$); (b) the nearly concave boundary was constructed within the region in (a) using the method proposed by Katz and Tal [62].

2.3.2 Visual Salience-Guided Mesh Decomposition for Extracting Significant Components

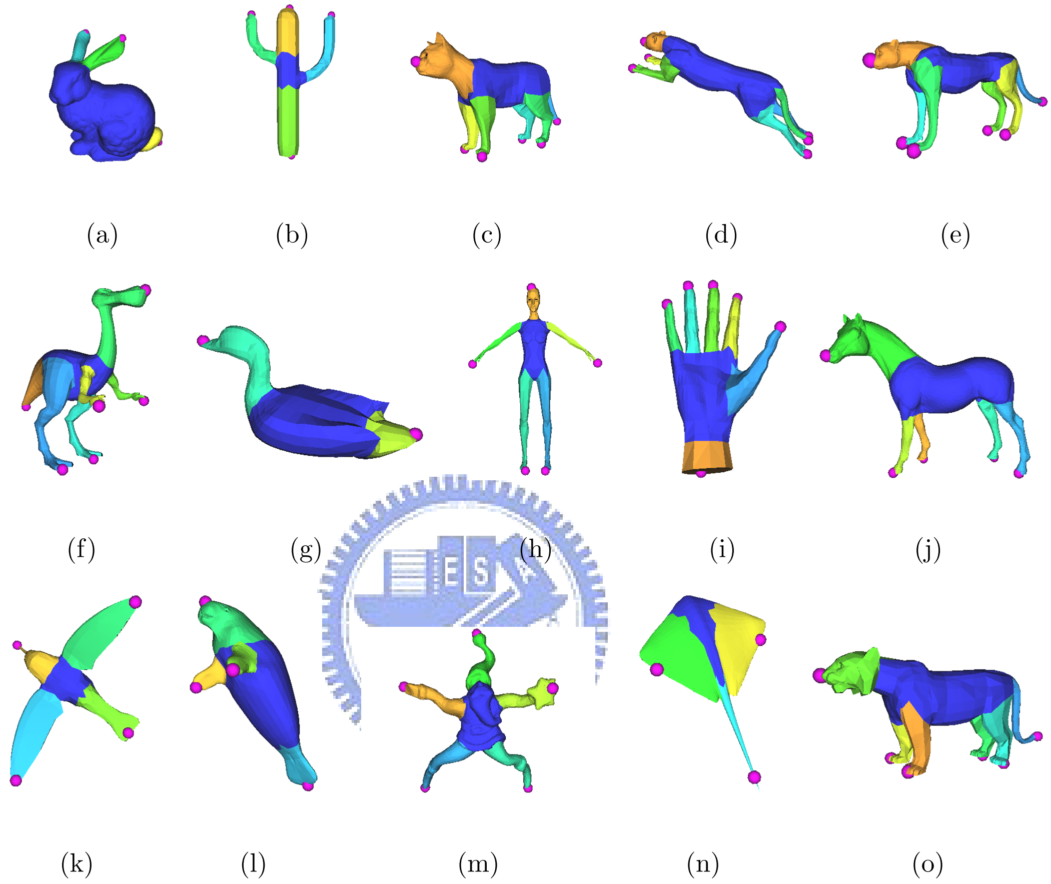
This experiment was comprised of three parts. The intent of the first part of this experiment was to show the effectiveness of the proposed method in decomposing a 3-D mesh into parts. In this experiment, each model listed in Table 2.1 was decomposed into a set of parts using the proposed decomposition method[4]. The last column of Table 2.1 lists the running time of decomposing the fifteen test

[4] Note that in the proposed mesh decomposition scheme, each part is individually decomposed from a 3-D mesh. As a result, it is possible that a part can be covered by other parts. To deal with this issue, the overlapped faces can be separated according to their geodesic distance to the salient representatives. On the other hand, the overlapped parts may be merged in the case where the amount of the overlapped area is larger than a predefined threshold. Currently, these two features are implemented by our system. However, a more intelligent process for merging parts should be developed to deal with the situation where the salient representatives are chosen incorrectly.

models into parts, on a Pentium IV, 1.98GHz, 1GB RAM PC. Each individual part of a model was then associated with its visually salient features so that the significant component could be identified. Fig. 2.10 shows the fifteen test models decomposed into visual parts after the visual salience-guided mesh decomposition method was applied. Since protrusion characterization was used to choose the salient representatives, it can be seen from Fig. 2.10 that most of the representatives were located at the tips of parts. On the other hand, as shown in Fig. 2.10, the boundaries between the parts and the main "body" were constructed precisely according to boundary strength.

Since human visual perception of parts is insensitive to noise and small undulations applied to the vertex coordinates of a 3-D object, the proposed method mimics the same visual processes. Thus, the intent of the second part of this experiment was to show the robustness of the proposed method under the randomization of vertex coordinates. The randomization was controlled by means of the noise strength, which is defined as the ratio of the largest displacement to the longest edge of the object's bounding box. Figs. 2.11(a)-2.11(c) show the effects of different levels of noise on the randomization applied to vertex coordinates of the object's surface. It is obvious that the proposed method still succeeds in decomposing the dinopet model into its component parts.

The intent of the third part of this experiment was to compare the proposed scheme against the method of Katz and Tal [62]. As mentioned in [62], there

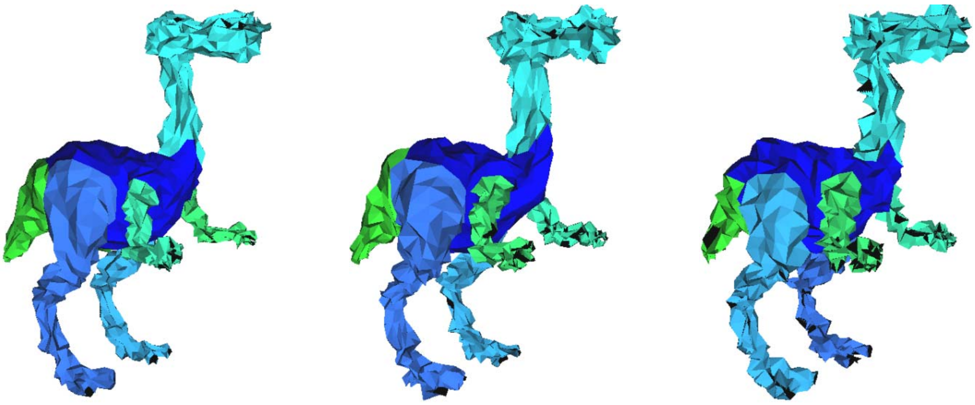


(a) (b) (c) (d) (e)

(f) (g) (h) (i) (j)

(k) (l) (m) (n) (o)

Figure 2.10: Visual salience-guided mesh decomposition results, where the salient representatives of parts chosen from different meshes are indicated by balls on the surfaces: (a) bunny - 3 representatives; (b) cactus - 4 representatives; (c) cat - 6 representatives; (d) cheetah - 6 representatives; (e) cheetah2 - 6 representatives; (f) dinopet - 6 representatives; (g) duck - 2 representatives; (h) female - 5 representatives; (i) hand - 6 representatives; (j) horse - 5 representatives; (k) hummingbird - 4 representatives; (l) manatee - 4 representatives; (m) santa - 5 representatives; (n) stingray - 3 representatives; (o) tiger - 6 representatives.



(a) ns=0.3 (b) ns=0.4 (c) ns=0.5

Figure 2.11: The robustness of the proposed visual salience-guided mesh decomposition method under the randomization of vertex coordinates, which is controlled by means of the noise strength ns (i.e., the ratio of the largest displacement to the longest edge of the object's bounding box).

are two versions of Katz and Tal's method: the (recursive) binary decomposition and the fuzzy k-means decomposition. Since the proposed method is type of k-way decomposition, we implemented Katz and Tal's fuzzy k-means decomposition method for the purpose of comparison. Fig. 2.12 shows two different results of decomposing the donkey model used in [62] into parts using the k-means based method and the proposed method respectively. As shown in Fig. 2.12(a), seven representatives were chosen in order to generate the corresponding patches. Fig. 2.12(b) shows that six salient representatives chosen using the method described in Section 2.2.2 were located at the tips of the donkey's four legs, the tip of the donkey's head, and the tip of the donkey's tail respectively. In the proposed decomposition scheme, contrary to the k-means based method [62], the termination base represents the main body of the model while each feature

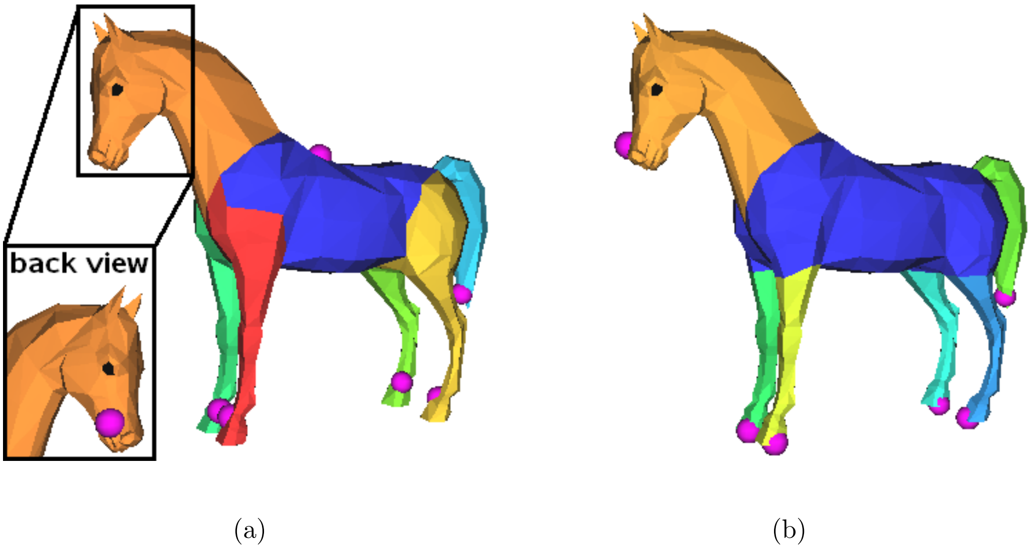
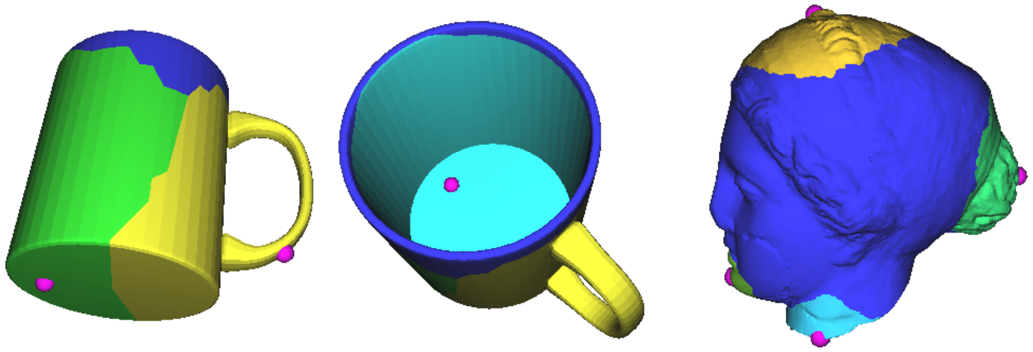


Figure 2.12: Results of decomposing the donkey model used in [62] into parts using (a) the fuzzy k-means clustering method (as in [62]) and (b) the proposed method respectively.

point represents the part that protrudes from the main body. As a result, the total number of patches generated by the proposed method is equal to the number of salient representatives plus one main body. Moreoever, as shown in Fig. 2.12(b), the proposed decomposition algorithm favors the boundaries between the protrusive parts and the main body. With regard to efficiency, the running time of decomposing the donkey model using the k-means based approach, which required 4 iterations to converge, was 2.707 seconds while that of using the proposed method was 0.452 seconds. Obviously, the proposed decomposition scheme is more efficient than the k-means scheme. However, the proposed method, like other decomposition techniques, has its limitations. In the following paragraph, we describe the limitations of the proposed method.

One limitation of the proposed method is that the method failed in decomposing the models containing complex topology and more concave features. As shown in Fig. 2.13(a), the proposed method succeeded in detecting the three feature points representing the handle, the interior, and the exterior of the coffee mug model; however, the proposed method failed to decompose the three parts because of the failure in boundary strength characterization. Fig. 2.13(b) shows that the proposed method failed to decompose the Venus head model since the head model contains less protrusive features but more concave features. According to Hoffman and Singh's theory [51], to decompose the model containing highly concave features, the boundary strength would be more useful for guiding the decomposition process than the protrusion-based features. As a result, Katz and Tal's method would properly decompose the models containing highly concave features into parts while the proposed scheme would be more efficient and effective in extracting the elongated parts from a given mesh. Another limitation of the proposed method is the ambiguity in parts decomposition. For example, the cactus shown in Fig. 2.10(b) should be divided into three parts, that is one trunk and two branches; however, as shown in Fig. 2.10(b), the proposed method cannot solve the problem. Finally, it may be useful to develop an optimization scheme that simultaneously incorporates all the three visually salient factors. Our current implementation is based on the two visually salient features: the protrusion and the boundary strength of a part. However, as mentioned in Section



(a) (b)

Figure 2.13: The situations in which the proposed method failed to decompose a 3-D mesh into parts: (a) the coffee mug model, which has genus-1 topology; (b) the Venus head model, which contains more highly concave features than protrusive features.

2.1, human visual perception determines a part's salience by three factors: the protrusion, the boundary strength, and the relative size of a part. Although the relative size feature was not used in the mesh decomposition process, it can be easily calculated and used in the 3-D mesh retrieval process.

2.4 Concluding Remarks

We have presented a visual salience-guided mesh decomposition scheme based on Hoffman and Singh's theory of part salience [51] for extracting significant components from 3-D meshes. More specifically, the protrusion and the boundary strength are modeled as the degree of center on the surface and the total-area-of-border change respectively. To extract the visually significant components from a given 3-D mesh, these salient features are incorporated into the mesh decomposition process. The proposed scheme has three remarkable features: (1) the protrusion characterized over the entire surface is used as a guide to choose the salient representatives of the parts; (2) the total-area-of-border change characterized over the entire surface is used as a guide to find the locale of a part's boundary; and (3) the robustness against randomization of vertex coordinates benefits greatly from the incorporation of visual salience into the decomposition process. To the best of our knowledge, this is the first 3-D mesh decomposition scheme that not only identifies the part's boundaries defined by the minima rule, but also associates the part with its visual salience.

Chapter 3 A Cognitive Psychology-based Approach for 3-D Shape Retrieval

Multimedia retrieval, including the retrieval of texts [2, 16, 129], images [42, 126], audios [32], videos [122, 126], and/or 3-D graphics [21, 86, 101], has become a very active research area, especially in the last decade. Most researchers believe that the phenomenal growth of the Internet since the early 1990s has been one of the major driving forces in the development of this important area. As a result of the proliferation of multimedia content on the Internet, there is an urgent need to develop an efficient indexing mechanism to assist users with the retrieval of requested content. Among different multimedia retrieval mechanisms, 3-D shape/object retrieval [21, 86, 101] via the Internet has become increasingly important. This is because 3-D geometry/mesh processing techniques (such as creation and acquisition, modeling and editing, storage and transmission, and copyright protection and authentication) have become more mature in recent years [37, 44, 79, 98, 116]. The retrieval/recognition performance that an efficient 3-D shape retrieval system can achieve usually depends on how a 3-D shape is represented/described. In this study, we emphasize high-level abstraction and organization of 3-D shapes through machine understanding. To equip our system with a more theoretical grounding, we propose a cognitive psychology-based representation scheme for 3-D shape retrieval.

High-level organization imposed on perceived data has been explored extensively in the fields of human visual processes and computer vision. Related studies can be found in [11, 50, 51, 106, 114]. To account for human visual processes, cognitive psychologists have identified a set of properties (or rules) that are very important in the perception of a form or a shape. Meanwhile, in the field of computer vision, perceptual organization has shown that computational resources can be effectively applied to extract structural and meaningful organization from perceived data. Furthermore, perceptual organization can be regarded as an intelligent process that can perform high-level abstraction for information understanding. In the literature, perceptual organization has been applied to the segmentation of range images [28], 2-D images [110], textures [102], patterns [134], and contours [136]. However, existing 3-D shape retrieval techniques lack a direct link to perceptual organization. Below, we briefly review some existing 3-D shape retrieval techniques and point out their common shortcomings in terms of perceptual organization.

In order to build an efficient 3-D shape retrieval system, one may ask: What kind of description/representation is most needed in a 3-D shape retrieval/recognition process? It is well known that there is always a trade-off between a representation scheme and its corresponding recognition result. A coarse description/representation usually results in an efficient search process, but it sacrifices the accuracy of the search result. Examples of coarse representation include normal and cord-based distribution [95], shape distributions [92], and higher-order moment analysis [30, 95, 123, 135]. On the other hand, a thorough description/representation scheme [49, 138] usually yields an accurate recognition/retrieval result, but the efficiency of the whole process is degraded. Among the different types of representation schemes, the frequency-domain approach [64, 123, 135] is often adopted because of its elegant formulation (e.g., frequency decomposition) and beautiful mathematical properties (e.g., invariance). However, this approach lacks a direct link to the operation of human visual systems. Kazhdan *et al.* are some of the few researchers who have incorporated salient features (e.g., symmetry [65]) and intrinsic properties (e.g., shape anisotropy [66]) into 3-D shape retrieval systems. Since the features they adopt are not related to human visual processes, either descriptor can be used in conjunction with existing methods. We firmly believe that if a 3-D shape retrieval system does not consider the functions of the human visual system, its discriminating power is to some extent limited.

In this chapter, to deal with the above-mentioned drawbacks, we incorporate a set of principles that originated in the field of cognitive psychology into the design of 3-D shape analysis and retrieval algorithms. In the modeling phase, a 3-D shape represented in mesh form is first broken up into parts such that the functions of the human visual system can be appropriately mimicked. Next, the decomposed parts are individually analyzed and quantified according to the psychological theory of visual salience [51]. Using the above concept, one can label a 3-D shape and then decide which components of the shape are the most salient. In order to properly represent 3-D meshes for the subsequent retrieval process, the spherical parameterization scheme proposed in [100] is adopted to map the decomposed parts onto the surface of a unit sphere. With this representation, the degree of similarity between a query provided by a user and models pre-stored in the database can be easily compared within a canonical coordinate system. Moreover, in the retrieval phase, one can use these visually salient components to efficiently retrieve similar 3-D shapes/objects from the database.

