Information Analysis for Steganography and Steganalysis in 3D Polygonal Meshes

YING YANG

Information Analysis for Steganography and Steganalysis in 3D Polygonal Meshes

Ying Yang

A Thesis presented for the degree of
Doctor of Philosophy

School of Engineering and Computing Sciences
University of Durham
United Kingdom

October 2013

Information Analysis for Steganography and Steganalysis in 3D Polygonal Meshes

Ying Yang

Submitted for the degree of Doctor of Philosophy
October 2013

Abstract

Information hiding, which embeds a watermark/message over a cover signal, has recently found extensive applications in, for example, copyright protection, content authentication and covert communication. It has been widely considered as an appealing technology to complement conventional cryptographic processes in the field of multimedia security by embedding information into the signal being protected. Generally, information hiding can be classified into two categories: *steganography* and *watermarking*. While steganography attempts to embed as much information as possible into a cover signal, watermarking tries to emphasize the robustness of the embedded information at the expense of embedding capacity.

In contrast to information hiding, *steganalysis* aims at detecting whether a given medium has hidden message in it, and, if possible, recover that hidden message. It can be used to measure the security performance of information hiding techniques, meaning a steganalysis resistant steganographic/watermarking method should be imperceptible not only to Human Vision Systems (HVS), but also to intelligent analysis.

As yet, 3D information hiding and steganalysis has received relatively less attention compared to image information hiding, despite the proliferation of 3D computer graphics models which are fairly promising information carriers. This thesis focuses on this relatively neglected research area and has the following primary objectives: 1) to investigate the trade-off between embedding capacity and distortion by considering the correlation between spatial and normal/curvature noise in triangle meshes; 2) to design satisfactory 3D steganographic algorithms, taking into account this tradeoff; 3) to design robust 3D watermarking algorithms; 4) to propose a steganalysis framework for detecting the existence of the hidden information in 3D models and introduce a universal 3D steganalytic method under this framework.

The thesis is organized as follows. Chapter 1 describes in detail the background relating to information hiding and steganalysis, as well as the research problems this thesis will be studying. Chapter 2 conducts a survey on the previous information hiding techniques for digital images, 3D models and other medium and also on image steganalysis algorithms.

Motivated by the observation that the knowledge of the spatial accuracy of the mesh vertices does not easily translate into information related to the accuracy of other visually important mesh attributes such as normals, Chapters 3 and 4 investigate the impact of modifying vertex coordinates of 3D triangle models on the mesh normals. Chapter 3 presents the results of an empirical investigation, whereas Chapter 4 presents the results of a theoretical study. Based on these results, a high-capacity 3D steganographic algorithm capable of controlling embedding distortion is also presented in Chapter 4.

In addition to normal information, several mesh interrogation, processing and rendering algorithms make direct or indirect use of curvature information. Motivated by this, Chapter 5 studies the relation between Discrete Gaussian Curvature (DGC) degradation and vertex coordinate modifications.

Chapter 6 proposes a robust watermarking algorithm for 3D polygonal models, based on modifying the histogram of the distances from the model vertices to a point in 3D space. That point is determined by applying Principal Component Analysis (PCA) to the cover model. The use of PCA makes the watermarking method robust against common 3D operations, such as rotation, translation and vertex reordering. In addition, Chapter 6 develops a 3D specific steganalytic algorithm to detect the existence of the hidden messages embedded by one well-known watermarking method. By contrast, the focus of Chapter 7 will be on developing a 3D watermarking algorithm that is resistant to mesh editing or deformation attacks that change the global shape of the mesh.

By adopting a framework which has been successfully developed for image steganalysis, Chapter 8 designs a 3D steganalysis method to detect the existence of messages hidden in 3D models with existing steganographic and watermarking algorithms. The efficiency of this steganalytic algorithm has been evaluated on five state-of-the-art 3D watermarking/steganographic methods. Moreover, being a universal steganalytic algorithm can be used as a benchmark for measuring the anti-steganalysis performance of other existing and most importantly future watermarking/steganographic algorithms.

Chapter 9 concludes this thesis and also suggests some potential directions for future work.

Declaration

The work in this thesis is based on research carried out at the School of Engineering and Computing Sciences, Durham University, United Kingdom. No part of this thesis has been submitted elsewhere for any other degree or qualification and it is all my own work unless referenced to the contrary in the text.

Publications

The contents of this thesis are based on the results from the following papers.