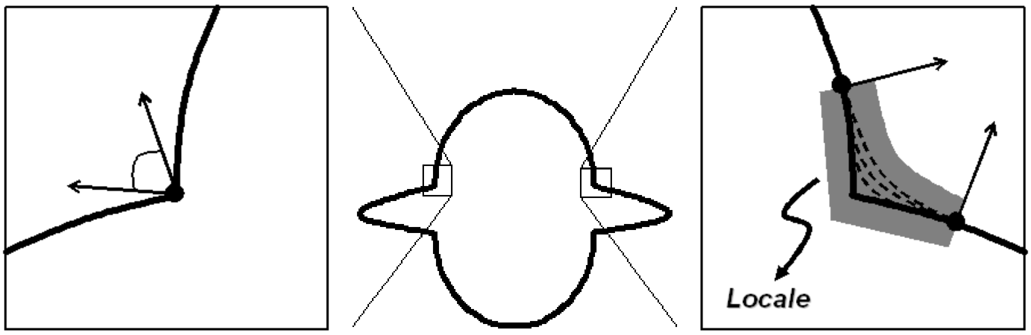
The remainder of this chapter is organized as follows. In Section 3.1, we introduce the theory of part salience and illustrate its importance in the perception of parts. In Section 3.2, we propose computational process for realizing the qualitative salient features, and describe in detail how to incorporate visual salience into the representation scheme for 3-D shape retrieval. The experiment results are presented in Section 3.3. Finally, in Section 3.4, we present our conclusions.

3.1 Theory of Part Salience and Its Importance in Visual Perception

In [51], Hoffman and Singh propose the theory of part salience, which states that at least three factors determine the salience of a part: the protrusion, the boundary strength, and the relative size of the part. We now present quantitative definitions of these salient factors and describe their importance in visual processes.

Protrusion of A Part This factor is the degree to which a part protrudes from its main body. For 2-D silhouettes, it can be quantified as the ratio of the perimeter of the part (excluding its base) to the sum of its base lengths. For 3-D shapes, the base of a part is referred to as the minimal surface formed by the boundary curve of the part. Hence, the protrusion of a 3-D part can be quantified as the ratio of the area of the part's surface to the area of its base surface.

Strength of A Part's Boundary The principle of transversality asserts that a part's boundaries are usually located at the concave creases, as shown in Fig. 3.1(b). Possible quantitative definitions of the boundary strength include the turning normals and locale turning, as shown in Figs. 3.1(a) and 3.1(c), respectively. Obviously, the indication of the normal direction must have a global orientation consistency so that the boundary strength can be captured precisely. The discriminating capability of turning normals and locale turning is shown by



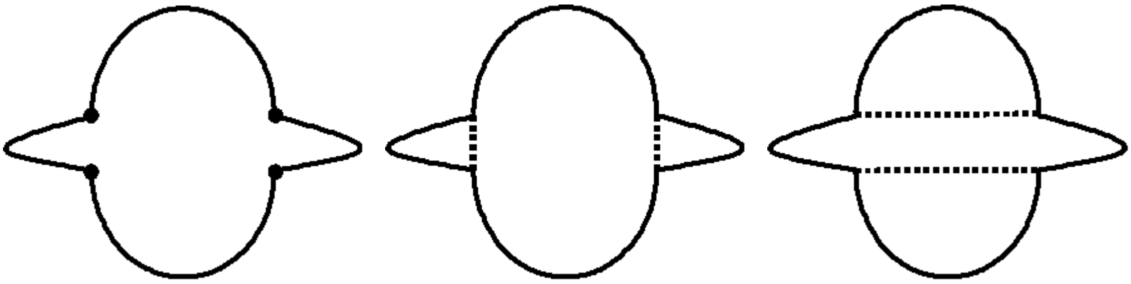
(a) Turning normals (b) 2-D Silhouette (c) Locale turning

Figure 3.1: Illustration of turning normals and locale turning at the boundary of a 2-D silhouette.

the following examples. For 2-D silhouettes, a crease boundary usually has two normals, and the angle between them can, in one sense, represent the strength of that boundary. On the other hand, for potentially smooth boundaries, which are represented by the dotted lines in Fig. 3.1(c), there is one normal at every point along a curve. To deal with this problem, Hoffman and Singh [51] proposed that the measure of turning in an appropriate region near the boundary should be obtained. As shown in Fig. 3.1(c), the gray region is the so-called *locale*[5] and the normals on its two sides (i.e., the so-called *locale turning*) are used to characterize the strength of the smooth boundary. For 3-D shapes, the principal curvatures can be used to measure the strength of a part's boundary.

Relative Size of A Part This factor indicates the size of a part relative to the whole object. For 2-D silhouettes, it can be defined as the ratio of the area

[5] By the definition in [51], a locale is an appropriate region near (but not infinitesimally near) a negative minimum of the curvature in which we can explore how the curve evolves.



(a) A part's boundaries (b) A part's cuts (c) A part's cuts

Figure 3.2: A part's boundaries and cuts in a 2-D silhouette (Re-sketched from [51]).

of a part to the area of the whole object. For 3-D shapes, the relative volume can be used to measure a part's relative size.

Having reviewed the factors that may be used to determine the salience of a part, we now discuss their effects on both visual perception and decomposition processes. For simplicity, the following discussion is based on 2-D silhouettes; however, the concept can be easily extended to 3-D models. Fig. 3.2 shows the boundaries and cuts of parts of a 2-D silhouette, indicated by isolated points and dotted lines, respectively. Note that, in Fig. 3.2(a), the four boundaries are used to form possible cuts; and, in Figs. 3.2(b) and 3.2(c), each part is generated by exactly one cut. According to the visually salient properties of interest, a 2-D silhouette may have different interpretations. For example, the 2-D silhouette might be interpreted as an alien's head with a pair of protrusive ears when the salience of the part is determined primarily by its protrusion (i.e., the part's cuts in Fig. 3.2(b)). On the other hand, the same silhouette might be interpreted as an unidentified flying object when the part's salience is determined primarily by its relative size (i.e., the part's cuts in Fig. 3.2(c)). As a result, part salience affects not only the low-level visual processes that determine how a shape is really decomposed, but also the high-level visual processes that determine the interpretation of the shape. In order to precisely determine a part's cuts, another independent theory that incorporates a priori knowledge about the shape is usually required. In the early 1980s, 3-D object recognition was a popular research topic [28]. Also, among the large number of research issues, 2-D perceptual organization [106] and recognition-by-components (or parts) [11, 50, 114] were two important directions. However, their development was hindered by some ill-posed early vision problems, such as edge detection and image segmentation. Since these issues could not be solved, 2-D perceptual organization and recognition by 2-D components (or parts) could not be converted into "complete" computational processes, so they both failed. Nowadays, there are many 3-D models distributed worldwide. Since 3-D models (or meshes) are not restricted by the limitations of 2-D images, perceptual organization is now possible in 3-D cases. In this chapter, we develop an efficient and accurate 3-D retrieval system, which incorporates Hoffman and Singh's cognitive psychology theory into the representation scheme, such that the human visual perception mechanism can be properly mimicked.

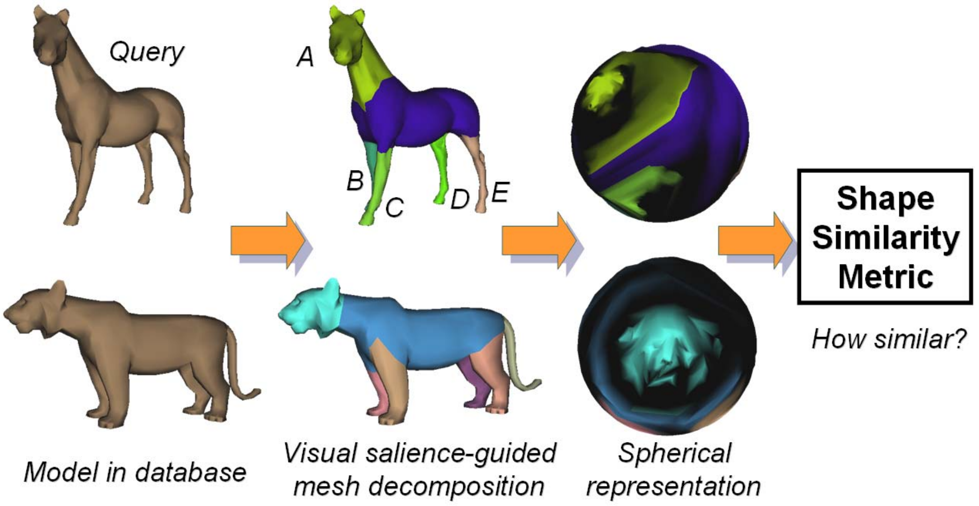


Figure 3.3: The flowchart of the proposed cognitive psychology-based scheme for 3-D shape analysis and retrieval.

3.2 A Cognitive Psychology-based Approach for 3-D Shape Retrieval

In this section, we introduce a cognitive psychology-based approach for 3-D mesh retrieval. Fig. 3.3 shows the flowchart of the proposed cognitive psychology-based scheme for 3-D shape analysis and retrieval. In Section 3.2.1, we provide a brief introduction to how visual salience-guided mesh decomposition works. In Section 3.2.2, we describe how to extract a set of cognitive psychology-based features (i.e., protrusion, boundary strength, and relative size) from the decomposed parts. In Section 3.2.3, we propose a spherical scheme for representing a 3-D shape and part salience. In Section 3.2.4, a shape similarity metric, which can be used to compare the difference between two 3-D shapes, is described in detail.

3.2.1 Review of Visual Salience-Guided Mesh Decomposition

We now briefly review the previously proposed mesh decomposition scheme called "visual salience-guided mesh decomposition" [77] because we use the algorithm in this chapter. In [77], we analyzed the theory of part salience proposed by Hoffman and Singh [51] and showed that it can be converted into computational processes to extract significant components from 3-D meshes. More specifically, the protrusion and boundary strength are modeled as the degree of center on the surface and the total-area-of-border change, respectively. These visually salient features are incorporated into the mesh decomposition process based on the following rules.

1. The protrusion degree characterized over the entire surface is used as a guide to choose the salient representatives of the parts.

2. The boundary strength characterized over the entire surface is used as a guide to find the locale of a part's boundary.

As shown in Fig. 3.4(a), for a given 3-D mesh, the protrusion is first characterized as the degree of center over the entire surface. After the protrusion characterization is obtained, all the constituent triangular facets of the mesh are labeled with a numerical protrusion degree ranging from 0 to 1. Note that, in Fig. 3.4(a), a darker color represents a protrusion degree close to 0, while a lighter color means the protrusion degree is close to 1. According to the protrusion characterization,

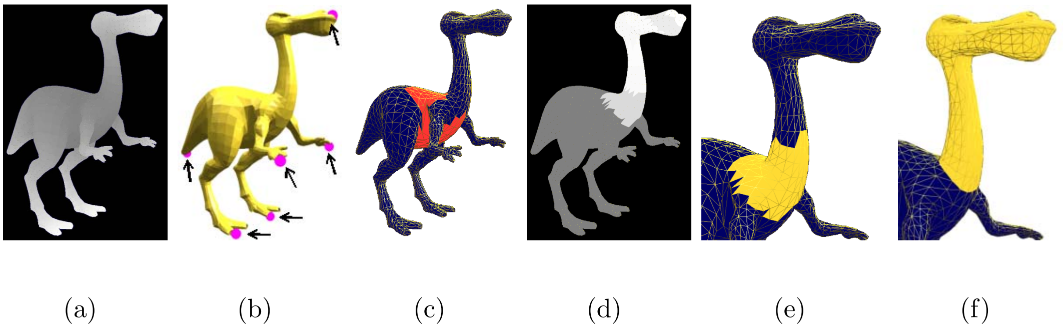


Figure 3.4: The visual salience-guided mesh decomposition process proposed in [77]: (a) protrusion characterization; (b) choosing salient representatives; (c) the termination-based constraint; (d)-(f) boundary strength characterization, finding the locale of boundary, and generating the boundary of a specific part.

a set of salient representatives of the parts can be selected from the 3-D mesh using a criterion based on the local maximum of protrusion degrees. In addition, a termination base used to impose a constraint on a subsequent process can be determined using a certain simple thresholding scheme. Figs. 3.4(b)-3.4(c) show six salient representatives and one termination base extracted from a dinopet model.

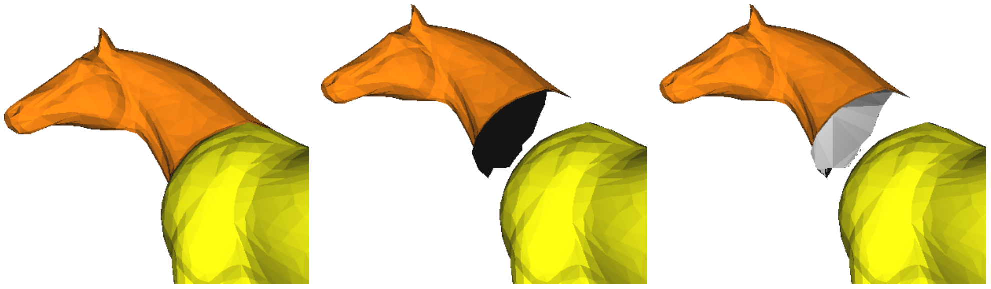
Using a part's salient representatives as the seeds of Dijkstra's algorithm, we can execute multiple region growing processes, which will stop when they "hit" the termination base. In response to the boundary's evolutionary process, the boundary strength can be quantitatively characterized as the change of the adjacent-area-of-border. As shown in Fig. 3.4(d), the change of the adjacent-area-of-border can be used to indicate the locale that is close to a part's boundary. To locate the locale, we propose a systematic method based on wavelet analysis to locate the boundary in a coarse-to-fine manner. Fig. 3.4(e) shows the result after the locale of the part's boundary is found (the gray area close to the neck of the dinopet). Katz and Tal's method [62] is applied in order to derive the exact boundary of the part. After the above-mentioned mesh decomposition process is completed, as shown in Fig. 3.4(f), each triangular facet in a mesh is symbolically labeled such that the corresponding part and the main body can be identified.

In the decomposition process in [77], since the features used to guide the process are closely related to Hoffman and Singh's theory of visual salience, the process itself can appropriately mimic the function of a human visual system decomposing a 3-D mesh into parts. Note that the features defined above only guide the mesh decomposition process (i.e., choose a part's representative and find the locale close to the part's boundary). To perform 3-D shape retrieval, the overall protrusion degree of every constituent part and the boundary strength between every part and the main body of a 3-D shape, as well as the relative size of every part to the whole object, have to be calculated based on Hoffman and Singh's theory [51]. In this way, all extracted features become real values that can be used to establish a set of indices for an efficient 3-D shape retrieval task. In what follows, we describe in detail how each feature is converted into a real value.

3.2.2 Part Salience Characterization

In this section, we report how to systematically calculate the protrusion degree of a part, the boundary strength between a part and its corresponding main body, and the relative size of every part. As mentioned in the previous section, after the mesh decomposition scheme is executed, the parts and their corresponding boundaries of the given 3-D model are extracted and identified. Due to the nature of a mesh-based object, the shape of an extracted part is represented by a form of open mesh, while its corresponding boundary is composed of a set of edges. In order to realize the quantitative definitions for part salience mentioned in Section 3.1, we propose the following characterization methods:

Protrusion of A Part In [51], Hoffman and Singh mention that the protrusion of a 3-D part should be defined as the ratio of the area of the part's surface to that of its base's surface. For a 3-D mesh, the base of a part is referred to as the minimal surface formed by the boundary polygon of the part. To obtain a part's base surface, we first apply the cutting algorithm proposed by Guéziec *et al.* [44] to the marked boundary edges in order to partition the connectivity between the part and its body. Next, Barequet and Sharir's filling gaps algorithm [4] is applied to form such a base surface, after which the protrusion of a mesh-based part can be characterized. Moreover, in order to normalize the protrusion feature, Gaussian normalization (as used in [104]) is applied to map the protrusion value

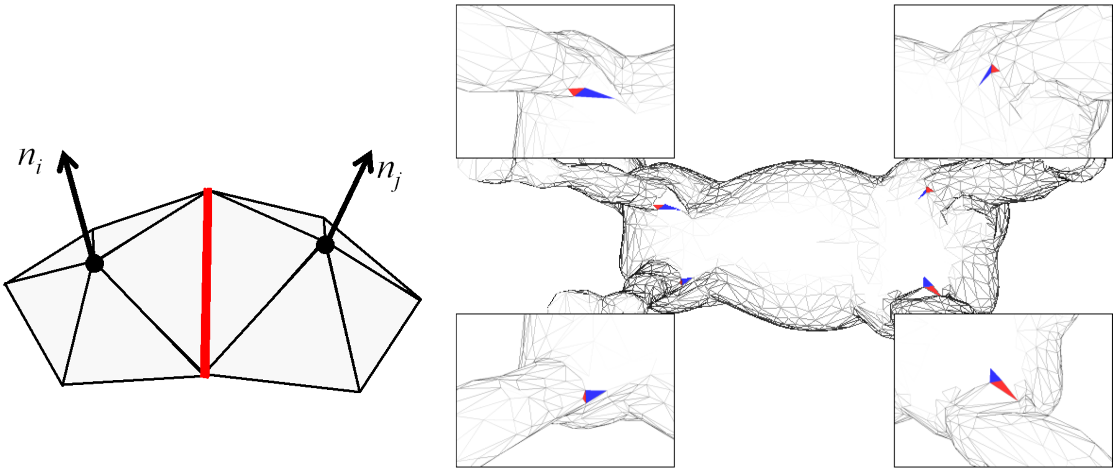


(a) Labeling (b) Cutting (c) Filling

Figure 3.5: The protrusion of a 3-D part can be obtained by first cutting the decomposed part (as in (b)), applying the filling holes algorithms [76] to form its base (as in (c)), and then calculating the ratio of the area of the part's surface to the area of the base.

in the range $[0, 1]$. Fig. 3.5 illustrates the process of how a base surface of a 3-D part is formed. Note that neither the filling gaps algorithm [4] nor the filling holes algorithm [76] can guarantee that a generated mesh will form a manifold surface.

Strength of A Part's Boundary In [51], the boundary strength is characterized by the turning normals at the crease boundaries. For a 3-D mesh, the dihedral angle formed by two adjacent faces has been commonly used to characterize the local curvature of an edge. However, as noted in [54], this measure is sensitive to noise due to its small support region. In order to ensure that the characterization of boundary strength is reasonable, we apply the extended second order difference (ESOD) operator [54] to characterize the local curvature of each boundary edge and then find the minimum value to represent the strength of a part's boundary. Fig. 3.6(a) illustrates the support of the ESOD operator.



(a) (b)

Figure 3.6: Illustration of the boundary strength characterization: (a) the support of the ESOD operator [54]; (b) the visualization of the boundary edges that possess the strongest boundary strength.

Fig. 3.6(b) represents the visualization of the boundary edges that possess the strongest boundary strength. Note that since the boundary strength is inversely proportional to the corresponding radian value, all radian values falling within the range $[-\pi, \pi]$ should be normalized by changing their signs and then mapping them in the range $[0, 1]$.

Relative Size of A Part In [51], the relative size of a 3-D part is defined as the ratio of the volume of the part to that of the whole object. However, calculating the volume of a 3-D mesh is tedious work, especially for an open and non-manifold mesh. To simplify this task, we characterize the relative size of a part by an area ratio. That is, the ratio between the area of the constituent facets that enclose a 3-D part and that of the constituent facets of the whole object.

Table 3.1: Part salience of the query “horse” shown in Fig. 3.3.