- **Ying Yang** and Ioannis Ivrissimtzis, "A logistic model for the degradation of triangle mesh normals," *7th International Conference on Curves and Surfaces 2010, LNCS* vol. 6920, pp. 697 – 710, 2012. (Chapter 3)

- **Ying Yang**, Norbert Peyerimhoff and Ioannis Ivrissimtzis, "Linear correlations between spatial and normal noise in triangle meshes," *IEEE Transactions on Visualization and Computer Graphics*, vol. 19, no. 1, pp. 45 – 55, 2013. (Chapter 4)

- **Ying Yang**, Norbert Peyerimhoff and Ioannis Ivrissimtzis, "Curvature degradation in quantised triangle meshes," *Under Preparation.* (Chapter 5)

- **Ying Yang** and Ioannis Ivrissimtzis, "Polygonal mesh watermarking using Laplacian coordinates," *Computer Graphics Forum (Proceedings of Eurographics/ACM SIGGRAPH Symposium on Geometry Processing, SGP 2010)*, vol. 29, no. 5, pp. 1585 – 1593, 2010. (Chapter 7)

- **Ying Yang** and Ioannis Ivrissimtzis, "Mesh Discriminative Features for 3D Steganalysis," Accepted to *ACM Transactions on Multimedia Computing, Communications and Applications.* (Chapter 8)

In addition to the above, the following paper was produced during my PhD studies.

- **Ying Yang**, David Günther, Stefanie Wuhrer, Alan Brunton, Ioannis Ivrissimtzis, Hans-Peter Seidel, Tino Weinkauf, "Correspondences of Persistent Feature Points on Near-Isometric Surfaces," *ECCV Workshop on Non-Rigid Shape Analysis and Deformable Image Alignment (NORDIA), LNCS* vol. 7583, 2012.

Acknowledgements

This thesis would not have been possible without the support of many people. First and foremost, I offer my sincerest gratitude to my supervisor, Dr. Ioannis Ivrissimtzis, who has offered me invaluable assistance, support, guidance and knowledge throughout my thesis whilst allowing me the room to work in my own way. I appreciate all his contributions of time and ideas to make my PhD experience productive and stimulating.

I wish to thank Prof. Hans-Peter Seidel and Dr. Tino Weinkauf for offering me the excellent opportunity to do six-month internship in the Computer Graphics Department at Max-Planck-Institut für Informatik, Saarbrücken, Germany. This experience has led me into a very interesting research area. During the period of internship, I have been aided consistently in developing and implementing the algorithms by Dr. Tino Weinkauf, Dr. Stefanie Wuhrer, David Günther and Alan Brunton. The intensive and frequent discussions with them have allowed the project to move forward smoothly.

Special thanks go to Durham University who has funded my PhD studies under Durham University Doctoral Fellowship scheme and to the School of Engineering and Computing Sciences who has provided the support and equipment needed to produce and complete my thesis.

Last but most importantly, I would express my love and gratitude to my beloved parents, Yang Yuanhou and Chen Xinggui, for their constant love and a solid foundation upon which I continue to build to this day, and to my dear wife, Chang Liu, for her endless love, support and encouragement throughout my studies at Durham.

Contents

List of Figures

List of Tables

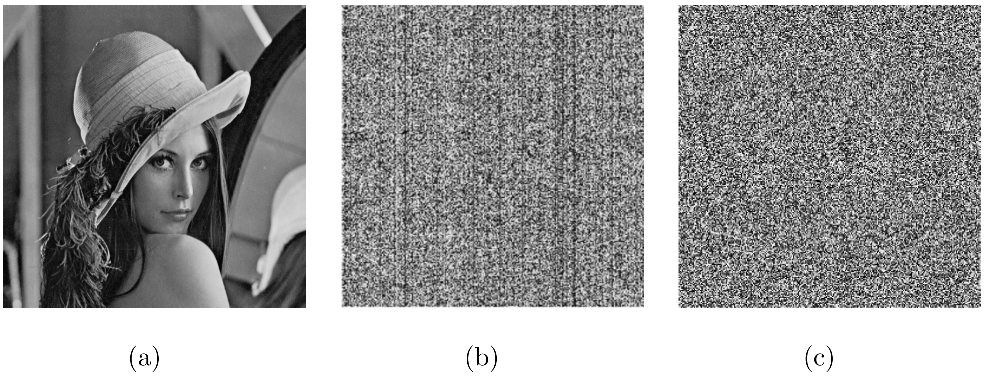
Chapter 1

Introduction

In the information era, multimedia content (e.g, audio, image, video and 3D computer graphics models) in digital form is being used in a wide range of application areas. However, at the same time, an increasing number of security problems have been revealed. For instance, the proliferation of intelligent editing tools can also facilitate misuse, illegal copying and distribution, plagiarism and misappropriation, which could seriously ruin the interests of the creator or owner of the multimedia work. This is creating a strong need for schemes that can efficiently cope with multimedia security and privacy, including copyright protection and integrity authentication [14].