Part	Protrusion	Boundary strength	Relative size
A	0.4476	0.61692	0.21006
B	0.4749	0.68476	0.05041
C	0.5364	0.69291	0.04940
D	0.4561	0.69982	0.06867
E	0.4152	0.58425	0.08684

Table 3.1 lists the part salience extracted from the horse model in Fig.3.3 using the characterization methods described above. Note that the three visually salient features (i.e., protrusion, boundary strength, and relative size) are invariant to rotation, scaling, and translation respectively. Moreover, the features have nothing in common in terms of perceptual organization. More precisely, the protrusion-based feature is characterized in response to a part’s shape. The boundary strength-based feature quantitatively proposed here describes the conjunction of a part and its main body. The relative size-based feature, on the other hand, can quantitatively “measure” the relationship between a part and the whole object.

3.2.3 A Spherical Representation Scheme for 3-D Meshes

In the previous section, we described how the set of human perception-related features is quantified. However, these features are still not sufficient to achieve a “thorough” description of a 3-D shape, so retrieval based on the above-mentioned feature set cannot achieve a satisfactory result. Under these circumstances, some relational features have to be introduced to enhance the discrimination capability of a shape descriptor. In the following, we describe in detail how to establish the proposed representation scheme that incorporates a part's salience and relational information for 3-D shape retrieval.

To represent the geometrical relations among the parts of a 3-D shape, we propose the use of spherical parameterization to map the labeled 3-D mesh onto a unit sphere. In our investigation, we found that the spherical parameterization algorithm proposed by Praun and Hoppe [100] is very suitable for this task, because it can efficiently generate a valid sphere embedding while minimizing a certain distortion metric. Fig. 3.8(a) shows the result of parameterizing the horse model in Fig. 3.3 onto the spherical domain. Clearly, there are two advantages to this method. On the one hand, the relative positions of the parts are retained by this spherical representation scheme. On the other hand, by treating the degree of a part's salience as the value of a function defined on the unit sphere, all salient features can be encoded as well. However, a variety of mesh resolutions makes the subsequent matching process very difficult. More precisely, in order to calculate the shape similarity, a uniform sampling within the spherical domain is necessary.

It is well known that there are only three types of regular meshes that satisfy the uniform sampling requirement for 3-D space. Therefore, after spherical parameterization, we apply the sphere tessellation process described in [48] to

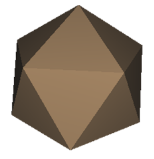
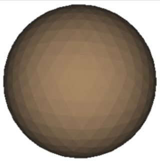
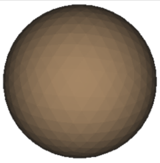
Frequency	1	2	3	4
# of faces	20 (icosahedron)	80	180	320
Geometry				
Frequency	5	6	7	8
# of faces	500	720	980	1280
Geometry				

Figure 3.7: Illustration of the n-tessellation of a unit sphere.

quantize the spherical domain. The process works as follows: First, an icosahedron is generated and translated to its center of mass. Second, each triangle is divided into n^2 sub-triangles, where n is the subdivision frequency. Third, the coordinates of all vertices are unitized such that all the vertices lie on the unit sphere. Fig. 3.7 shows a set of tessellated spheres with different subdivision frequencies.

To simplify the notations, let S denote a 3-D mesh, obtained by applying the spherical parameterization method [100] to a labeled 3-D mesh, and let D denote an n-tessellated sphere. Assume the two meshes, S and D, are scaled and fitted onto the unit sphere. Based on the location of each mesh's center-of-mass, we further align each mesh into the correct position. We now describe the steps required to construct a spherical domain-based shape descriptor:

1. Map the position of a part's representative on S onto a facet f_i on D, where i is the integer index of a facet.

2. Map a radially symmetric function to the support neighborhood of the facet f_i on D.

Since a part's representative belongs to a facet on S (i.e., all the circles indicated by an arrow in Fig. 3.8(a)), our first step is to project the normal of the facet onto a facet f_i on D (i.e., the dark triangles indicated by an arrow in Fig. 3.8(b)). Note that a part's representative can be determined either by calculating the center of mass of the region belonging to a part on S or by using the salient representatives chosen in the mesh decomposition process.

The second step spreads a part's salience over the sphere domain D, beginning with support neighborhood construction. Given a facet, f_i, obtained in the first step, the set of neighboring facets, $F_i = \{f_{i,j}\}$, is constructed by applying Dijkstra's algorithm to the dual edges of the mesh graph of D. The cost of computing each dual edge is defined as the distance between the centers of mass of two corresponding facets. The construction is constrained by a radius threshold.

In each support neighborhood, F_i, we map a radially symmetric function to this region. More precisely, the mapping, $\phi_i^k(j)$, is computed using the following formula:

$$\phi_i^k(j) = \left(1 - (\frac{dist(f_i, f_{i,j})}{dist_i^{max}})\right) \times V_i^k, \qquad (3.1)$$

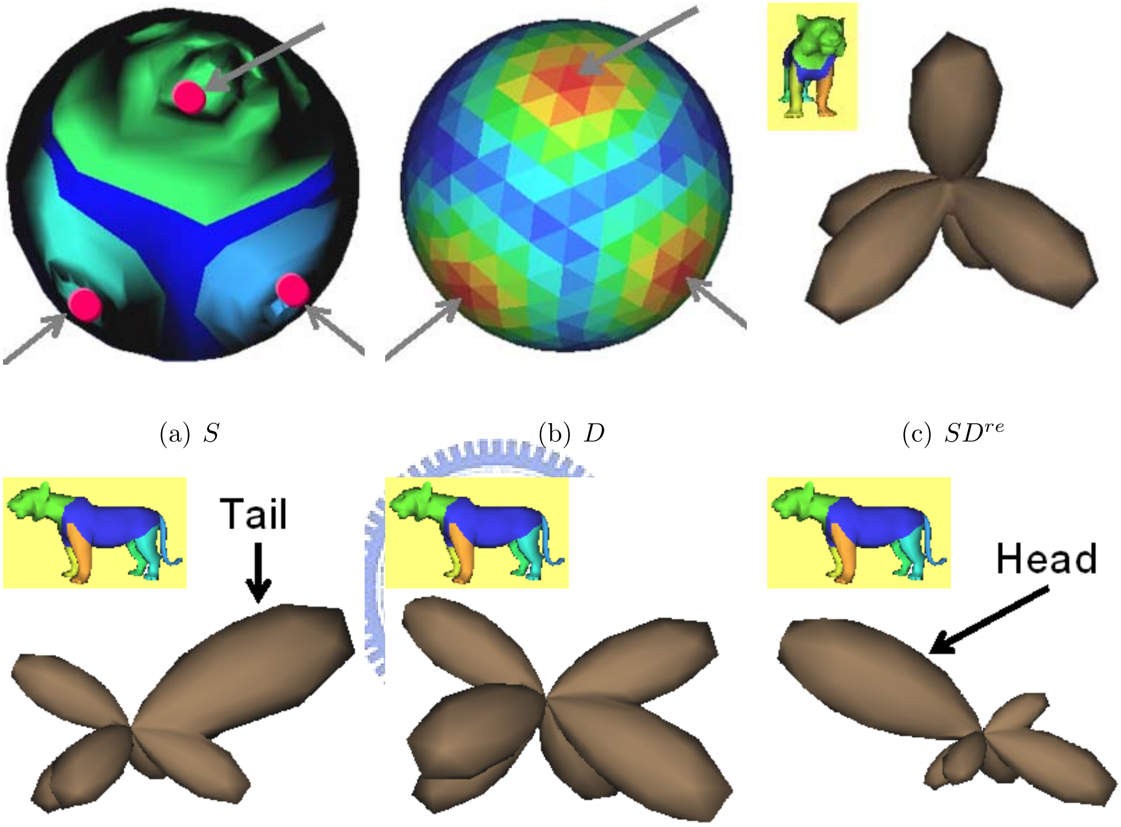
where $dist(f_i, f_{i,j})$ is the distance between the center of mass of the facet $f_{i,j} \in F_i$ and that of the facet f_i. Also, $dist_i^{max}$ is the maximum distance between the center of mass of the facet f_i and that of any facet in the neighborhood. $V_i^k \in [0, 1]$ is the degree of salience of the part with the corresponding facet, f_i, on D. The superscript $k \in \{re, pro, bs, rs\}$ is used to specify the type of visual salience, where re, pro, bs, rs stand for relation, protrusion, boundary strength, and relative size, respectively. Note that the term V_i^{re} in Eq. (3.1) is always set to 1. In this way, a relation-based shape descriptor can be constructed for a coarse search. Fig. 3.8(b) shows that a part's relational information is spread over its corresponding neighborhood on the tessellated sphere D. On the other hand, Fig. 3.8(c) shows the visualization result after the relation-based shape descriptor is represented in the spherical domain. With regard to part salience-based shape descriptors, their visualization results are shown in Figs. 3.8(d)-3.8(f), respectively.

In [48], Hebert *et al.* use a similar spherical representation scheme to map the curvature distribution of a 3-D surface onto a unit sphere. In contrast to their intrinsic spherical representation for 3-D object recognition, our system provides a coarse-to-fine search scheme as follows:

1. A recognition-by-components strategy for coarse search;

2. A recognition-by-visually-salient-components strategy for fine search.

More precisely, our relation-based shape descriptor implements the recognition-by-components strategy to perform a coarse search. On the other hand, the visual salience-based shape descriptor realizes the recognition-by-visually-salient-components strategy to refine the results (up to the best m objects) retrieved in the coarse search stage.



(a) S

(b) D

(c) SD^{re}

(d) SD^{pro}

(e) SD^{bs}

(f) SD^{rs}

Figure 3.8: Illustration of the spherical domain-based shape descriptor construction. The original 3-D shape is first parameterized onto the spherical domain. Next, the spherical domain is tessellated as an icosahedron. Based on the resulting icosahedron, the shape descriptor stores the geometrical relations of the parts.

3.2.4 Comparing 3-D Shapes on A Normalized Sphere

Since the spherical parameterization process creates a one-to-one mapping between the points on the surface of a 3-D shape and those on a representative sphere, the shape descriptor mentioned in Section 3.2.3 can be regarded as a function defined on the representation sphere. It is obvious that some existing mathematical analysis tools (e.g., spherical harmonic analysis [31] and spherical wavelets [107]) can be utilized to perform shape analysis and calculate shape similarity. However, such tools do not have any direct correlation with the functions of the human visual system. In this chapter, we compare the shapes of two distinct targets directly in the proposed spherical domain. The reason is twofold: First, the proposed shape analysis algorithms (including mesh decomposition and part salience characterization) are explicitly considered as natural ways to mimic the human visual system. Second, mathematicians usually favor elegant formulations (e.g., frequency decomposition) and beautiful mathematical properties (e.g., invariance), while cognitive psychologists prefer a mechanism that is close to the way the human visual system operates. In this work, we have tried to build a 3-D shape retrieval system that would satisfy both mathematicians and cognitive psychologists. We now describe the proposed similarity metric in more detail.

Let $\Phi^k(A)$ and $\Phi^k(B)$ denote the spherical domain-based shape descriptors of two mesh-based shapes A and B, respectively. The superscript k indicates which type of shape descriptor (either a part's relation or visual salience) should be

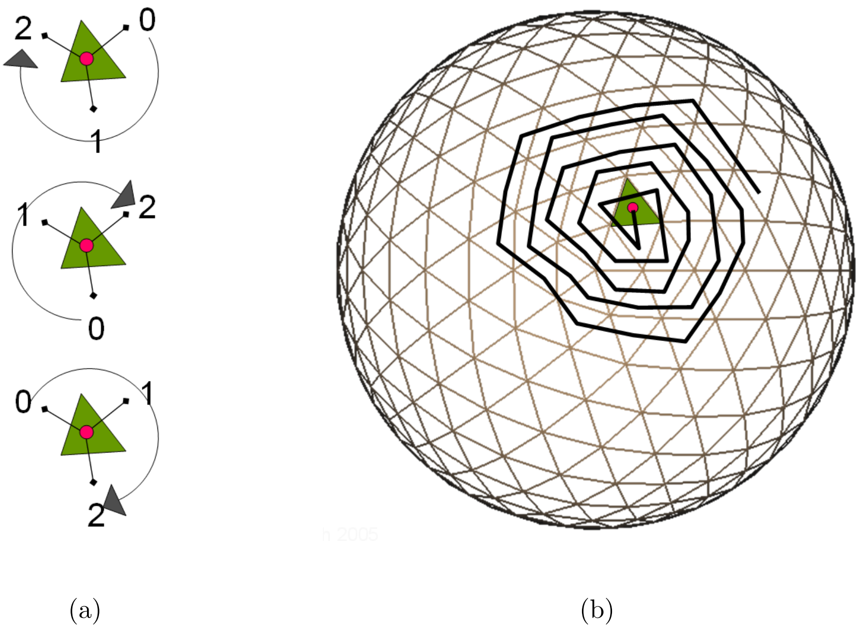


Figure 3.9: Illustration of how to speed up the similarity calculation.

used to perform the retrieval task. Now, the distance between A and B subject to a certain spherical rotation, R, can be defined as

$$Dist(A, B, R, k) = \sum_{D} min(\Phi_I^k(A), \Phi_R^k(B)), \tag{3.2}$$

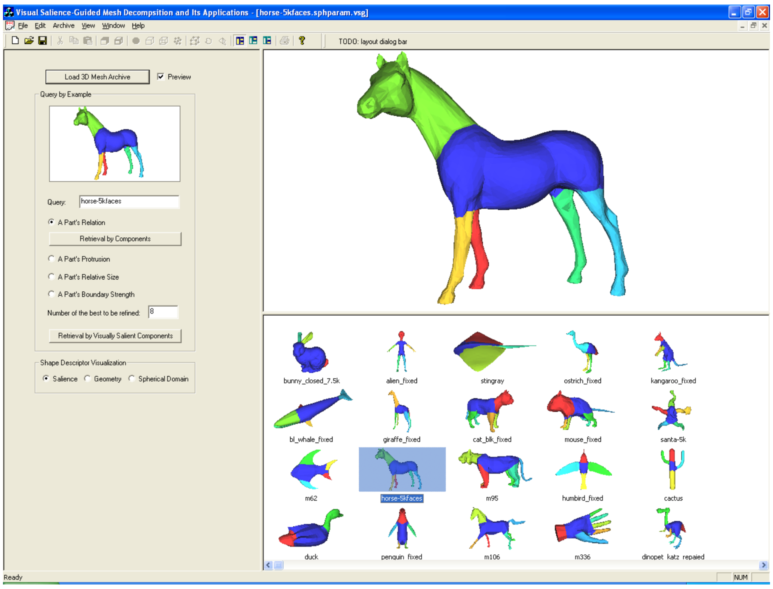
where D is the tessellated sphere domain and I is the identity matrix. Under these circumstances, the similarity between A and B becomes

$$Similarity(A, B, k) = \max_{R} Dist(A, B, R, k). \tag{3.3}$$

Eq. (3.3) maximizes $Dist(A, B, R, k)$ over all possible spherical rotations, R. From Eqs. (3.2)-(3.3), it is apparent that the proposed shape matching algorithm continuously rotates one of the shapes until the salience of the parts perceived by the human visual system is maximized. In [48], Hebert *et al.* stated that the number of rotations for which $Dist(A, B, R, k)$ should be evaluated is confined by the facet number of a tessellated sphere. Since both A and B are represented in the tessellated spherical domain, the comparisons should be directed to finding the best correlation between the instances of $\Phi(A)$ and those of $\Phi(B)$. More precisely, given an initial correspondence (i, j) for f_i on $\Phi(A)$ and f_j on $\Phi(B)$ respectively, only three rotations should be evaluated and applied to $\Phi(B)$. As shown in Fig. 3.9(a), each facet is surrounded by only three neighboring facets. Fig. 3.9(b) shows that given an initial correspondence, the remaining correspondence between the facets of $\Phi(A)$ and $\Phi(B)$ can be determined by applying a counterclockwise (or clockwise) traversal of the remaining facets on $\Phi(A)$ and $\Phi(B)$. Hence, a set of correspondence tables, $\{(P_i, P_{i,j})\}$, can be constructed to speed up the similarity calculation, where the pair $(P_i, P_{i,j})$ provides a valid correspondence between the facets of $\Phi(A)$ and $\Phi(B)$. Now, based on the form of Eq. (3.1), the right-hand term of Eq. (3.2) can be rewritten as

$$Dist(A, B, R, k) = \sum_i min(\phi^k(P_i), \phi^k(P_{i,j})). \tag{3.4}$$

As noted in [48], the algorithm for calculating Eq. (3.4) tries all possible rotations and the resolution is controlled by how the sphere is tessellated. Under these circumstances, the global optimum of $Dist$ can always be found without an initial estimation of the transformation.



☑ Preview

Query by Example

Query: horse-5kfaces

● A Part's Relation
○ A Part's Protrusion
○ A Part's Relative Size
○ A Part's Boundary Strength

Number of the best to be refined: 8

Shape Descriptor Visualization
● Salience ○ Geometry ○ Spherical Domain

Figure 3.10: The user interface of our 3-D shape retrieval system.

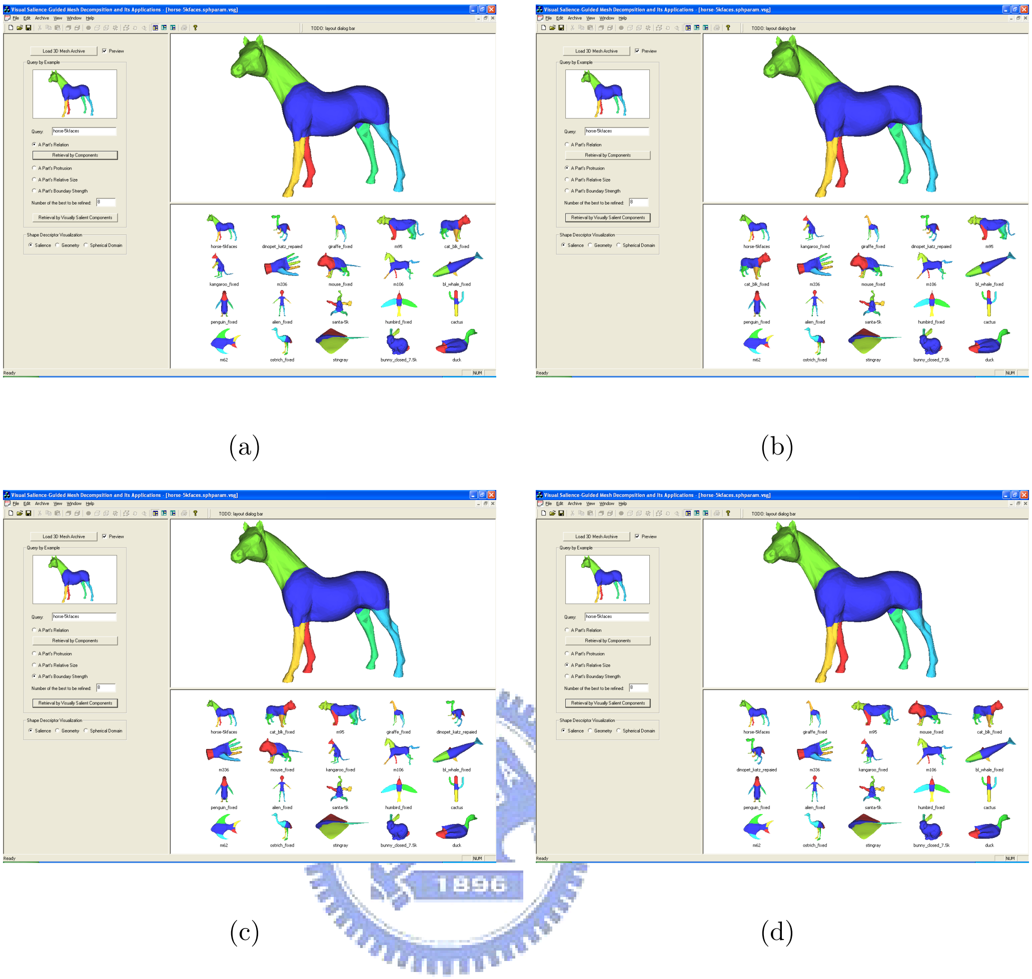
3.3 Experiment Results

We counducted a series of experiments to test the effectiveness of the proposed method. The data set used in our experiments was comprised of 20 triangulated meshes. Note that in order to properly apply spherical parameterization, all the models must be genus-0 and closed meshes. Fig. 3.10 shows the user interface of the proposed 3-D shape retrieval system. According to the different functionalities, the interface is split into three sub-windows: (1) Thumbnail-View, (2) Model-View, and (3) Operating-View. Below, we describe the functions of the three sub-windows in detail.

In the Thumbnail-View sub-window (the bottom half of the right column of Fig. 3.10), our system lists all of the thumbnail pictures of 3-D models in the database. Note that after a retrieval/search command is executed, all the thumbnail pictures will be listed from left-to-right and top-to-bottom, based on the degree of shape similarity between the query and the model in the database. Since the thumbnail pictures listed are types of 2-D images, it is necessary to provide users with a method for viewing the 3-D mode from different angles. We, therefore, designed the Model-View sub-window to perform 3-D manipulations such as rotation, zoom in, zoom out, and panning. Within the Model-View sub-window, as shown on the top right-hand side of Fig. 3.10, a user can manipulate a model in 3-D by using the drag command of the mouse. In the Operating-View sub-window, as shown on the left-hand side of Fig. 3.10, a user can choose an existing model from the Thumbnail-View sub-window as a query and specify which strategy he/she would like to adopt to perform the retrieval task. We now report the experiment results of the coarse search, the fine search, and the comparison.

3.3.1 Recognition-by-components as A Coarse Search

In this experiment, a recognition-by-components search strategy was adopted and the horse model was chosen as the query to find a set of similar models in the database. Fig. 3.11(a) shows the search results after the strategy was executed. Since a part relation-based shape descriptor manages the search strategy, the purpose of the search was to retrieve those models that had a similar part relation to that of the query. It can be seen from Fig. 3.11(a) that most of the animal



(a)

(b)

(c)

(d)

Figure 3.11: Search results after the coarse-to-fine search strategy was executed: (a) coarse search using a relation-based shape descriptor; (b) fine search using a protrusion-based shape descriptor; (c)fine search using a boundary strength-based shape descriptor; (d) fine search using a relative size-based shape descriptor.

models retrieved as the first six candidates have four feet and appear at the top of the Thumbnail-View sub-window.

3.3.2 Recognition-by-visually-salient-components as A Fine Search

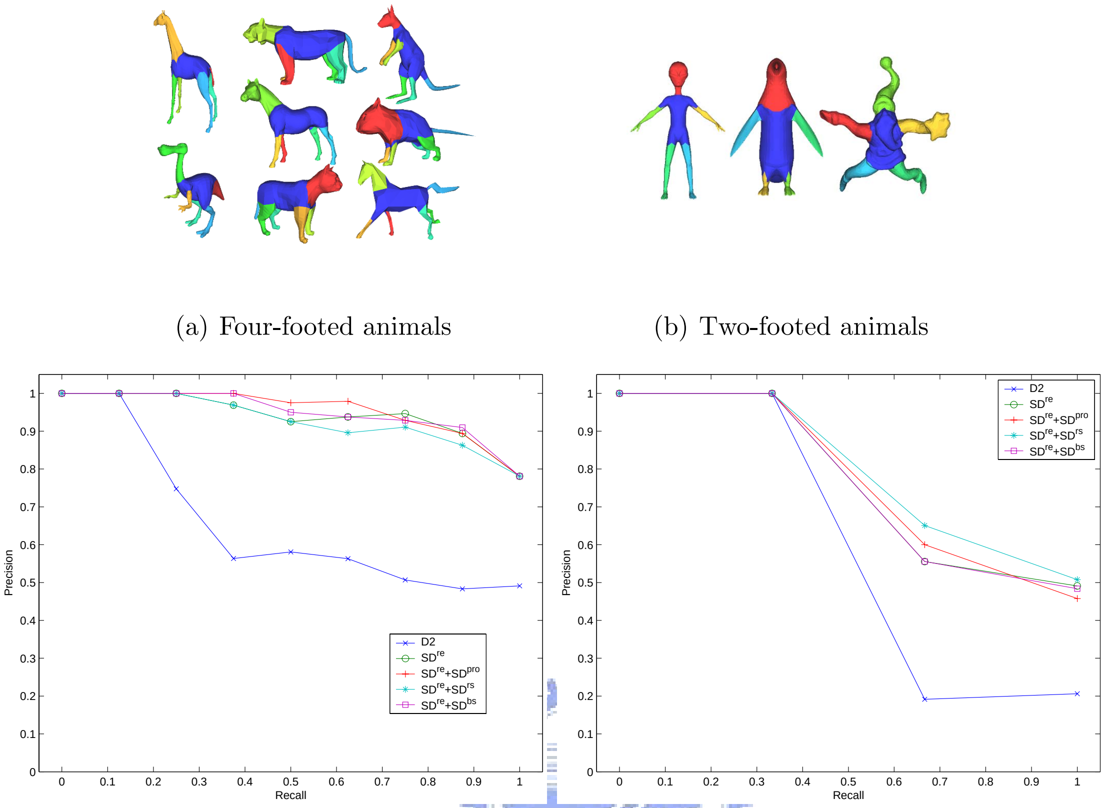
In this experiment, the recognition-by-visually-salient-components search strategy described in Section 3.2.3 was adopted to refine the best $m(=8)$ models shown in Fig. 3.11(a). Since there are three types of part salience-based shape descriptors (i.e., SD^{pro}, SD^{bs}, and SD^{rs}), this experiment was comprised of three parts. The intent of the first part was to use the protrusion-based shape descriptor to refine the best eight models shown in Fig. 3.11(a), such that the models with a similar degree of part protrusion to that of the horse model could be placed at the top of the list in the Thumbnail-View sub-window. Since the protrusion-based shape descriptor takes a part's relation and protrusion into account, using it as the fine search strategy can retrieve those models that not only hold a similar part relation to that of the horse model, but also possess a similar part protrusion to that of the model. Fig. 3.11(b) shows the search results after the protrusion-based shape descriptor was executed in the fine search process. Because of the introduction of the protrusion-based feature, it can be seen from Fig. 3.11(b) that most of the models listed at the top of the Thumbnail-View sub-window have a similar degree of part protrusion to that of the horse model.

The intent of the second part of the experiment was to use the boundary strength-based shape descriptor to refine the best eight models shown in Fig. 3.11(a), such that the models with a boundary strength similar to that of the horse model could be placed at the top of the list in the Thumbnail-View sub-window. Fig. 3.11(c) shows the search results after the boundary strength-based shape descriptor was applied. Due to the use of the boundary strength-based feature, most of the models at the top of the list in Fig. 3.11(c) have a similar parts-body conjunction to that of the horse model.

In the third part of the experiment, we used the relative size-based shape descriptor to refine the best eight models shown in Fig. 3.11(a). Fig. 3.11(d) shows the search results after the relative size-based shape descriptor was applied to the fine search process. From the results, it is obvious that the introduction of the relative size-based feature improved the search outcome.

3.3.3 Comparison to D2 Shape Distribution

In this experiment, we compared our spherical domain-based shape descriptors against the D2 shape distribution-based approach [92] by using the well-known precision-recall analysis. To perform this task, the models in the database were classified into four classes according to a part's functionality: (1) four-footed animals, (2) two-footed animals, (3) birds, and (4) miscellaneous. After the classification was done, each model in a specific class was used as a query to perform the retrieval task. Next, the retrieved models were classified into relevant and irrelevant in order to calculate the precision and recall. For all the models in the same class, we then averaged their precisions and recalls. These results were plotted on the precision-recall curves for comparision. Figs. 3.12(a)-3.12(b) show the 2-D images of the four-footed and two-footed animal models, respectively. On the other hand, Figs. 3.12(c)-3.12(d) show the precision-recall plots for the classes of four-footed and two-footed animal models, respectively. It can be observed



(a) Four-footed animals

(b) Two-footed animals

(c) Analysis for the class (a)

(d) Analysis for the class (b)

Figure 3.12: The precision-recall curves comparing our methods against the D2 shape distribution-based method presented in [92].

from the plots that the proposed shape descriptors performed better than the D2 shape distribution method.

3.4 Concluding Remarks

We have presented a cognitive psychology-based approach for 3-D shape analysis and retrieval. The proposed scheme has four remarkable features: (1) a 3-D shape retrieval system that mimics human visual perception and recognition can be constructed by incorporating Hoffman and Singh's theory of part salience into the design of shape analysis and retrieval algorithms; (2) a recognition-by-components search strategy can be achieved by using part relational information; (3) a recognition-by-visually-salient-components search strategy can be achieved by using part salience (including protrusion, boundary strength, and relative size); and (4) a coarse-to-fine shape retrieval strategy can be accomplished by introducing the concepts of part relation and visual salience, respectively. To the best of our knowledge, this is the first 3-D shape retrieval scheme that realizes the psychological theories of recognition-by-components [11] and visual salience [51].

Chapter 4 Fragile Watermarking for Authenticating 3-D Polygonal Meshes

Transferring digitized media via the Internet has become very popular in recent years. Content providers who present or sell their products through networks are, however, faced with the copyright protection problem. In order to properly protect the rights of a content owner, it is desirable to develop a robust protection scheme that can prevent digital contents from being stolen or illegally distributed. From a user's point of view, after receiving a piece of digital content, he/she usually needs to verify the integrity of the content. As a result, there should be an authentication mechanism that can be used to perform the verification task. With the rapid advance of watermarking technologies in recent years, many investigators have devoted themselves to conducting research in this fast growing area. According to the objectives that a watermarking technique may achieve, two main-stream digital watermarking categories are: robust watermarking and fragile watermarking. While the former aims to achieve intellectual property protection of digital contents, the latter attempts to authenticate the integrity of digital contents.

There are a great number of existing robust watermarking algorithms designed to protect 3-D graphic models [3,7,9], [18,19], [59,68], [85,87–91,98], [133], [125]. Their common purpose is to provide a robust way to protect target contents when attacks are encountered. The existing fragile watermarking algorithms that are designed to authenticate 3-D graphic models are relatively few. In [34], Fornaro and Sanna proposed a public key approach to authenticating constructive solid geometry (CSG) models. In [60], Kankanhalli *et al.*, proposed the use of content-based signature to authenticate 3-D volume data. In [131], Yeo and Yeung proposed a fragile watermarking algorithm for authenticating 3-D polygonal meshes. They embed a fragile watermark by iteratively perturbing vertex coordinates until a predefined hash function applied to each vertex matches the other predefined hash function applied to that vertex. Since their embedding algorithm relies heavily on an ordered traversal of vertices, it is capable of detecting object cropping. However, the consideration of causality disables it from localization of changes and robustness against vertex reordering. In addition, particular attacks, such as floating-point truncation or quantization, applied to vertex coordinates might increase the false-alarm probability of tampering detection.

In this chapter, we trade off the causality problem in Yeo and Yeung's method for a new fragile watermarking scheme. The proposed scheme can not only achieve localization of malicious modifications in visual inspection, but also is immune to the aforementioned unintentional data processings. In addition, the allowable range for alternating a vertex is explicitly defined so that the new scheme is able to tolerate quantization of vertex coordinates (up to a certain amount). During the process of watermark embedding, a local mesh parameterization approach is employed to perturb the coordinates of invalid vertices while cautiously maintaining the visual appearance of the original model. Since the proposed embedding method is independent of the order of vertices, the hidden watermark is immune to some vertex order-dependent attacks, such as vertex reordering.

The remainder of this chapter is organized as follows. In Section 4.1, Yeo and Yeung's scheme for authenticating 3-D polygonal meshes is briefly reviewed. In Section 4.2, the proposed fragile watermarking method is described in detail. Experiment results are given in Section 4.3. Finally, conclusions are drawn in Section 4.4.

4.1 Yeo and Yeung's Approach and Its Drawbacks

In [131], Yeo and Yeung proposed a novel fragile watermarking algorithm which can be applied to authenticate 3-D polygonal meshes. In Yeo and Yeung's scheme [131], there are three major components, i.e., two predefined hash functions and an embedding process. For a given vertex, the vertex is identified as valid if and only if the values calculated by both hash functions are identical. Otherwise, the vertex is identified as invalid. During the authentication process, invalid vertices are considered as the set of vertices that has been tampered with. On the other hand, valid vertices indicate the set of vertices which has never been modified. In the embedding process, the coordinates of valid vertices are kept unchanged, but those of invalid vertices are iteratively perturbed until each of them becomes valid.

The first step in Yeo and Yeung's approach is to compute location indices. In this step, the first hash function is defined by a conversion function and associated with a given watermark pattern WM. The conversion function is used to convert a vertex coordinate $v = (v_x, v_y, v_z)$ into a location index $L = (L_x, L_y)$. The idea behind the conversion function is to map a three dimensional coordinate onto a two dimensional plane formed by a watermark pattern of dimension $WM\_X\_SIZE \times WM\_Y\_SIZE$. As a result, the location index L is used to point to a particular position in the watermark pattern. Then, the content of that particular position $WM(L)$ (either 0 or 1) is used for the purpose of comparison. Since the conversion function defined in [131] calculates the centroid of the neighboring vertices of a given vertex, the causality problem occurs. Furthermore, the traversal of vertices during the alternation of vertex coordinates must take causality into account so as to avoid error propagation.

The second step in Yeo and Yeung's approach is to compute value indices. In this step, the second hash function is related to a set of look-up tables, i.e., K_1, K_2, and K_3. These look-up tables, which are composed of sequences of bits, are generated and protected by an authentication key. Yeo and Yeung [131] proposed to convert each component of a vertex coordinate into an integer number so as to index it into each of the look-up tables. The content of an indexed location is either 0 or 1. The three binary values derived from the three coordinates $p = (p_1, p_2, p_3)$ are then XOR processed to generate a final binary value. This binary value $K(p)$ is used as one of the components for deciding whether the current vertex is valid or not. If the vertex is not valid, then it is perturbed until it is valid. The amount of change that makes this vertex valid is the watermark embedded.

After establishing the above-mentioned two hash functions, the next step is to perturb the coordinates of all invalid vertices until they become valid. In [131], the authors proposed an iterative procedure which can gradually perturb an invalid vertex until both hash functions are matched. On the one hand, in order to maintain transparency, the embedding procedure must traverse in an orderly manner each vertex during the alteration of vertex coordinates. In addition, the ordering of vertices must be maintained during the watermark extraction process. The benefit of taking the causality into account is for protection against changes of connectivity (in particular cropping). However, the drawback is that their method cannot achieve localization of malicious modifications in visual inspection. In addition, their method cannot tolerate certain incidental modifications, such as quantization of vertex coordinates and vertex reordering. This drawback to some extent limits the power of Yeo and Yeung's method. In this chapter, we shall propose a new scheme that is more powerful than the existing fragile watermarking algorithms.

4.2 The Proposed Fragile Watermarking Method

In this section, we shall propose a new fragile watermarking scheme for authenticating 3-D polygonal meshes. In order to tackle the issues that were not handled by Yeo and Yeung [131], we employ the following concepts: 1) Each hash function can be designed so as to form a binary state space particularly helpful for defining the domain of allowable alternation for a given vertex. Accordingly, the domain of acceptable alternation for a given vertex can be defined as the intersection of the binary state spaces where the values of both hash functions match each other; 2) In order to resolve the causality problem, the conversion function used in the first hash function can be designed to simply perform the mapping from the 3-D space to a 2-D plane without considering the neighboring vertices of a vertex. Based on the above two concepts, we have designed a new scheme, which is shown in Fig. 4.1. With the new authentication scheme, malicious attacks applied to 3-D polygonal meshes can be easily distinguished from certain incidental modifications. In what follows, we shall describe our authentication scheme in more detail.

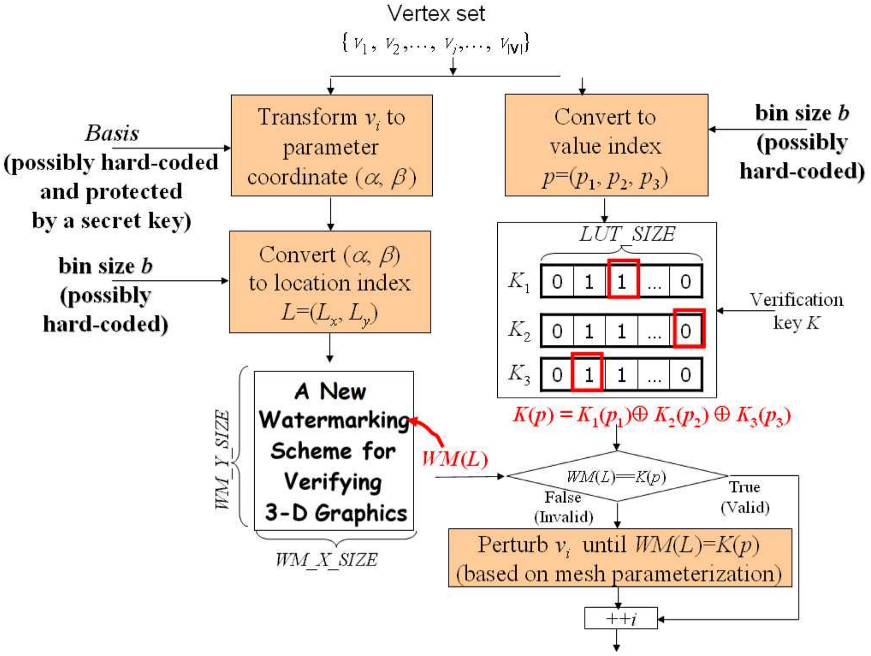

Vertex set
{ v1, v2,…, vi,…, v|V|}
Basis (possibly hard-coded and protected by a secret key)
Transform vi to parameter coordinate (α, β)
bin size b (possibly hard-coded)
Convert (α, β) to location index L=(Lx, Ly)
Convert to value index p=(p1, p2, p3)
bin size b (possibly hard-coded)
LUT_SIZE
K1 0 1 1 … 0
K2 0 1 1 … 0
K3 0 1 1 … 0
Verification key K
K(p) = K1(p1)⊕ K2(p2) ⊕ K3(p3)
WM_Y_SIZE
A New Watermarking Scheme for Verifying 3-D Graphics
WM_X_SIZE
WM(L)
WM(L)==K(p)
False (Invalid)
True (Valid)
Perturb vi until WM(L)=K(p) (based on mesh parameterization)
++i

Figure 4.1: The flowchart of the proposed authentication scheme for 3-D polygonal meshes.