Previously, security-related issues were commonly addressed using cryptographic algorithms, such as RSA [15], DSA [16] and ECC [17]. That is, cryptography enables the security of the message by using a key to encrypt it and hence only the person who possesses the correct decryption key can decrypt the encrypted message and read the original message, unless, of course, the encryption system has been broken. Generally, there are two basic kinds of encryption algorithms: *private key* and *public key* cryptography. The former uses the same key to encrypt and decrypt the message of interest and is also known as symmetric key cryptography. The latter utilizes a public key to encrypt the message and a private key to decrypt it and is also known as asymmetric key cryptography.

Cryptography is being widely used in a wide range of practical applications, such as ATM cards, computer passwords and electronic commerce; however, it is



(a) (b) (c)

Figure 1.1: Illustration of image encryption. (a) original image, (b) encrypted image by Ye's method [1], and (c) encrypted image by Gao et al.'s method [2].

ineffective in the context of multimedia data security mainly due to the following two reasons. Firstly, once the encrypted multimedia data have been successfully decrypted, cryptography does not offer an efficient way to track its reproduction or retransmission [14]. In other words, a pirate can purchase the decryption key, decrypt the date, using the key, to obtain an unprotected copy of the multimedia content and then proceed to distribute the illegal copies. Secondly, the multimedia content when encrypted likely undergoes significantly visual degradation and thus needs decryption before use, which might be computationally expensive. An illustration of image quality degradation as a result of encryption is shown in Fig. 1.1. Fig. 1.1 (b) and (c) appear to carry meaningless visual information and hence are of no practical use in real applications prior to decryption.

The above example of image encryption implies the expectation that after adopting certain protection schemes the visual degradation of multimedia data should not be beyond a certain tolerance, avoiding incurring a heavy penalty of the content. Unlike protecting other kinds of data (e.g., a military document or a private message) where we make our best effort to conceal the real content of the data itself by turning it into an unintelligible or unreadable ciphertext, usually there are different requirements regarding the protection of multimedia data. Therefore, there is a need for an alternative scheme to complement cryptography in multimedia data protection. The alternative should satisfy the following requirements.

- It is able to make successful solution to a disputation over the ownership of the multimedia content possible.

- It does not significantly degrade the visual quality of the multimedia data being protected and should maintain the distortion below the threshold that our human eyes can tolerate.

Information hiding or *data hiding*, which refers to the process of imperceptibly embedding a secret message into a cover/host media so as to conceal the content of the hidden message, is of the above-mentioned properties and has been widely regarded as a promising solution to multimedia security and privacy.

In the rest of the chapter, we shall describe the background and principles of *information hiding* and *steganalysis*, present the problems we will be investigating in this thesis and the research plan that helps to cope with these problems.

1.1 Information Hiding

Since 1990s, the investigation of a technology that is able to serve as a complement to cryptography has attracted extensive attention from both academic and industrial organizations [3, 18]. *Information hiding* has been widely deemed as a fairly promising technology to fulfill this purpose. Information hiding works by secretly embedding a message within a host digital signal. The message to be embedded can be whatever we want to insert, such as a personal identification code, a company logo or a to-be-delivered secret information string, depending on the specific application scenarios. Because of its potential applications, information hiding has become an emerging research area over the past few decades.

In contrast to cryptography which tries to make the meaning of information obscure to a person who intercepts it, information hiding technique aims chiefly at keeping the existence of the information secret or making the embedded information imperceptible [19]. While cryptography arouses suspicion due to the unreadability of encrypted information, information hiding avoids this through invisible message embedding. The embedding of message is achieved by employing human binaural or perceptual redundancies. More specifically, the redundancies are the details of

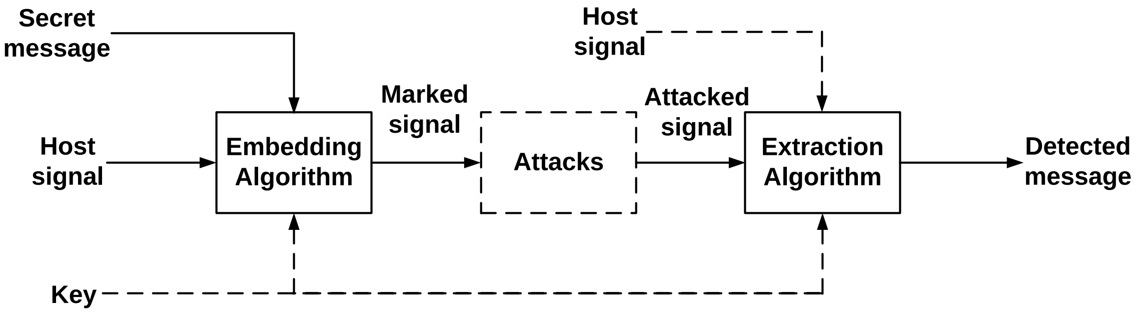


Figure 1.2: A generic framework for message embedding and extraction.

a multimedia signal that a human ear cannot hear or that a human eye cannot perceive. Basic hiding techniques include modifying the least significant bits of the pixels of a host image [20, 21] and adding tabs and spaces at the end of the lines of an HTML document [22].

As Fig. 1.2 shows, information hiding usually consists of two procedures: *message embedding* and *message extraction*. The embedding process takes the host/carrier signal, the secret message and probably the private key as the input and yields the watermarked/stego signal. It is noted that for security reason, cryptography is generally incorporated into information hiding systems in case of correct detection/extraction of the hidden information by an adversary. During distribution, the watermarked signal may be subject to routine operations, such as compression and format conversion, or even worse, malicious attacks, such as noise addition and cropping, by an attacker with the hope of removing the embedded information. The malicious attack likely occurs when the stego signal has aroused suspicion, or when it has been detected to carry hidden data by steganalysis techniques [23, 24]. The extraction process is the inverse of the embedding process. The decoder extracts the embedded message from the stego signal, or probably the attacked stego one, with the assistance of the private key and/or the original signal shared with the embedder.

Fig. 1.3 shows that information hiding, as a general term, encompasses a number of applications, such as digital watermarking, fingerprinting and steganography. Among these applications, two important branches that we study in this thesis are *steganography* and *watermarking*.

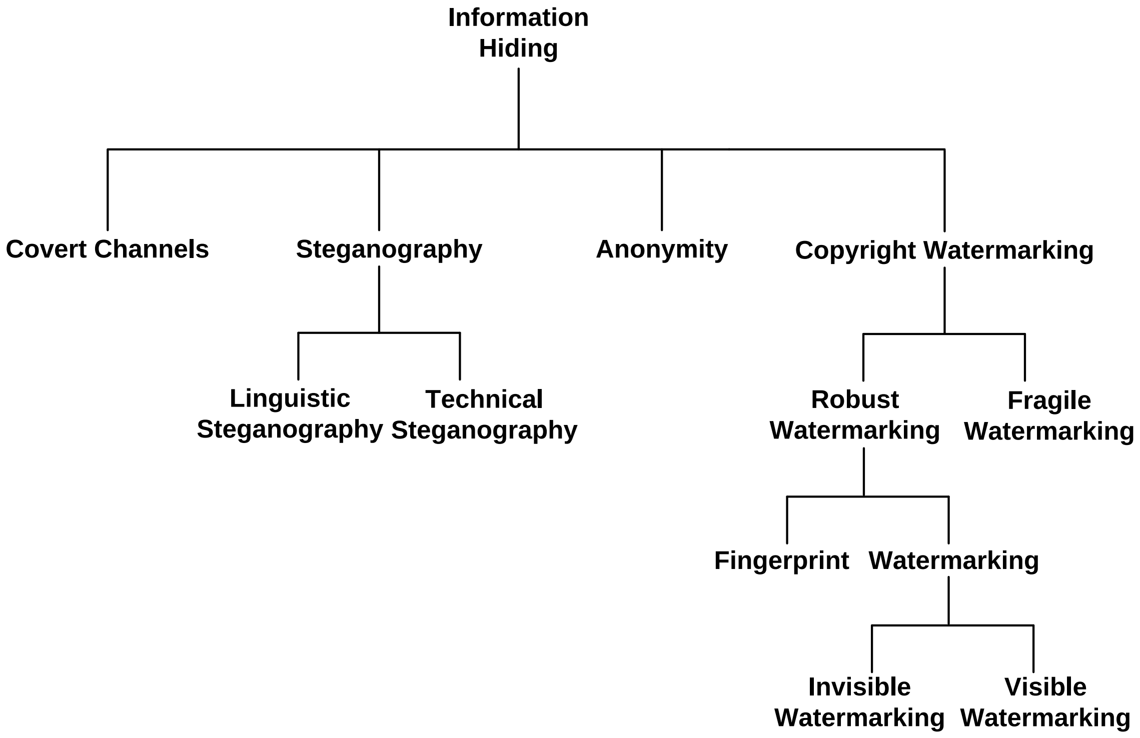


Figure 1.3: A classification of the different types of information hiding based on [3].

1.1.1 Steganography

Steganography is the art and science of covert communications among trusting parties, where the confidential message is embedded imperceptibly about an innocent-looking cover signal so that nobody apart from the sender and the intended recipient can detect the existence of the hidden data. The technical term steganography, derived from the Greek words steganos ($\sigma\tau\epsilon\gamma\alpha\upsilon\acute{o}\varsigma$) and graphein ($\gamma\rho\acute{\alpha}\varphi\epsilon\iota\upsilon$), literally means "covered or protected writing" [18, 25]. Thus, a steganographic system must provide a method to embed data imperceptibly, allow the data to be readily extracted, promote a high embedding capacity, and preferably incorporate a certain degree of resistance to removal [26].