4.2.1 Computing Location Indices

Since the conversion function used in the first hash function (the left hand side of Fig. 4.1) aims to calculate the location index that can be used to locate a particular bit in the watermark pattern, any functions that can transform a 3-D coordinate into a 2-D coordinate can serve this purpose. Therefore, it is possible to use some parameterization schemes to achieve the goal. As mentioned in the previous section, Yeo and Yeung did not use an analytical method to perturb invalid vertices. However, a systematic perturbation strategy is always preferable. Therefore, we propose to adopt the parameterization-based approach to make the vertex perturbation process analytic. For the purpose of clarity, we propose to split the location index computation process into two steps:

Step 1 Given a vertex coordinate v, the specified parameterization $S : \mathbb{R}^3 \rightarrow \mathbb{R}^2$ converts the vertex coordinate into a parameter coordinate. We propose to use so-called cylindrical parameterization [57] to perform the conversion task. The procedure involved in performing cylindrical parameterization is as follows [57]:

Given an oriented 3-D point, it is composed of a 3-D point m and its orientation n. As shown in Fig. 4.2(a), a cylindrical parameterization process[6] can be expressed as

$$S_{m,n}(v) \rightarrow (\alpha, \beta) = (\sqrt{\|v - m\|^2 - (n \cdot (v - m))^2}, n \cdot (v - m)), \quad (4.1)$$

where (α, β) is the coordinate in the parameter domain. The range for each dimension of the parameter domain is $\alpha \in [0, \infty)$ and $\beta \in (-\infty, \infty)$, respectively.

Step 2 Convert the parameter coordinate formed in Step 1 into the so-called bin coordinate, i.e., the location index (L_x, L_y). This conversion can be accomplished by quantizing the parameter domain. In addition, a modulus operator is required to map them onto the dimension of a watermark pattern. In what follows, we shall describe how the parameter domains are quantized. Assume that the size of a 2-dimensional watermark pattern is $WM\_X\_SIZE \times WM\_Y\_SIZE$,

[6] Note that although an oriented point defines 5 degree of freedom (DOF) basis (m, n), the proposed method is not immune to geometrical transformations. This results from the fact that whether a vertex is valid or not is guarded by the two hash functions.

the quantization formula for a cylindrical parameterization domain is as follows:

$$L = (L_x, L_y) = (\left\lfloor \frac{\alpha}{b} \right\rfloor \% WM\_X\_SIZE, \left\lfloor \frac{\beta}{b} \right\rfloor \% WM\_Y\_SIZE), \quad (4.2)$$

where b is the quantization step for ordinary numeric values and % represents a modulus operator.

A very important feature of the above design is that the quantized parameterization domain and the watermark pattern together form a binary state space. Such a state space is helpful for defining a legal domain of alternation for a given vertex. The side-view of the binary state space corresponding to the quantized cylindrical parameterization domain is illustrated in Fig. 4.2(b).

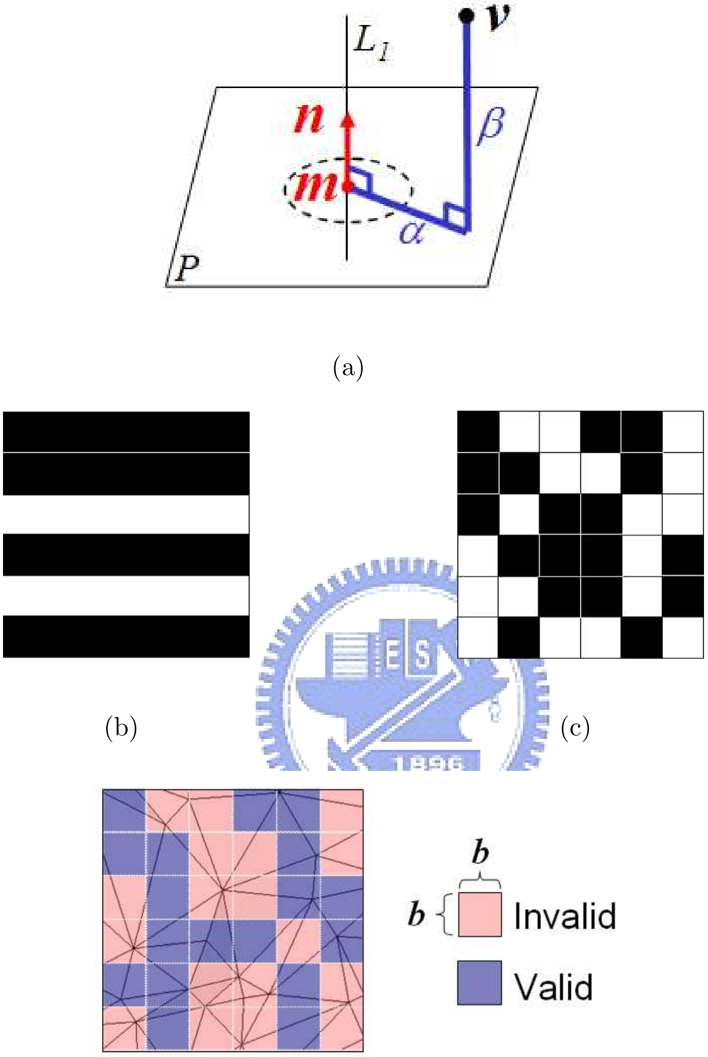
4.2.2 Computing Value Indices

Even though any functions for converting a floating-point number into an integer can be used to calculate value indices, the following conversion function was designed since it is able to form a binary state space. Assuming that the size of each look-up table is LUT_SIZE, the conversion function is formulated as

$$p = (p_1, p_2, p_3) = (\left\lfloor \frac{v_x}{b} \right\rfloor \% LUT\_SIZE, \\ \left\lfloor \frac{v_y}{b} \right\rfloor \% LUT\_SIZE, \left\lfloor \frac{v_z}{b} \right\rfloor \% LUT\_SIZE), \quad (4.3)$$

where b is the same quantization step as used to compute location indices.

The side-view of the binary state space corresponding to the above conversion function is illustrated in Fig. 4.2(c). In addition, Fig. 4.2(d) reveals that



(a)

(b)

(c)

(d)

Figure 4.2: Illustration of the robustness construction: (a) the basis for cylindrical parameterization [57]; (b) the side-view of the binary state space formed by the quantized cylindrical parameterization domain; (c) the side-view of the binary state space formed by the conversion function for computing value indices; (d) the two binary state spaces superimposed on a sidepiece of the cylindrical mesh with irregular connectivity.

the domain of acceptable alternation for a given vertex can be defined as the intersection of the binary state spaces where the values of both hash functions applied to that vertex match each other. More precisely, for a valid vertex the displacement applied to its original coordinates will depend on the value of (α, β) and thus it will make (L_x, L_y) change as well. As long as the displacement for both location and value indices does not vary beyond the aforementioned domain of acceptable alternation, the vertex will be identified as intact by our scheme. As a result, the encoded location and value indices will be robust to a certain extent of quantization.

4.2.3 Watermark Embedding

Since both hash functions have been well-designed to define the domain of acceptable alternation for a given vertex, the embedding procedure can focus on perturbing the coordinates of invalid vertices while maintaining transparency. In the remeshing-related literatures [33], [71], [99], [120], [116], the points over the surface of the polygonal model have frequently been used for resampling the geometry of a model. We, therefore, apply a local mesh parameterization approach proposed in [71] for finding a valid point on the surface of a polygonal mesh. Assume that the polygonal model to be watermarked is a closed and oriented 2-manifold mesh that has been triangulated, our method is as follows: Given an invalid vertex $v \in \mathbb{R}^3$ and its neighboring vertices in the counter-clockwise order

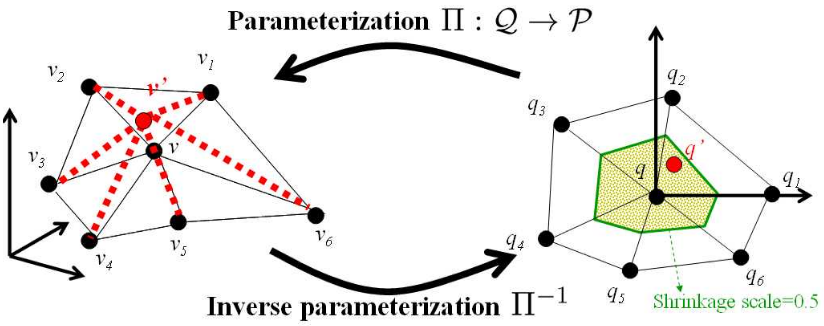

Parameterization $\Pi : \mathcal{Q} \rightarrow \mathcal{P}$
Inverse parameterization Π^{-1}
v_1
v_2
v_3
v_4
v_5
v_6
v
v'
q_1
q_2
q_3
q_4
q_5
q_6
q
q'
Shrinkage scale=0.5

Figure 4.3: The proposed alternation procedure for an invalid vertex.

$v_1, v_2, \ldots, v_{|N(v)|} \in \mathbb{R}^3$, where $|N(v)|$ is the number of v's neighboring vertices, the proposed alternation procedure for an invalid vertex is divided into five steps, which can be explained with the help of Fig. 4.3. The details of the five steps are as follows:

Step 1 Transform the vertex coordinate v into the parameter coordinate $q \in \mathbb{R}^2$ and its neighboring vertices $v_1, v_2, \ldots, v_{|N(v)|}$ to $q_1, q_2, \ldots, q_{|N(v)|} \in \mathbb{R}^2$, respectively, using arc-length parameterization. Let $ang(a, b, c)$ be the angle formed by vectors $\overrightarrow{ba}$ and $\overrightarrow{bc}$. Then, the parameter coordinates are provided with the following properties [33]:

$$\|q_k - q\| = \|v_k - v\|, \tag{4.4}$$

$$ang(q_k, q, q_{k+1}) = 2\pi \cdot ang(v_k, v, v_{k+1})/\theta, \tag{4.5}$$

where

$$\theta = \sum_{k=1}^{|N(v)|} ang(v_k, v, v_{k+1}), v_{|N(v)|+1} = v_1,$$
$$q_{|N(v)|+1} = q_1, \text{ and } k = 1, \ldots, |N(v)|.$$

If we set $q{=}(0,0)$ and $q_1{=}(\|v_1{-}v\|,0)$, the parameter coordinates $q_2, q_3, \ldots, q_{|N(v)|}$ can be easily derived from (4.4) and (4.5). Hence, $q_1, q_2, \ldots, q_{|N(v)|}$ form the boundary vertices of the star-shaped planar polygon $\mathcal{Q}$ with q in its kernel. In addition, $v_1, v_2, \ldots, v_{|N(v)|}$ are the boundary vertices of the polygon $\mathcal{P}$ with one internal vertex v. Let $\sqcup_k$ denote the triangle formed by the parameter coordinates q, q_k, q_{k+1} and $\mathcal{T}_k$ denote the triangle formed by the vertex coordinates v, v_k, v_{k+1} for $k = 1, \ldots, |N(v)|$. Then, the two triangle sets $\{\sqcup_k\}$ and $\{\mathcal{T}_k\}$ form the triangulation of the planar polygon $\mathcal{Q}$ and the polygon $\mathcal{P}$, respectively.

Step 2 Establish the local mesh parameterization $\Pi : \mathcal{Q} \rightarrow \mathcal{P}$ by means of the well-known barycentric mapping. Let $\hat{q}$ denote an arbitrary point inside the planar polygon $\mathcal{Q}$ and $area(a, b, c)$ denote the signed area of the triangle formed by the vertices a, b, c. Then, there exists a unique $t \in \{1, \ldots, |N(v)|\}$ such that the barycentric coordinates of $\hat{q}$ will correspond to the triangle $\sqcup_t$ and have the following forms:

$$\lambda_{t,1} = \frac{area(\hat{q}, q_t, q_{t+1})}{area(q, q_t, q_{t+1})}, \lambda_{t,2} = \frac{area(q, \hat{q}, q_{t+1})}{area(q, q_t, q_{t+1})}, \lambda_{t,3} = \frac{area(q, q_t, \hat{q})}{area(q, q_t, q_{t+1})}. \tag{4.6}$$

The three barycentric coordinate components are all of the same sign. Hence, the corresponding point $\hat{v}$ on the surface of the polygon $\mathcal{P}$ can be represented as a combination of the points v, v_t, v_{t+1} with respect to $\mathcal{T}_t$ as follows:

$$\hat{v} = \lambda_{t,1} v + \lambda_{t,2} v_t + \lambda_{t,3} v_{t+1}. \tag{4.7}$$

Step 3 Define an allowable region for alternating an invalid vertex in the parameter domain. Let the region be a shrunken ring whose origin is the parameter coordinate, q, and let the scale for shrinkage be 0.5. (As shown in Fig. 4.2(d), for some invalid vertices to find a valid state may sacrifice a great deal of the original quality. As a result, the shrunken ring defined here can be regarded as the maximum bound of distortion induced by alternating an invalid vertex. In addition, it can avoid geometrical degeneracies, like triangle flipping, T-joints, etc.)

Step 4 Distribute a set of points $\tilde{\mathbf{q}} = \{\tilde{q}_i \in \mathbb{R}^2 : i = 1, \ldots, r\}$ randomly on the allowable region.[7] Next, find a new parameter coordinate $q' \in \tilde{\mathbf{q}}$ satisfying the condition

$$WM(L(S_{m,n}(\Pi(q')))) = K(p(S_{m,n}(\Pi(q')))), \tag{4.8}$$

where Π is the barycentric mapping derived from (4.6) and (4.7). If there does not exist such a new parameter coordinate, alternation for the current invalid vertex is skipped, and $q' = q$ is assigned.

Step 5 Record the new vertex coordinate $v' = \Pi(q')$.

Note that the set of random points generated in Step 4 can be sorted according to its geometric distance to the parameter coordinate q, in such a way that the new parameter coordinate q' can be chosen not only satisfying (4.8) but also

[7] A random point in a triangle is generated using the method described in [121].

minimizing the distortion. Currently, this feature has not been considered in our implementation since the maximum distortion has been bounded as described in Step 3. As for maximizing the robustness, the domain of acceptable alternation can be mapped onto the parameter domain using the inverse of the parameterization $\Pi^{-1} : \mathcal{P} \rightarrow \mathcal{Q}$. Then, the new parameter coordinate q' that maximizes the robustness can be determined. Since the efficiency of the algorithm would be degraded, this feature has not been implemented in our system. As for the performance of our method, it is a natural outcome of the Step 4 that a certain amount of invalid vertices may remain untouched/invalid. We, therefore, propose some possible solutions to optimize the performance in the following section.

4.2.4 Analysis and Discussion

In this section, we shall conduct a thorough analysis of our authentication scheme for 3-D polygonal meshes. The watermarking parameters that can influence the quality of transparency and robustness are the shrinkage scale and bin size. On the other hand, we also know that the correlation value C can never reach 1. Therefore, we shall examine several crucial issues: 1) how to optimize the performance so that C can be very close to 1; 2) how to balance the competition between transparency and capacity using the shrinkage scale; and 3) how to guarantee the robustness of a hidden watermark. Before discussing the above mentioned issues, we adopt the correlation value used by Yeo and Yeung [131] and formulated it as

$$C = \frac{|\{v : K(p(v)) = WM(L(v))\}|}{|V|}, \tag{4.9}$$

where V is the vertex set of a mesh and $|V|$ is the total number of vertices. Note that the correlation C is the ratio of the number of valid vertices to the total number of vertices (instead of a linear correlation coefficient). In what follows, we shall discuss the aforementioned issues.

First of all, we aim to optimize the performance of our algorithm so that the watermark correlation value C can be very close to 1. In our investigation, there are two possible solutions to optimize the performance. The first solution is to adopt a smaller quantization step, which would increase the possibility of finding a valid state. Such an approach will be a great benefit to the maintenance of transparency. However, the drawback is that the robustness would be sacrificed as well. An alternative solution is to make the spacing between vertices regular while maintaining the shape of a 3-D mesh. In such an approach, the robustness can benefit greatly from the specified quantization step (i.e., the bin size). However, the drawback is that the shape of the mesh would be simplified significantly when the spacing between vertices is increased. Our intention here is to maintain the robustness when encountering certain incidental modifications, such as vertex quantization and noise addition. We, therefore, picked five different models to generate analysis models with different mesh resolutions using a mesh resolution control algorithm described in [56].[8] Furthermore, for each model, we generated five analysis models corresponding to different mesh resolutions. Thirty analysis models and their mesh resolutions are listed in Table 4.1. Fig. 4.4 shows the flat-shaded HIV model and its analysis models corresponding to five different mesh resolutions. In the watermarking process, we fixed the shrinkage scale as 0.5 and the bin size as 2. With varied mesh resolution levels, our fragile watermark was embedded into each model to test the effect of the mesh resolution on the watermark correlation value. In addition, we ran each test five times using different keys and reported the median value. Fig. 4.5(a) shows the effect of different mesh resolutions on the watermark correlation value. Obviously, the curves shown in Fig. 4.5(a) reveal that a polygonal mesh with higher mesh resolution would possess higher capacity for watermarking.

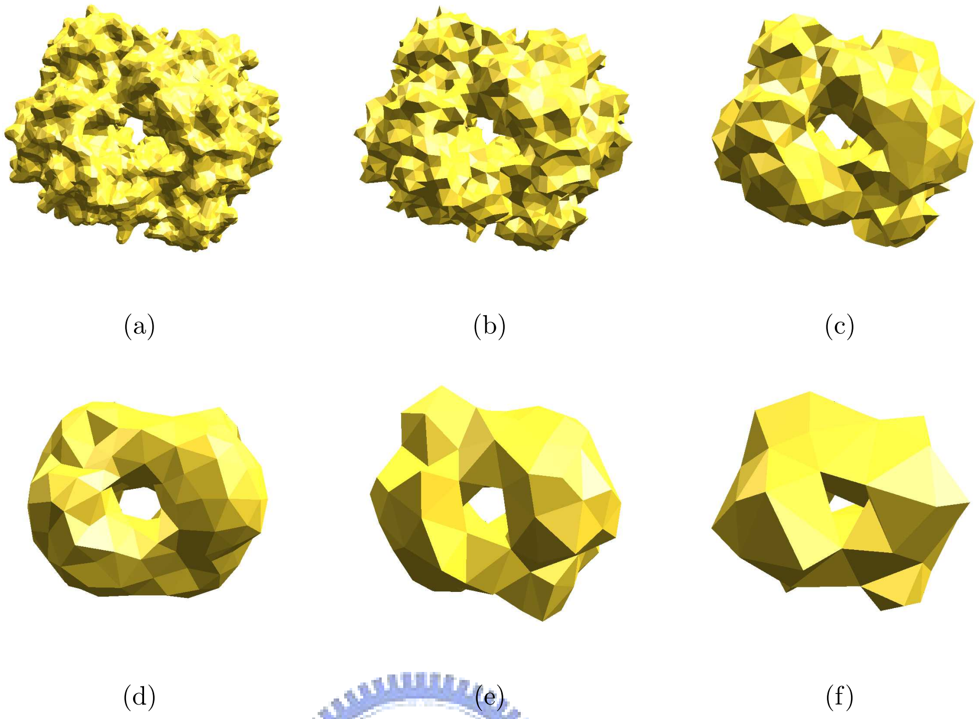
In order to investigate how the shrinkage scale can force a compromise between transparency and capacity, a suitable visual metric was needed to evaluate the difference between the original model M and the watermarked model M'. In [61], Karni and Gotsman proposed the use of Root-Mean-Square measure plus a Laplacian-based visual metric to capture human visual preceptibilities. The RMS metric simply captures the geometric distance between corresponding vertices in both models. On the other hand, the Laplacian-based metric can capture more

[8] In [56], the resolution of a mesh is defined as the median of its edge length histogram. In addition, the edge length spread is defined as the half-width (upper quartile minus lower quartile) of the histogram. The goal of the mesh resolution algorithm is to adjust the resolution of the original mesh to a desired resolution while minimizing the edge length spread of the histogram.