One can say, whereas classical cryptography is about concealing the content of messages, steganography is about concealing their presence. In addition, prudent cryptographic algorithm is based on the assumption that the method employed for data encryption is known to the public and that security lies in the private key rather than in the encryption algorithm [27]. However, this principle is not immediately applicable to steganographic systems. This is the case with cryptography as well, where many purveyors of such systems keep their mechanisms subject to nondisclosure agreements [19].

Steganography was already in use thousands of years ago in ancient Greece and China [23, 26, 28].Regarding modern applications in the digital era, the publishing and broadcasting industries have become interested in techniques for hiding the encrypted copyright marks and serial numbers in digital films, audio recordings, books, and multimedia products [19]. Other applications for steganography include military communications, where military agencies take advantage of steganographic technique to make signals difficult for the enemy to detect or jam [3], and medical imaging systems, where a link between the patient's image and his/her captions (e.g., physician, patient's name, address and other particulars) is considered necessary [29]. Thus, embedding the patient's information about the image could be an efficient safety measure to help solve such a problem.

Modern steganographic algorithms do the utmost to keep the presence of the hidden message imperceptible and confidential while maintaining high embedding capacity. Nevertheless, steganography, due to its invasive nature, has to alter the original cover object at the phase of embedding. This process probably leaves some traceable clues, that is, disturbance of the statistics of the carrier object [30,31], even though the alterations may be too slight to perceive. Consequently, these clues promote another technology referred to as *steganalysis*, an opposite to steganography, whose purpose is to detect the existence of the hidden messages; this is analogous to cryptanalysis applied to cryptography.

1.1.2 Watermarking

Similar to steganography, watermarking refers to the process of inserting messages, often called as *watermark* in this circumstance, into a host multimedia signal without incurring noticeable visual degradation. The main focus of a watermarking system is to achieve a high level of robustness against attacks. In other words, it is impossible or fairly difficult to remove the embedded watermark without degrading the stego object's visual quality.

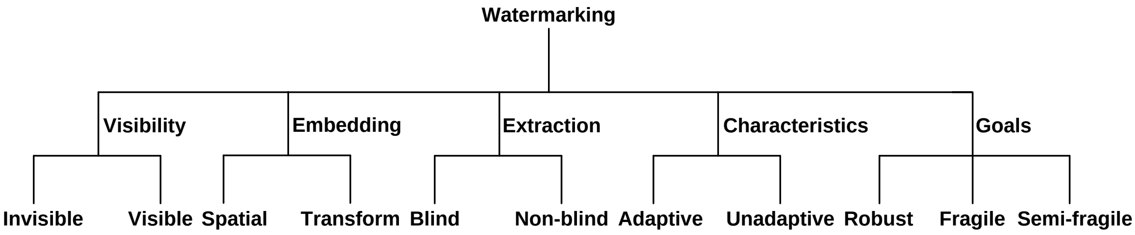
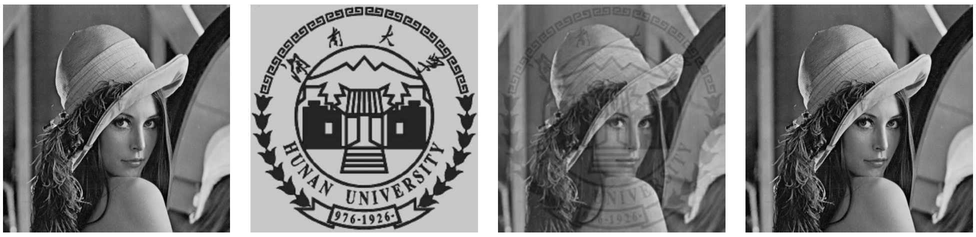


Figure 1.4: Classification of watermarking schemes.



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

Figure 1.5: Comparison between the visible and invisible watermark embedding. (a) original image, (b) watermark, (c) visibly watermarked image by Yang et al.'s method [4], and (d) invisibly watermarked image by embedding the watermark image into the original image.

As Fig. 1.4 shows, watermarking can be divided into several categories, depending on the classification criterion.

- By visibility, the watermark embedded is either *invisible* or *visible*, as illustrated in Fig. 1.5. Intuitively, the watermark that we cannot perceive by eye is referred to as visible; otherwise, invisible. To date, invisible watermarking has been widely studied, and thousands of papers on it have been reported. The research on visible watermarking, compared with that on invisible watermarking, has received less attention, but this has changed dramatically over the last few years [4, 32–37].

- By embedding domain, the watermark can either be inserted into the spatial domain of the cover signal directly (e.g., the pixel values of an image) [38, 39], or be embedded into the transform domain obtained by applying certain transformation (e.g., DFT, DCT, and DWT) to the cover signal [14, 40, 41]. It has been observed that the transform-based watermarking methods tend to provide higher resistance against attacks than the spatial-based ones, so an increasing number of researchers would investigate the watermarking approaches that embed watermark in the transform domain.

- Depending on extraction method, the watermarking methods can be classified as blind and non-blind. In the former, the extraction process of watermark is carried out blindly without reference to the original cover object, or vice versa. In comparison to non-blind watermarking, blind watermarking obviously has a relatively wider application scope; however, it offers weaker robustness.

- By adaptivity, the watermarking methods can be divided as adaptive and non-adaptive. Generally, the watermarking that takes into account the content of the host medium and varies the watermark strength in different embedding regions to avoid obtrusiveness is regarded as adaptive; otherwise, non-adaptive. The adaptivity is commonly achieved as a result of taking advantage of the characteristics of HVS (Human Visual System) [42] and/or HAS (Human Auditory System) [43], each of which is of distinct sensitivity to the changes in different embedding areas. For example, it is much easier for the HVS to observe the change in the smooth regions of an image than in the rough areas.

- By goal, the watermarking methods can be classified as robust, fragile and semi-fragile. Robust watermarking attempts to make the watermark capable of surviving all kinds of attacks for the purpose of, for instance, copyright protection. By contrast, a fragile watermark is such a mark that it is readily altered or destroyed when the stego image is modified. The very sensitivity of fragile marks to modification leads to their use in authentication applications. Semi-fragile watermarking is robust to acceptable content preserving manipulations (e.g., JPEG compression) while fragile to malicious attacks. Since it is, by definition, able to discriminate normal processing from evil attacks, it has found application in authentication as well.

1.1.3 Steganography Versus Watermarking

The above descriptions indicate that, while the two terms, namely *steganography* and *watermarking*, are closely related to each other and share a great deal of overlap, they can yet be distinguished from each other by their own inherent properties or requirements. More specifically, they distinguish each other in terms of embedding capacity and robustness and they have distinct application domains.

The main goal of steganography is to hide a message in covert point-to-point communications between two trustable parties, so steganographic methods especially emphasize embedding capacity and usually provide weaker robustness against attacks.

By contrast, the target of watermarking is to hide a message in one-to-many communications among several parties, and as a result, watermarking schemes often offer relatively lower embedding capacity and should withstand common operations, as well as the malicious attacks that attempt to remove or modify the hidden message. It is worth mentioning here that embedding capacity is generally achieved at the expense of robustness, and vice versa.

In summary, we expect watermarking algorithms to be able to survive both unintentional and malicious attacks, but we do not have such an expectation on steganographic approaches. Due to their respective inherent properties, steganography and watermarking have different applications. Steganography is used in the applications, for instance covert communications, where the transmission of a large amount of information is needed. Watermarking is more appropriate for content authentication, ownership authentication and copy control, where the robustness of watermark rather than the capacity is more important.

1.2 Steganalysis

In contrast to steganography and watermarking, steganalysis is the art and science of detecting whether a given medium has hidden message in it, and, if possible, recover that hidden message. The message is hidden using steganography or watermarking; this is analogous to cryptanalysis applied to cryptography.

Steganalysis is indeed a very challenging task, due to the wide diversity of natural media, the wide variation of data embedding algorithms and usually the low embedding distortion. Despite these, steganalysis is still possible since data embedding will nevertheless disturb the statistics of a host medium [30]. In other words, the presence of embedded messages still makes an original cover medium and its corresponding stego-version different in some aspects, though this presence is often imperceptible to the human eye or ear.

Depending on the applicability, steganalysis methods can be generally classified into two categories: *specific* and *universal*. While the former aims at breaking a specific steganographic/watermarking algorithm, the latter attempts to frustrate all the steganographic/watermarking algorithms. In general, specific approaches achieve higher detection accuracy as compared to universal ones because they have prior knowledge of how the specific target method works. Nevertheless, universal steganalysis is more attractive in practical use since they can work independently of the embedding technique and even generalize to unknown algorithms.

Steganalysis can be considered as a task of pattern recognition, where it determines which class (clean medium with no hidden message or stego-medium with hidden message) a given medium belongs to. As such, a general principle of designing a steganalysis algorithm is to identify and extract features that are particularly sensitive to data embedding, that is, those features that are able to capture the variations resulting from embedding. This means that the features extracted from clean media are expected to be quite different from those from the stego-media [44]. Generally, the larger the difference is, the better the choice of features is considered.

Following feature extraction, a classifier is usually designed to distinguish the non-marked and marked medium via training the features. Overall, the performance of a steganalysis system relies heavily on both feature extraction and classifier design.