Table 4.1: A list of thirty triangulated meshes used in the analysis.

Model	Number of vertices/faces	Mesh resolution
spock	16386/32768	1.974926
spock-lv1	11543/23082	2.828352
spock-lv2	3604/7204	5.127024
spock-lv3	1819/3634	7.390215
spock-lv4	878/1752	10.350958
spock-lv5	426/848	14.553292
skull	20002/40000	1.524550
skull-lv1	9559/19114	2.876386
skull-lv2	3063/6122	5.138119
skull-lv3	1418/2832	7.652043
skull-lv4	747/1490	10.350958
skull-lv5	365/726	14.622301
holes3	5884/11776	3.581405
holes3-lv1	10620/21248	2.871731
holes3-lv2	3429/6866	5.061736
holes3-lv3	1555/3118	7.652731
holes3-lv4	847/1702	10.258323
holes3-lv5	410/828	14.520646
HIV	9988/20000	1.493200
HIV-lv1	2691/5398	2.915812
HIV-lv2	683/1372	5.233385
HIV-lv3	291/594	7.255211
HIV-lv4	135/270	10.562971
HIV-lv5	62/124	14.92876
isis	46912/93820	1.217783
isis-lv1	8727/17450	2.879494
isis-lv2	2765/5526	5.157736
isis-lv3	1347/2690	7.480489
isis-lv4	658/1312	10.380298
isis-lv5	329/654	14.454476

subtle visual properties (such as smoothness) with respect to both the topology and geometry. The geometric Laplacian operator applied to a vertex v_i is defined


(a) (b) (c)

(d) (e) (f)

Figure 4.4: Analysis models for the HIV protease surface model: (a) original HIV model; (b) HIV-lv1 model; (c) HIV-lv2 model; (d) HIV-lv3 model; (e) HIV-lv4 model; (f) HIV-lv5 model.

as

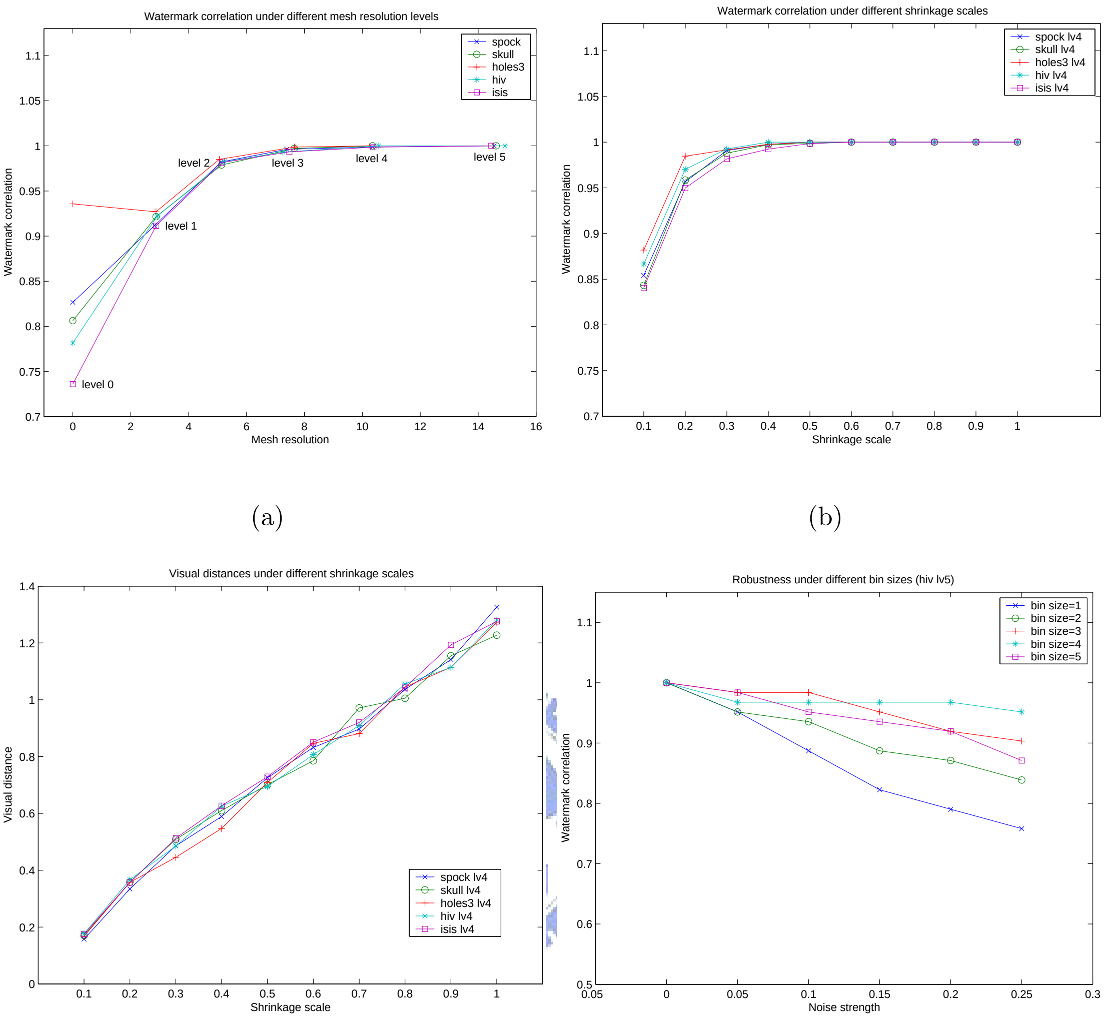
$$GL(v_i) = v_i - \frac{\sum_{j \in n(i)} l_{ij}^{-1} v_j}{\sum_{j \in n(i)} l_{ij}^{-1}}, \tag{4.10}$$

where $n(i)$ is the set of indices of v_i's neighboring vertices, and l_{ij} is the geometric distance between vertices i and j. Hence, the visual difference between the original model M and the watermarked model M' can be expressed as

$$diff(M, M') = \frac{1}{2|V|}\left(\sum_{i=1}^{|V|} \|v_i - v_i'\| + \sum_{i=1}^{|V|} \|GL(v_i) - GL(v_i')\|\right). \tag{4.11}$$

In the mesh-based watermarking literature [19], the above mentioned visual metric has been used to capture the geometric distortion between two models. We, therefore, adopted this visual metric to measure the transparency. In this analysis, we picked five models that were at the fourth resolution. We chose the bin size and the shrinkage scale as 2 and 0.5, respectively. With various shrinkage scales, our fragile watermark was embedded into each model for transparency and capacity tests. In the same way, we ran each test five times using different keys and reported the median value. Figs. 4.5(b)-4.5(c) show the effects of different shrinkage scales on the watermark correlation value and PSNR value, respectively. From Figs. 4.5(b)-4.5(c), it is clear that the best choice of shrinkage scale is 0.5.

In order to demonstrate how robust our watermark is, we attacked the embedded watermark by means of randomization of vertex coordinates. To simulate such attacks, randomization of vertex coordinates was controlled by means of the noise strength, which is defined as the ratio of the largest displacement to the longest edge of the object's bounding box. In this analysis, we picked five models with the largest resolution level from the set of analysis models and fixed the shrinkage scale at 0.5. With various bin sizes, our watermark was embedded into each model and then attacked using different noise strengths in robustness tests. In the same way, we ran each test five times using different keys and reported the median value. Fig. 4.5(d) shows the results of robustness tests using different bin sizes for the HIV-lv5 model. From these plots, it can be seen that a larger bin size can provide a hidden watermark with higher robustness. However, the drawback is that the false-alarm rate is increased as well.



(a)

(b)

(c)

(d)

Figure 4.5: (a) Effect of the mesh resolution on the watermark correlation value. Note that the mesh resolution of “0” indicates that the original models were not influenced by the mesh resolution control algorithm; (b) effect of the shrinkage scale on the watermark correlation value; (c) effect of the shrinkage scale on the transparency of our fragile watermark; (d) robustness under different bin sizes for the HIV-lv5 model.

4.3 Experiment Results

A series of experiments were conducted to test the performance of the proposed fragile watermarking method. We shall start with parameter selection and then report quantitatively some experiment results. In addition, we shall present a set of visualization results that can demonstrate the power of the proposed method in distinguishing malicious attacks from incidental modifications.

4.3.1 Selecting Appropriate Parameters

We have reported in Section 4.2 that several parameters were needed during watermark embedding and detection. These parameters included a binary watermark pattern, a set of look-up tables, a basis for parameterization, and the degree of quantization. All of the parameters used in our experiments were set as follows. A binary watermark pattern with a size of 512×512 (as indicated in Fig. 4.6) was used in our experiments. That means, $WM\_X\_SIZE = WM\_Y\_SIZE = 512$. In addition, a set of look-up tables were generated and protected by one authentication key. The size of each table was 256. Therefore, $LUT\_SIZE = 256$. As to the basis for parameterization, since the 3-D vertex space is periodically aggregated into binary state spaces, its selection is not crucial to the proposed method. Therefore, we fixed the basis as $m(0, 0, 0)$ and $n(1, 0, 0)$ in the experiments. As for appropriate quantization steps, we assigned the ordinary numeric value, $b = 0.2$, in all the experiments such that the performance of our method

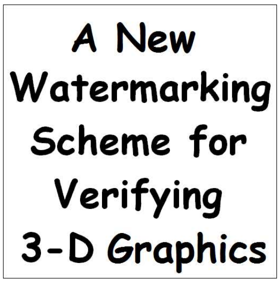


Figure 4.6: The binary watermark pattern used in our experiments.

is close to optimal (i.e., $C \cong 1$). The selection of $b = 0.2$ was based on the experiments gained from conducting quite a number of experiments. However, since the selection is an ill-posed problem, it is hard to systematically determine a right value that can fit in all cases.

One thing to be noted is that the basis (m, n) and the quantization step b together can possibly be hard-coded into the algorithm so that detecting a watermark for the purpose of authentication can be realized as oblivious detection. However, the drawback is that the robustness (i.e., the domain of acceptable alternation) certainly varies with applications. These are the restriction that are associated with an LUT/secret key approach in general.

4.3.2 Experiment Results of Authentication

The data set used in our experiments was a set of triangulated and closed meshes, listed in Table 4.2. Each of them was watermarked using our fragile watermarking method presented in Section 4.2. The last column in Table 4.2 shows the watermark correlation values for the five different models. The five test models were

Table 4.2: A list of five triangulated meshes used in our experiments and their watermark correlation values detected using the proposed method.

Model	**Number of vertices/faces**	**Correlation value**
dolphins	855/1692	1
spock	16386/32768	1
mannequin	711/1418	1
holes3	5884/11776	1
HIV	9988/20000	1

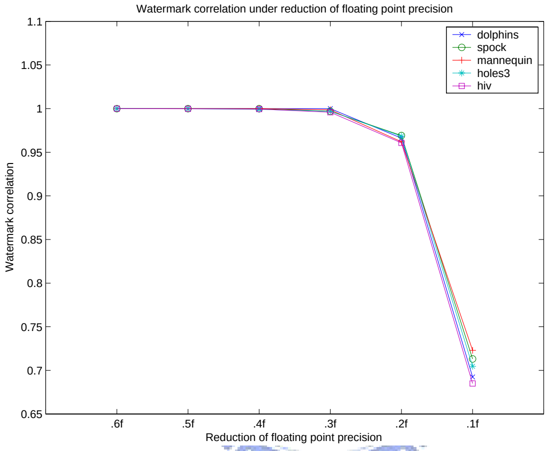


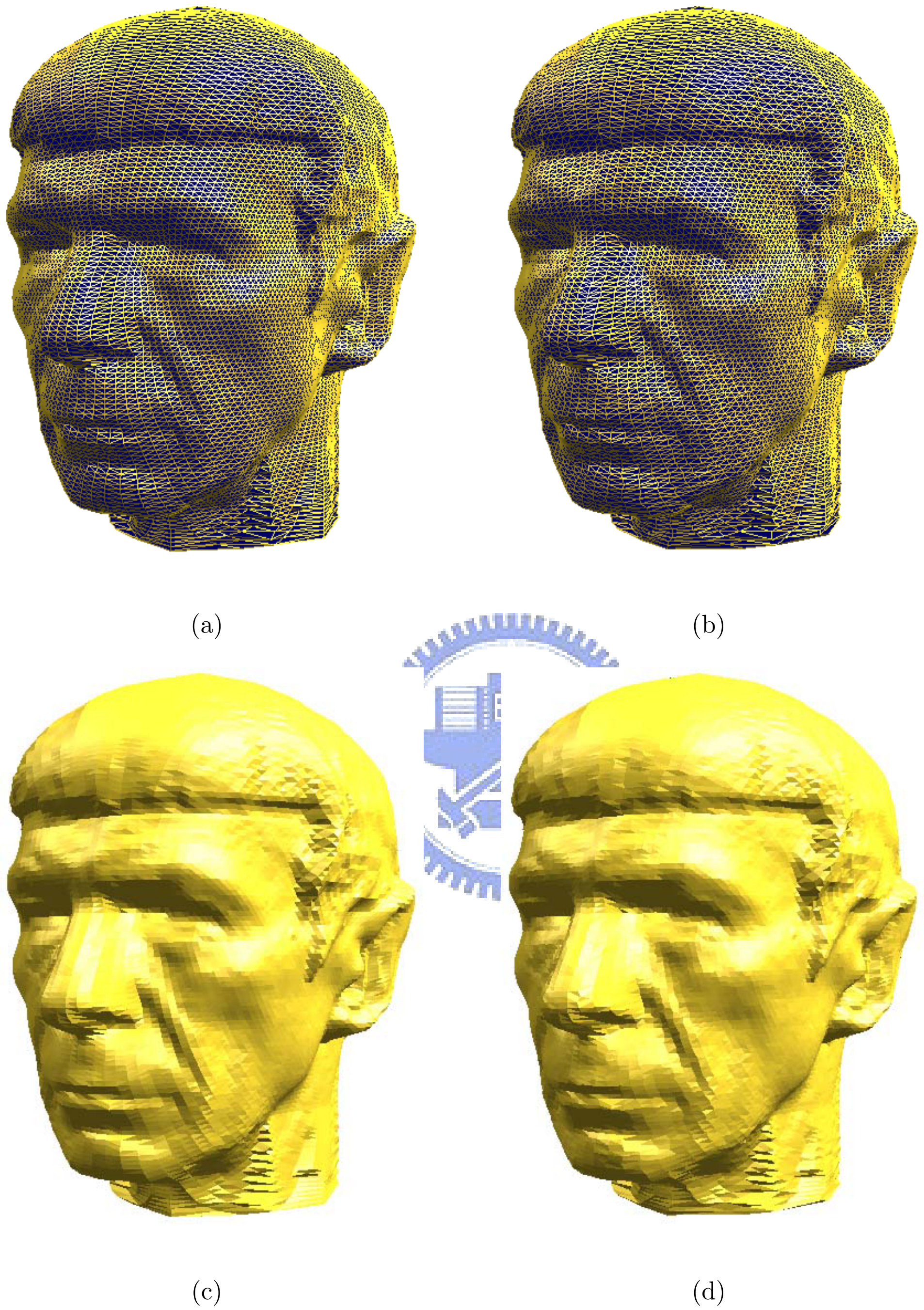
Figure 4.7: Five test models were watermarked and tested to evaluate the robustness against reduction of floating-point precision.

watermarked and tested to evaluate the robustness against reduction of floating-point precision. The results of this experiment are shown in Fig. 4.7, where the precision of a floating-point number is specified by a nonnegative decimal integer preceded by a period (.) and succeeded by a character f. It is clearly shown in Fig. 4.7 that the proposed method is very robust against vertex quantization down to 3 decimal digits. Note that for authentication applications one has to rely on visual inspection since the correlation coefficient does not signal. For instance, for meshes with a large number of vertices, only modifying a small region does not affect the correlation value substantially. In addition, finding a threshold that is suitable for all kinds of meshes and resolutions is very difficult. In what follows, therefore, we shall show how to visualize the authentication results.

4.3.3 Visualization of Authentication Results

Visualization is a good way to “see” whether the proposed watermarking method is valid or not. Fig. 4.8 shows that the original and the watermarked Spock models were rendered as either wireframe or flat-shaded models, respectively. It can be seen that the watermarked model maintained high correlation with the original model, whether in a wireframe format or in a flat-shaded format.

The results of experiments on detecting malicious attacks from some incidental modifications are shown in Figs. 4.9-4.10. Fig. 4.9(a) shows that the watermarked Spock model was tampered with by stretching out Spock’s nose. In addition, the quantization down to 2 decimal digits was applied to the vertex coordinates of the watermarked Spock model that has been tampered with. Fig. 4.9(b) shows some detected potentially modified regions before the closing operator was applied. Note that approximately 50 percent of vertices on Spock’s nose were identified as invalid vertices, as shown in Fig. 4.9(b). Therefore, in order to amplify the effect of the authentication results, the morphological operators described in [103] were adopted so that the parts being tampered with in a model



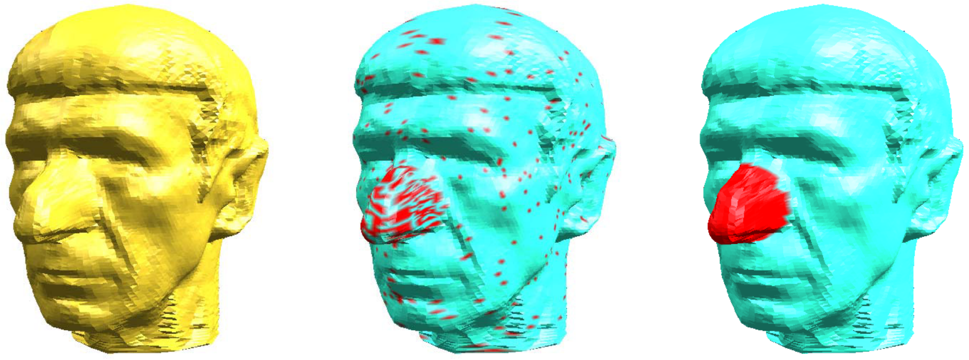
(a)

(b)

(c)

(d)

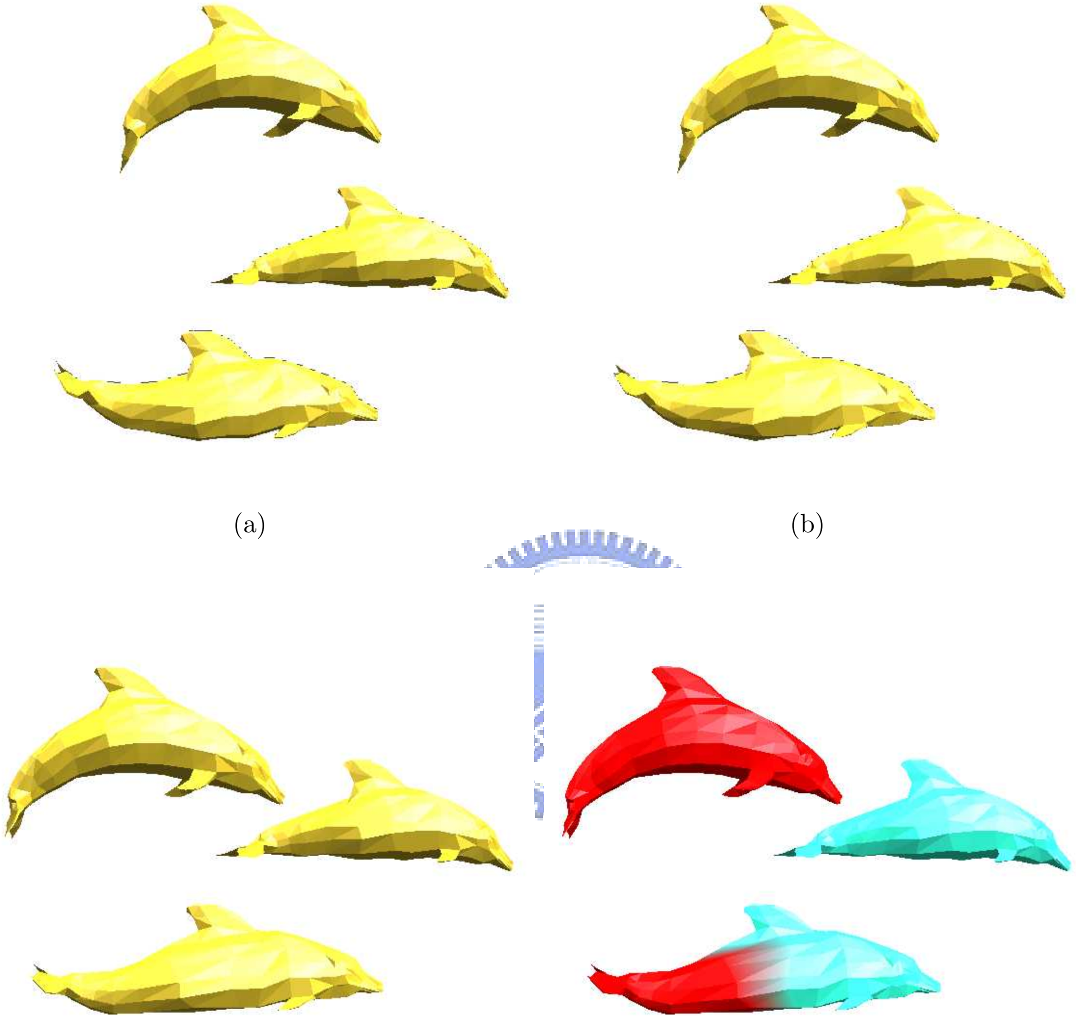
Figure 4.8: Visualization of the transparency test: (a) the original Spock model rendered in a wireframe format; (b) the watermarked Spock model rendered in a wireframe format; (c) the original Spock model rendered in a flat-shaded form; (d) the watermarked Spock model rendered in a flat-shaded form.



(a) (b) (c)

Figure 4.9: Region-based tampering detection: (a) the watermarked Spock model tampered with by stretching out its nose, which was followed by applying the quantization (down to 2 decimal digits) to the vertex coordinates.; (b) the detected potentially modified regions (before morphological operators were applied); (c) the detected modified regions after the morphological operators were applied.

could be detected and highlighted. Fig. 4.9(c) shows the authentication results of Fig. 4.9(b) after some morphological operations were applied. Fig. 4.10 shows another example of malicious tampering involving vertex quantization, which could possibly occur in the real world. In this case, it is not obvious that the two dolphins were tampered with. Nevertheless, the proposed method still succeeded in malicious tampering detection. As shown in Fig. 4.10(d), among the two dolphins that were tampered with, one was translated, and the other one stretched out. Both attacks were detected and highlighted.



(a)

(b)

(c)

(d)

Figure 4.10: Detection of malicious attack involving the incidental modification (such as quantization of vertex coordinates): (a) the original dolphins model; (b) the watermarked dolphins model; (c) a slightly modified dolphins model; (d) two out of the three dolphins have been tampered with. The maliciously modified dolphins were effectively detected.

4.4 Concluding Remarks

A new fragile watermarking scheme which can be applied to authenticate 3-D polygonal meshes has been presented in this chapter. Watermarks are embedded using a local mesh parameterization technique and can be blindly extracted for authentication applications. The proposed scheme has three remarkable features: 1) the domain of allowable alternation for a vertex is explicitly defined by two well-designed hash functions; 2) region-based tampering detection is achieved by a vertex-order-independent embedding process; 3) fragile watermarking is achieved for localization of malicious modifications and tolerance of certain incidental manipulations (such as quantization of vertex coordinates and vertex reordering). To the best of our knowledge, this is the first 3-D mesh authentication scheme that can detect malicious attacks involving certain incidental modifications.

Chapter 5   Conclusions and Future Work

In this dissertation, we have presented three mesh processing techniques for different 3-D graphics-related applications. The three mesh processing methods include the visual salience-guided mesh decomposition (Chapter 2), the cognitive psychology-based approach for 3-D shape retrieval (Chapter 3), and the fragile watermarking method for authenticating 3-D polygonal meshes (Chapter 4).

In Chapter 2, we analyzed and pointed out that the theory of part salience proposed by Hoffman and Singh [51] can be converted into computational processes for extracting significant components from 3-D meshes. More specifically, the protrusion and boundary strength are modeled as the degree of center on the surface and the total-area-of-border change, respectively. These visually salient features are incorporated into the mesh decomposition process based on two rules. They are (1) the protrusion degree characterized over the entire surface can be used as a guide to choose the salient representatives of the parts and (2) the boundary strength characterized over the entire surface can be used as a guide to find the locale of a part's boundary. Since the features used to guide the decomposition process are closely related to Hoffman and Singh's theory of visual salience, the proposed decomposition algorithm can not only appropriately mimic the function of a human visual system, but also efficiently decompose a 3-D mesh into parts. Moreover, the proposed decomposition method is robust against randomization of vertex coordinates. To the best of our knowledge, this is the first 3-D mesh decomposition scheme that not only identifies the part's boundaries defined by the minima rule, but also associates the part with its visual salience.

In Chapter 3, we incorporate a set of cognitive psychology-based principles into the design of 3-D shape analysis and retrieval algorithms. In order to realize the conceptual rule of "recognition-by-components," the proposed visual salience-guided mesh decomposition is adopted to decompose a 3-D mesh-based shape into parts. Next, the decomposed components are individually analyzed and quantified according to the psychology theory of visual salience [51]. Using the above concept, one can label a 3-D shape and then decide on which components on the 3-D shape are the most salient ones. In order to represent the geometrical relations among the parts of a 3-D shape, we propose the use of spherical parameterization to map the labeled 3-D mesh onto a unit sphere. In this way, a set of spherical domain-based shape descriptors, which encodes a part's relation and salience, can be constructed such that comparing 3-D shapes can be done within a normalized sphere. Moreover, our system provides a coarse-to-fine search scheme: (1) a recognition-by-components strategy for coarse search and (2) a recognition-by-visually-salient-components strategy for fine search. More precisely, our relation-based shape descriptor implements the recognition-by-components strategy to perform a coarse search. On the other hand, the visual salience-based shape descriptor realizes the recognition-by-visually-salient-components strategy to refine the results (up to the best m objects) retrieved in the coarse search stage.

Finally, in Chapter 4, we propose a fragile watermarking method for authenticating 3-D polygonal meshes. The proposed authentication scheme can tolerate unintentional modifications, such as vertex re-ordering and floating-point truncation. The robustness comes from the principle that the two hash functions can be designed to form binary state spaces particularly helpful for defining the robustness. Another benefit of our scheme is that region-based tampering detection is achieved by our fragile watermarking method. To the best of our knowledge, this is the first 3-D mesh authentication scheme that can detect malicious attacks involving incidental modifications.

In future work, we shall establish a large 3-D model database for performance evaluation and present more quantitative results for comparison with the existing 3-D shape retrieval systems. Moreover, use of combined features to perform the retrieval task will be the main subject for our future work.

Bibliography

[1] M. Alexa, “Merging Polyhedral Shapes with Scattered Features,” *The Visual Computer*, Vol. 16, No. 1, pp. 26-37, 2000.

[2] Alta Vista: Altavista Search Engine (http://www.altavista.com/), 1995.

[3] N. Aspert, E. Drelie, Y. Maret, and T. Ebrahimi, “Steganography for Three-Dimensional Polygonal Meshes,” in *Proc. SPIE*, Vol. 4970, 2002.

[4] G. Barequet and M. Sharir, “Filling Gaps in the Boundary of a Polyhedron,” *Computer Aided Geometric Design*, Vol. 12, No. 2, pp. 207-229, Mar. 1995.

[5] O. Benedens, “Two High Capacity Methods for Embedding Public Watermarks into 3-D Polygonal Models,” in *Proc. Multimedia and Security-Workshop at ACM Multimedia*, Orlando, Florida, 1999, pp. 95-99.

[6] O. Benedens, “Geometry-Based Watermarking of 3-D Models,” *IEEE Computer Graphics and Applications*, Vol. 19, pp. 46-45, 1999.

[7] O. Benedens, “Affine Invariant Watermarks for 3-D Polygonal and NURBS Based Models,” in *Proc. Information Security Workshop*, Wollongon, Australia, 2000, pp. 20-21.

[8] O. Benedens and C. Busch, “Towards Blind Detection of Robust Watermarks in Polygonal Models,” in *Proc. EUROGRAPHICS*, Vol. 19, Amsterdam, Netherlands, 2000, pp. C199-C208.

[9] O. Benedens, “Robust Watermarking and Affine Registration of 3-D Meshes,” in *Proc. Information Hiding*, Noordwijkerhout, The Netherlands, 2002, pp. 177-195.

[10] P. J. Besl and N. D. McKay, “A Method for Registration of 3-D Shapes,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 13, No. 3, pp. 209-216, Mar. 1991.

[11] I. Biederman, “Recognition-By-Components: A Theory of Human Image Understanding,” *Psychological Review*, Vol. 94, pp. 115-147, 1987.

[12] H. Biermann, A. Levin, and D. Zorin, "Piecewise Smooth Subdivision Surfaces with Normal Control," in *Proc. ACM SIGGRAPH*, New Orleans, Louisiana, 2000, pp. 113-120.

[13] H. Biermann, I. Martin, F. Bernardini, and D. Zorin, "Cut-and-Paste Editing of Multiresolution Surfaces," in *Proc. ACM SIGGRAPH*, San Antonio, Texas, 2002, pp. 312-321.

[14] S. Bischoff and L. Kobbelt, "Ellipsoid Decomposition of 3-D Models," in *Proc. 1st Int. Symposium on 3-D Data Processing, Visualization, and Transmission*, Padova, Italy, 2002, pp. 480-488.

[15] S. Bischoff and L. Kobbelt, "Streaming 3-D Geometry Data Over Lossy Communication Channels," in *Proc. IEEE Int. Conf. Multimedia and Expo*, Lausanne, Switzerland, 2002, pp. 361-364.

[16] S. Brin and L. Page, "The Anatomy of A Large-scale Hypertextual Web Search Engine," *Computer Networks and ISDN Systems*, Vol. 30, pp. 107-117, 1998.

[17] P. J. Burt and E. H. Adelson, "Laplacian Pyramid as A Compact Image Code," *IEEE Trans. Communications*, Vol. 31, No. 4, pp. 532-540, 1983.

[18] F. Cayre and B. Macq, "Data Hiding on 3-D Triangle Meshes," *IEEE. Trans. Signal Processing*, Vol. 51, No. 4, pp. 939-949, 2003.

[19] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, and H. Maître, "Application of Spectral Decomposition to Compression and Watermarking of 3-D Triangle Mesh Geometry," *Signal Processing: Image Communication*, Vol. 18, pp. 309-319, 2003.

[20] L.-F. Chen, H.-Y. Mark. Liao, and J.-C. Lin, "Wavelet-Based Optical Flow Estimation," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 12, No. 1, pp. 1-12, 2002.

[21] D.-Y Chen, X.-P. Tian, Y.-T. Shen and M. Ouhyoung, "On Visual Similarity Based 3D Model Retrieval ," *Computer Graphics Forum (EUROGRAPHICS'03)*, Vol. 22, No. 3, pp. 223-232, Sept. 2003 (http://3d.csie.ntu.edu.tw/∼dynamic/).

[22] J.-H. Chuang, C.-H. Tsai, and M.-C. Ko, "Skeletonization of Three-Dimensional Object Using Generalized Potential Field," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 11, pp. 1241-1251, 2000.

[23] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, McGraw-Hill, 2001.

[24] J. Corney, H. Rea, D. Clark, J. Pritchard, M. Breaks, and R. MacLeod, "Coarse Filters for Shape Matching," *IEEE Computer Graphics and Applications*, Vol. 22, No. 3, pp. 65-74, May/June 2002

[25] I. Daubechies, I. Guskov, P. Schroder, and W. Sweldens, "Wavelets on Irregular Point Sets," *Phil. Trans. R. Soc. Lon. A.*, Vol. 357, No. 1760, pp. 2397-2413, 1999. (Also as part of book entitled *Wavelets: The Key to Intermittent Information,* edited by B.W. Silverman and J.C. Vassilicos, to be released in September 2000.)

[26] M. Desbrun, M. Meyer, P. Schröder, and A. Barr, "Implicit Fairing of Irregular Meshes Using Diffusion and Curvature Flow," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1999, pp. 317-324.

[27] T. K. Dey, H. Edelsbrunner, S. Guha and D. V. Nekhayev, "Topology Preserving Edge Contraction," *Publications De L'Institut Mathématique(Beograd) (N.S.),* Vol. 66, No. 80, pp. 23-45, 1999.

[28] C. Dorai and A. K. Jain, "COSMOS–A Representation Scheme for 3-D Free-Form Objects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 10, pp. 1115-1130, 1997.

[29] M. Eck, T. DeRose, T. Duchamp, H. Hoppe, M. Lounsbery, W. Stuetzle, "Multiresolution Analysis of Arbitrary Meshes," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1995, pp. 173-182

[30] M. Elad, A. Tal, and S. Ar, "Content Based Retrieval of VRML Objects: An Iterative and Interactive Approach," *EG Multimedia*, pp. 97-108, Sept. 2001.

[31] Fast Spherical Harmonic Transforms: SpharmonicKit 2.5 (http://www.cs.dartmouth.edu/~geelong/sphere/), 1998.

[32] FindSounds: Sound File Search Engine (http://www.findsounds.com/), 1998.

[33] M. S. Floater, "Parameterization and Smooth Approximation of Surface Triangulations," *Computer Aided Geometric Design*, Vol. 14, No. 3, pp. 231-250, 1997.

[34] C. Fornaro and A. Sanna, "Public Key Watermarking for Authentication of CSG Models," *Computer-Aided Design*, Vol. 32, No. 12, pp. 727-735, 2000.

[35] K. Fujiwara, "Eigenvalues of Laplacians on A Closed Riemannian Manifold and Its Nets," in *Proc. of AMS.* Vol. 123, pp. 2585-2594, 1995.

[36] T. Funkhouser, P. Min, M. Kazhdan, J. Chen, A. Halderman, D. Dobkin, and D. Jacobs, "A Search Engine for 3-D Models," *ACM Trans. on Graphics*, Vol. 22, No. 1, pp. 83-105, Jan. 2003.

[37] T. Funkhouser, M. Kazhdan, P. Shilane, P. Min, W. Kiefer, A. Tal, S. Rusinkiewicz, and D. Dobkin, "Modeling by Example," *ACM Trans. Graphics*, Vol. 23, No. 3, pp. 652-663, 2004 (Special issue: *Proc. ACM SIGGRAPH* 2004).

[38] T. Funkhouser and M. Kazhdan, *Course 15: Shape-Based Retrieval and Analysis of 3-D Models,* ACM SIGGRAPH Course Note, 2004.

[39] M. Garland and P. S. Heckbert, "Surface Simplification Using Quadric Error Metrics," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1997, pp. 209-216.

[40] M. Garland, A. Willmott, and P. S. Heckbert, "Hierarchical Face Clustering on Polygonal Surfaces," in *Proc. ACM Symposium on Interactive 3-D Graphics*, 2001, pp. 49-58.

[41] A. Goldberg and R. Tarjan, "A New Approach to the Maximum Flow Problem," *Journal of the ACM*, Vol. 35, No. 1, pp. 921-940, 1988

[42] Google Image Search (http://www.google.com/images).

[43] E. Gotsman, X. Gu, and A. Sheffer, "Fundamental of Spherical Parameterization," *ACM Trans. on Graphics*, Vol. 22, No. 3, pp. 358-363, Aug. 2003 (Special issue: *Proc. ACM SIGGRAPH* 2003).

[44] A. Guéziec, G. Taubin, F. Lazarus, and B. Horn, "Cutting and Stitching: Converting Sets of Polygons to Manifold Surfaces," *IEEE Trans. Visualization and Computer Graphics*, Vol. 7, No. 2, pp. 136-151, 2001.

[45] V. Guillemin and A. Pollack, *Differential Topology*, Englewood Cliffs, N. J. : Prentic Hall, 1974.

[46] I. Guskov, W. Sweldens, and P. Schröder, "Multiresolution Signal Processing for Meshes," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1999, pp. 325-334.

[47] I. Goskov, K. Vidimče, W. Sweldens, and P. Schröder, "Normal Meshes," in *Proc. ACM SIGGRAPH*, New Orleans, Louisiana, 2000, pp. 95-102.

[48] M. Hebert, K. Ikeuchi, and H. Delingette, "A Spherical Representation for Recognition of Free-Form Surfaces," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17, No. 7, pp. 681-690, 1995.

[49] M. Hilaga, Y. Shinagawa, T. Kohmura, and T. L. Kunii, "Topology Matching for Fully Automatic Similarity Estimation of 3-D Shapes," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 2001, pp. 203-212.

[50] D. D. Hoffman, "Parts of Recognition," *Cognition*, Vol. 18, pp. 65-96, 1984.

[51] D. D. Hoffman and M. Singh, "Salience of Visual Parts," *Cognition*, Vol. 63, pp. 29-78, 1997.

[52] H. Hoppe, "Progressive Meshes," in *Proc. ACM SIGGRAPH*, New Orleans, Louisiana, 1996, pp. 99-108.

[53] B. Horn, "Extended Gaussian Images," *Proceedings of the IEEE*, Vol. 72, pp. 1656-1678, 1987.

[54] A. Hubeli and M. Gross, "Multiresolution Feature Extraction from Unstructured Meshes," in *Proc. IEEE Visualization*, San Diego, CA, 2001, pp. 287-294.

[55] T. Igarashi, S. Matsuoka, H. Tanaka, "Teddy: A Sketching Interface for 3-D Freeform Design," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1999, pp.409-416.

[56] A. Johnson and M. Hebert, "Control of Polygonal Mesh Resolution for 3-D Computer Vision," *Graphical Models and Image Processing*, Vol. 60, pp. 261-285, 1998.