Typically, an N-dimensional feature vector is generated from each single medium and thus, each medium is represented as a point by a feature vector in the N-dimensional space. Naively, it might be expected that N should be as large as possible for the feature vector to be more effective. However, recent studies [23, 24] show that a very large value of N is not necessary and that a large N may result in high computational costs and/or negative impact on the detection accuracy.

To create a classifier, the feature vectors extracted from a training set of medium with and without hidden information are fed into a machine learning algorithm in the hope of separating the two kinds of medium. Thus far, both linear discriminant analysis (LDA), such as Fisher Linear Discriminant [30], and nonlinear discriminant analysis, such as kernelised Support Vector Machines (SVM) [45], have been been successfully employed in prior steganalysis works.

1.3 3D Models

3D computer graphics, in contrast to 2D computer graphics, are graphics that use a three-dimensional representation of geometric data stored in the computer for the purposes of performing calculations and rendering 2D images. The process of creating 3D models can be sequentially decomposed into three basic stages, namely, *modeling*, *scene layout setup* and *rendering*.

3D modeling is the process of forming the shape of an object. There are a number of modeling techniques and several 3D model representations, for example, constructive solid geometry, NURBS modeling, polygonal modeling, subdivision surfaces and implicit surfaces. Here, we deal with polygonal models which are the ubiquitous standard for surface representation in graphics and visualization applications. Particularly, triangle meshes play an important role in CAD/CAM applications when scans of physical objects are processed into triangle mesh models, or when NURBS surfaces are converted into triangle meshes for fast interactive visualizations.

A 3D polygonal model is formed by a set of 3D points, called *vertices*, connected between them with a set of polygons, called *faces*. A polygon is traditionally a plane figure bounded by a closed path that is composed of a finite sequence of straight line segments. These segments are called *edges* or *sides* and the points where two edges meet are the polygon's vertices. An n-gon is a polygon with n sides. A 3D model that is composed of 3-gons only is referred to as a *triangle* mesh, while a 3D model that is composed of 4-gons only is called a *quadrilateral* mesh. An an illustration, Fig. 1.6 shows a triangle model and a quadrilateral mesh.

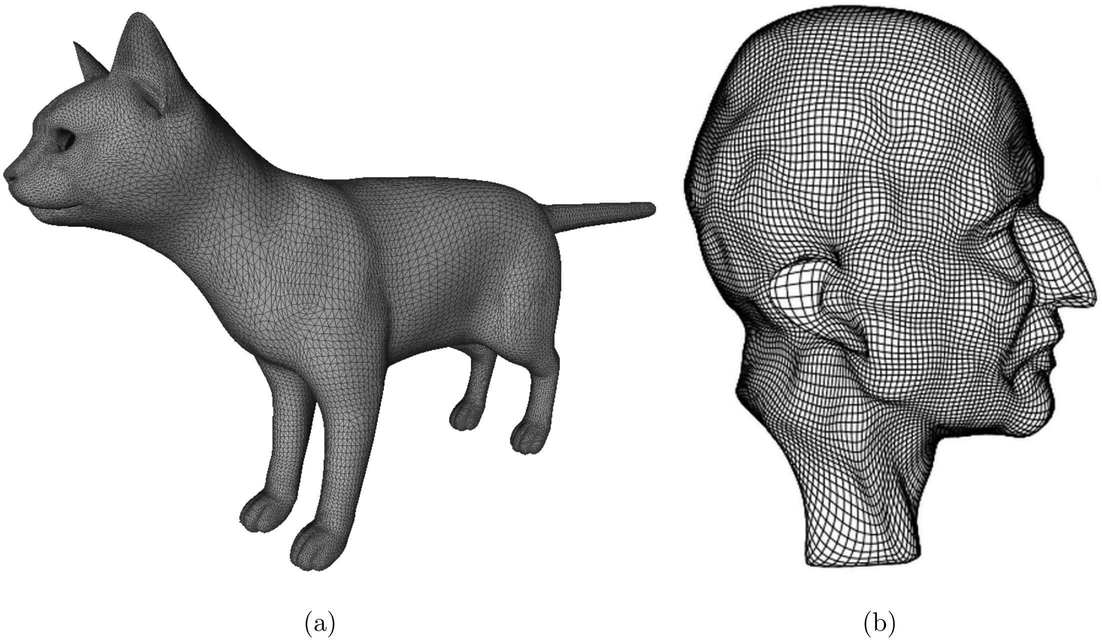


Figure 1.6: An illustration of (a) a triangle mesh and (b) a quadrilateral mesh.