[57] A. Johnson and M. Hebert, "Using Spin-Images for Efficient Multiple Model Recognition in Cluttered 3-D Scenes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, pp. 433-449, 1999.

[58] T. Ju, "Robust Repair of Polygonal Models," *ACM Trans. on Graphics*, Vol. 23, No. 3, pp. 888-895, Aug. 2004 (Special Issue: *Proc. ACM SIGGRAPH* 1004).

[59] S. Kanai, H. Date, and T. Kishinami, "Digital Watermarking for 3-D Polygons Using Multiresolution Wavelet Decomposition," in *Proc. Sixth IFIP WG 5.2 GEO-6*, Tokyo, Japan, 1998, pp. 296-307.

[60] M. S. Kankanhalli, E.-C. Chang, X. Guan, Z. Huang, and Y. Wu, "Authentication of Volume Data Using Wavelet-Based Foveation," in *Proc. the Sixth Eurographics Workshop on Multimedia*, Manchester, UK, 2001.

[61] Z. Karni and C. Gotsman, "Spectral Compression of Mesh Geometry," in *Proc. ACM SIGGRAPH*, New Orleans, Louisiana, 2000, pp. 279-286.

[62] S. Katz and A. Tal, "Hierarchical Mesh Decomposition Using Fuzzy Clustering and Cuts," *ACM Trans. Graphics*, Vol. 22, No. 3, July 2003 (Special issue: *Proc. ACM SIGGRAPH* 2003).

[63] S. Katz, G. Leifman, and A. Tal, "Mesh Segmentation Using Feature Point and Core Extraction," in *Visual Computer*, Vol. 21, No. 8-10, pp. 865-875, 2005.

[64] M. Kazhdan, T. Funkhouser and S. Rusinkiewicz, "Rotation Invariant Spherical Harmonic Representation of 3-D Shape Descriptor," *Symposium on Geometry Processing*, pp. 167-175, 2003.

[65] M. Kazhdan, B. Chazelle, D. Dobkin, T. Funkhouser, and S. Rusinkiewicz, "A Reflective Symmetry Descriptor for 3-D Models," *Algorithmica: Special Issue*, Vol. 38, No. 2, pp. 201-225, Nov. 2003.

[66] M. Kazhdan, T. Funkhouser, and S. Rusinkiewicz, "Shape Matching and Anisotropy," *ACM Trans. on Graphics*, Vol. 23, No. 3, pp. 623-629, Aug. 2004 (Special issue: *Proc. ACM SIGGRAPH*).

[67] A. Khodakovsky, P. Schroder, and W. Sweldens, "Progressive Geometry Compression," in Proc. ACM SIGGRAPH, New Orleans, Louisiana, 2000, pp. 271-278.

[68] B. Koh and T. Chen, "Progressive Browsing of 3-D Models," in *Proc. IEEE Signal Processing Society Workshop on Multimedia Signal Processing*, Copenhagen, Denmark, 1999, pp. 71-76.

[69] I. Kolonias, D. Tzovaras; S. Malassiotis, M.G. Strintzis, "Fast Content-based Search of VRML Models Based on Shape Descriptors," *IEEE Trans. Multimedia*, Vol. 7, No. 1, pp. 114-126, Feb. 2005.

[70] F. Lazarus and A. Veroust, "Level Set Diagrams of Polyhedral Objects," in *Proc. ACM Symposium on Solid Modeling and Applications*, 1999, pp. 130-140.

[71] A. W. F. Lee, W. Sweldens, P. Schröder, L. Cowsar, and D. Dobkin, "MAPS: Multiresolution Adaptive Parameterization of Surfaces," in *Proc. ACM SIGGRAPH*, Orlando, Florida, 1998, pp. 95-104.

[72] B. Lévy, S. Petitjean, N. Ray, and J. Maillot, "Least Square Conformal Maps for Automatic Texture Atlas Generation," in *Proc. ACM SIGGRAPH*, San Antonio, TX, 2002, pp. 362-371.

[73] J. Li and C.-C. Jay Kuo, "Progressive Coding of 3-D Graphic Models," *Proceedings of the IEEE*, Vol. 86, No. 6, pp. 1052-1063, 1998.

[74] X. Li, T. W. Woon, T. S. Tan, and Z. Huang, "Decomposing Polygon Meshes for Interactive Applications," in *Proc. ACM Symposium on Interactive 3-D Graphics*, 2001, pp. 35-42.

[75] J. M. Lien and N. M. Amato, "Approximate Convex Decomposition," in *Proc. ACM Symposium on Computational Geometry*, Brooklyn, New York, Jun, 2004, pp. 457-458.

[76] P. Liepa, "Filling Holes in Meshes," *Eurographics Symposium on Geometry Processing*, pp. 200-205, 2003.

[77] H.-Y. Sean Lin, H.-Y. Mark Liao, and J.-C. Lin, "Visual Salience-Guided Mesh Decomposition," in *Proc. IEEE International Workshop on Multimedia Signal Processing*, Siena, Italy, Sept. 29-Oct. 1, 2004.

[78] H.-Y. S. Lin, H.-Y. M. Liao, and J.-C. Lin, "A Cognitive Psychology-based Approach for 3-D Shape Retrieval," *Proc. IEEE Int. Conf. on Multimedia and Expo*, Amsterdam, The Netherlands, July 2005.

[79] H.-Y. S. Lin, H.-Y. M. Liao, C.-S. Lu, and J.-C. Lin, "Fragile Watermarking for Authenticating 3-D Polygonal Models," *IEEE Trans. Multimedia*, Vol. 7, No. 6, pp. 997-1006, Dec. 2005.

[80] R. Liu and H. Zhang, "Segmentation of 3-D Meshes through Spectral Clustering," in *Pacific Conference on Computer Graphics and Appications*, Seoul, Korea, Oct 6-8, 2004, pp. 298-305.

[81] M. Lounsbery, T. DeRose, J. D. Warren, "Multiresolution Analysis for Surfaces of Arbitrary Topological Type," *ACM Trans. Graph.* Vol. 16, No. 1, pp. 34-73, 1997.

[82] D. P. Luebke, "A Developer's Survey of Polygonal Simplification Algorithms," *IEEE Computer Graphics and Application*, Vol. 21, No. 3, pp. 34-35, 2001.

[83] S. G. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 11, No. 7, pp. 674-693, 1989.

[84] A. Mangan and R. Whitaker, "Partitioning 3-D Surface Meshes Using Watershed Segmentation," *IEEE Trans. Visualization and Computer Graphics*, Vol. 5, No. 4, pp. 308-321, 2001.

[85] X. Mao, M. Shiba, and A. Imamiya, “Watermarking 3-D Geometric Models through Triangle Subdivision,” in *Proc. SPIE*, Vol. 4314, 2001, pp. 253-260.

[86] MeshNose: The 3-D Shape Objects Search Engine (http://www.deepfx.com/mehnose).

[87] R. Ohbuchi, H. Masuda, and M. Aono, “Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications,” *IEEE J. Select. Areas in Commun.*, Vol. 16, pp. 551-560, 1998.

[88] R. Ohbuchi, H. Masuda, and M. Aono, “Data Embedding for Geometrical and Non-Geometrical Targets in Three-Dimensional Polygonal Models,” *Computer Commun.*, Vol. 21, pp. 1344-1354, 1998.

[89] R. Ohbuchi, H. Masuda, and M. Aono, “A Shape-Preserving Data Embedding Algorithm for NURBS Curves and Surfaces,” in *Proc. Computer Graphics International*, Canmore, Alberta Canada, 1999, pp. 177-180.

[90] R. Ohbuchi, S. Takahashi, and T. Miyazawa, “Watermarking 3-D Polygonal Meshes in the Mesh Spectral Domain,” in *Proc. Graphics Interface*, Ontario, Canada, 2001, pp. 9-17.

[91] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, “A Frequency-Domain Approach to Watermarking 3-D Shapes,” in *Proc. EUROGRAPHICS*, Vol. 21, Saarbrucken, Germany, 2002, pp. 373-382.

[92] R. Osada, T. Funkhouser, B. Chazelle, and D. Dobkin, “Shape Distributions,” *ACM Trans. on Graphics*, Vol. 21, No. 4, pp. 807-832, Oct. 2002.

[93] D. L. Page, A. F. Koschan, and M. A. Abidi, “Perception-based 3-D Triangle Mesh Segmentation Using Fast Marching Watersheds,” *Proc. Computer Vision and Pattern Recognition*, Vol. 2, pp. II27-32, 2003.

[94] E. Paquet and M. Rioux, “Nefertiti: A Query by Content System for Three-dimensional Model and Image Database Management,” *Image and Vision Computing*, Vol. 17, pp. 157-166, 1999.

[95] E. Paquet, M. Rioux, A. Murching, T. Naveen, and A. Tabatabai, “Description of Shape Information for 2-D and 3-D Objects,” *Signal Processing: Image Communication*, Vol. 16, pp. 103-122, 2000.

[96] J. R. Parker, *Algorithms for Image Processing and Computer Vision.* John Wiley and Sons, Inc, 1996.

[97] J. Peng and C.-C. Jay Kuo, "Geometry-guided Progressive Lossless 3-D Mesh Coding with Octree (OT) Decomposition," *ACM Trans. on Graphics*, Vol. 24, No. 3, pp. 609-616, August 2005 (Special issue: *Proc. ACM SIGGRAPH* 2005).

[98] E. Praun, H. Hoppe, and A. Finkelstein, "Robust Mesh Watermarking," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1999, pp. 49-56.

[99] E. Praun, W. Sweldens, and P. Schröder, "Consistent Mesh Parameterizations," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 2001, pp. 179-184.

[100] E. Praun and H. Hoppe, "Spherical Parameterization and Remeshing," *ACM Trans. on Graphics*, Vol. 22, No. 3, pp. 340-349, Aug. 2003 (Special issue: *Proc. ACM SIGGRAPH* 2003).

[101] Princeton 3-D Model Search Engine (http://shape.cs.princeton.edu).

[102] T. R. Reed and H. Wechsler, "Segmentation of Textured Images and Gestalt Organization Using Spatial/Spatial-Frequency Representations," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp. 1-12, 1990.

[103] C. Rössl, L. Kobbelt, and H. P. Seidel, "Extraction of Feature Lines on Triangulated Surfaces Using Morphological Operators," in *Symposium Smart Graphics*, Stanford University, 2000.

[104] Y. Rui, T. S. Huang, M. Ortega, and S. Mehrotra, "Relevance Feedback: A Power Tool for Interactive Content-based Image Retrieval," *IEEE Trans. Circuits and Systems for Video Technology*, Vol. 8, No. 5, pp. 644-655, Sept. 1998.

[105] P. Sander, Z. Wood, G. Gortler, J. Snyder, and H. Hoppe, "Multi-Chart Geometry Images," in *Proc. Eurographics Symposium on Geometry Processing*, 2003, pp. 146-155.

[106] S. Sarkar and K. L. Boyer, "Perceptual Organization in Computer Vision: A Review and A Proposal for A Classificatory Structure," *IEEE Trans. System, Man, and Cybernetics*, Vol. 23, No. 23, pp. 382-400, 1993.

[107] P. Schröder and W. Sweldens, "Spherical Wavelets: Efficiently Representing Functions on the Sphere," *Proc. ACM SIGGRAPH*, Los Angeles, CA, pp. 161-172, 1995.

[108] J. P. Serra, *Image Analysis and Mathematical Morphology.* London: Academic Press, 1982.

[109] C. Shen, J. F. O'Brien, and J. R. Shewchuk, "Interpolating and Approximating Implicit Surfaces from Polygon Soup," *ACM Trans. on Graphics*, Vol. 23, No. 3, pp. 896-904, 2004 (Special issue: *Proc. ACM SIGGRAPH*).

[110] J. Shi and J. Malik, "Normalized Cuts and Image Segmentation," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 8, pp. 888-905, 2000.

[111] Y. Shinagawa, T. L. Kunii, and Y. L. Kergosien, "Surface Coding Based on Morse Theory," *IEEE Computer Graphics and Applications,* Vol. 11, No. 5, pp. 66-78, 1991.

[112] S. Shlafman, A. Tal, and S. Katz, "Metamorphosis of Polyhedral Surfaces Using Decomposition," in *Proc. EUROGRAPHICS*, 2002, pp. 219-228.

[113] H.-Y. Shum, M. Hebert, and K. Ikeuchi, "On Shape Similarity," *Proc. Computer Vision and Pattern Recognition*, pp. 526-531, 1996.

[114] K. Siddiqi and B. B. Kimia, "Part of Visual Form: Computational Aspects," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 17, No. 3, pp. 239-251, 1995.

[115] S. Svensson and G. S. Baja, "Using Distance Transforms to Decompose 3-D Discrete Objects," *Image and Vision Computing*, Vol. 20, pp. 529-540, 2002.

[116] W. Sweldens and P. Schröder, Course 50: Digital Geometry Processing, ACM SIGGRAPH Course Note, 2001.

[117] J. W. H. Tangelder and R. C. Veltkamp, "A Survey of Content Based 3-D Shape Retrieval Methods," *Proc. Shape Modeling International*, pp. 145-156, 2004.

[118] G. Taubin, "A Signal Processing Approach to Fair Surface Design," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1995, pp. 351-358.

[119] G. Taubin and J. Rossignac, "Geometric Compression through Topological Surgery," *ACM Trans. on Graphics*, Vol. 17, No. 2, pp. 84-115, 1998.

[120] G. Turk, "Re-tiling Polygonal Surfaces," in *Proc. ACM SIGGRAPH*, Chicago, IL, 1992, pp. 55-64.

[121] G. Turk, "Generating Random Points in Triangles," in *Graphics Gems*, edited by Andrew Glassner, Academic Press, 1999.

[122] VideoQ: A Object Oriented Video Search Engine (http://persia.ee.columbia.edu/VideoQ/), 1997.

[123] D. Vranic and D. Saupe, "3-D Model Retrieval with Spherical Harmonics and Moments," *Proc. of the DAGM*, pp. 392-397, 2001.

[124] D. Vranic, "An Improvement of Rotation Invariant 3-D Shape Descriptor Based on Functions on Concentric Spheres," *IEEE Int. Conf. on Image Processing*, Sept. 2003.

[125] M. G. Wagner, "Robust Watermarking of Polygonal Meshes," in *Proc. Geometric Modeling and Processing*, Hong Kong, China, 2000, pp. 201-208.

[126] WebSEEK: A Content-based Image and Video Catalog and Search System (http://persia.ee.columbia.edu:8008/), 1996.

[127] K. Wu and M. D. Levine, "3-D Part Segmentation Using Simulated Electrical Charge Distributions," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 19, No. 11, pp. 1223-1235, 1997.

[128] Y. Xiao, P. Siebert, and N. Werghi, "A Discrete Reeb Graph Approach for the Segmentation of Human Body Scans," *Proc. 3-D Digital Imaging and Modeling*, 2003, pp. 378-385.

[129] Yahoo (http://www.yahoo.com.tw/).

[130] Z. Yan, S. Kumar, and C.-C. Jay Kuo, "Mesh Segmentation Schemes for Error Resilient Coding of 3-D Graphic Models," *IEEE Trans. on Circuits and Systems for Video Technology*, Vol. 15, No. 1, pp. 138-144, January 2005.

[131] B. L. Yeo and M. M. Yeung, "Watermarking 3-D Objects for Verification," *IEEE Computer Graphics and Application*, Vol. 19, pp. 36-45, 1999.

[132] M. M. Yeung and B. L. Yeo, "An Invisible Watermarking Technique for Image Verification," in *Proc. Into'l Conf. Image Processing*, Piscataway, N.J., 1997, Vol. 2, pp. 680-683.

[133] K. Yin, Z. Pan, S. Jiaoying, and D. Zhang, "Robust Mesh Watermarking Based on Multiresolution Processing," *Computers and Graphics*, Vol. 25, pp. 409-420, 2001.

[134] C. T. Zahn, "Graph-Theoretical Methods for Detecting and Describing Gestalt Clusters," *IEEE Trans. Computers*, Vol. C-20, No. 1, pp. 68-86, 1971.

[135] C. Zhang and T. Chen, "Efficient Feature Extraction for 2-D/3-D Objects in Mesh Representation," *Proc. IEEE Int. Conf. Image Processing*, Vol. 3, pp. 935-938, 2001.

[136] S. C. Zhu, "Embedding Gestalt Laws in Markov Random Fields," *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 21, No. 11, pp. 1170-1187, 1999.

[137] D. Zorin, P. Schröder, and W. Sweldens, "Interactive Multiresolution Mesh Editing," in *Proc. ACM SIGGRAPH*, Los Angeles, CA, 1997, pp. 256-268.

[138] E. Zuckerberger, A. Tal, and S. Shlafman, "Polyhedral Surface Decomposition with Applications," *Computers and Graphics*, Vol. 26, No. 5, pp. 733-743, 2002.

Publication List of Hsueh-Yi Sean Lin

A. Journal Papers

1. Hsueh-Yi Sean Lin, Hong-Yuan Mark Liao, and Ja-Chen Lin, “Visual Salience-Guided Mesh Decomposition,” *IEEE Trans. Multimedia*, accepted and to appear.

2. Hsueh-Yi Sean Lin, Hong-Yuan Mark Liao, Chun-Shien Lu, and Ja-Chen Lin, “Fragile Watermarking for Authenticating 3-D Polygonal Meshes,” *IEEE Trans. Multimedia*, Vol. 7, No. 6, pp. 997-1006, Dec. 2005.

B. Conference Papers

1. Hsueh-Yi Sean Lin, Hong-Yuan Mark Liao, and Ja-Chen Lin, “A Cognitive Psychology-based Approach for 3-D Shape Retrieval,” *Proc. IEEE Int. Conf. on Multimedia and Expo*, Amsterdam, The Netherlands, Jul. 6-8, 2005.

2. Hsueh-Yi Sean Lin, Hong-Yuan Mark Liao, and Ja-Chen Lin, “Visual Salience-Guided Mesh Decomposition,” *Proc. IEEE Int. Workshop on Multimedia Signal Processing*, Siena, Italy, Sept. 29-Oct. 1, 2004.

3. Hsueh-Yi Lin, Hong-Yuan Mark Liao, Chun-Shien Lu, and Ja-Chen Lin, “Authentication of 3-D Polygonal Meshes,” *Proc. 2nd Int. Workshop on Digital Watermarking (IWDW)*, Seoul, Korea, Oct. 20-22, 2003, LNCS 2713, pp. 168-183.

4. Hsueh-Yi Lin, Hong-Yuan Mark Liao, Chun-Shien Lu, and Ja-Chen Lin, “Fragile Watermarking for Authenticating 3D Polygonal Meshes,” *Proc. 16th IPPR Conf. on CVGIP*, Taiwan, Aug. 17-19, 2003, pp. 298-304 (Paper award).

C. Other Publications

1. Hsueh-Yi Lin, “A 3-D Deformation Based Technique for Rectifying Registration Errors,” *M.S. Thesis*, *Yuan Ze University*, June 1999.

2. 林學億、許格超、鐘俊仁及梅普華編譯, “Visual C++ 程式設計指南—應用程式開發架構篇,” 松崗, 1999。

3. 林學億、許格超、鐘俊仁及梅普華編譯, “Visual C++ 程式設計指南—資料庫及 Internet 系統開發進階篇,” 松崗, 1999。