The scene layout setup arranges virtual objects, lights, cameras and other entities on a scene. This step is helpful before rendering an object into an image. Lighting plays a significant role in during scene setup, and more specifically it is the main contributing factor to the resulting aesthetic and visual quality of the finished

Following scene layout setup, the rendering stage converts a prepared scene into an image or an animation from the prepared scene. The two basic operations in realistic rendering are *transport* and *scattering* [46]. While the former is about how much light gets from one place to another, the latter is about how surfaces interact with light. Given the complex variety of physical processes being simulated, the rendering process is generally computationally expensive. Fortunately, due to the recent breakthroughs in computer processing power, a progressively higher degree of realistic rendering is possible.

1.4 Problem Statement

Over the past few years, digitization has become incredibly prevalent across a wide range of sectors of economic and social life. The proliferation of digitized objects could, however, lead to various security and privacy issues that require high-level and immediate attention. The potential issues include, for instance, the protection of ownership and authorship as well as the authentication of integrity.

Having the above in mind, researchers from both academic and industrial communities have made a great effort to investigate information hiding, which is commonly deemed as a replacement of cryptography in the context of digital multimedia. Given their immediate practical applications, information hiding algorithms have been developed for objects of various dimensions, including digital images [40], text documents [47], audio [48] and video [49]. However, the research into the information embedding of 3D models has received relatively less attention, although the proliferation of the 3D graphical models in various application scenarios shows them to be fairly promising message carriers. As pointed out in [50], this situation is chiefly due to the difficulties encountered while handling the arbitrary topology and irregular sampling of 3D meshes, as well as the existence of various intractable attacks on watermarked 3D models. Moreover, in spite of the advancements of text/image/audio/video data embedding techniques, they cannot be applied to mark 3D objects directly.

As a result, several of the 3D data hiding algorithms that have already been proposed have significant shortcomings. In terms of 3D watermarking, most of the previous watermarking algorithms are indeed robust against common attacks, such as geometric transformations, noise addition and mesh smoothing, but do not survive more realistic processing attacks, for instance, mesh editing attacks. Regarding 3D steganography, most of the existing steganographic approaches have low embedding capacity. Also, they are not able to control the embedding distortion resulting from message embedding. One obvious disadvantage of the loss of control over embedding distortion is that the embedding of message likely results in unpleasant visual quantity, that is, the level of quality degradation is higher than expected.

As the counterpart of steganography and watermarking, steganalysis attempts to detect the presence of information embedded through steganography and watermarking methods and hence further stimulate their development. Thus far, a number of steganalysis approaches have been proposed for detection of message hidden over digital images. However, to the best of our knowledge, there is no reported steganalysis that is able to work on detecting hidden information within 3D computer graphics models.

The purpose of this thesis is to mathematically study the tradeoff between embedding capacity and distortion, to develop novel 3D information hiding algorithms and also to develop 3D steganalysis algorithms as a stimulus to the development of 3D information hiding. Thus after a comprehensive review of the literature, we will be conducting research into the following broad topics: *3D steganography*, *3D watermarking* and *3D steganalysis*.

1.5 Overview

The layout of the thesis reflects the research plan for dealing with the problems as mentioned above. The main results and the evaluation methodology are discussed.

1.5.1 Thesis Overview

Chapter 2 will be reviewing the related work, including steganography, watermarking and steganalysis algorithms for digital image, video, audio, text and 3D models.

Chapters 3, 4 and 5 study the trade-off between embedding capacity and visual distortion. The results from these three chapters will help to better understand the relationship between the degradation in 3D visual quality and the modification of 3D geometry and hence help to devise more excellent data hiding algorithms.

As the visual quality of the mesh is mainly governed by the quality of its normal and curvature information, Chapter 3 will be empirically investigating the degradation of face normals resulting from perturbation of vertex coordinates.

In Chapter 4, we will theoretically investigate the same problem and propose a Least Significant Bits (LSB) data hiding algorithm, utilizing this trade-off for achieving maximum embedding capacity for a given tolerance of normal degradation. As well as its simplicity, the method has the merits of high-capacity and low-distortion.

Chapter 5 will be looking into how the discrete Gaussian curvature will change when modifying the geometry of 3D triangle model.

Following the previous five chapters, we have presented two 3D watermarking methods, that is, a histogram-based watermarking in Chapter 6 and a Laplacian coordinates-based watermarking in Chapter 7. Moreover, a *specific* steganalysis has been proposed in Chapter 6 to detect the presence of the watermark embedded by Cho et al. [7].

Chapter 8 introduces a *universal* 3D steganalysis, so that it can be used for detecting the existence of the messages embedded by the previous 3D steganography and watermarking and probably the future algorithms.

We conclude this thesis and suggest some potential research directions in Chapter 9.

1.5.2 Evaluation Methodology

For the proposed steganographic algorithm, it is of a high embedding capacity and a low visual distortion. The 3D watermarking methods are strongly robust against routine 3D processing, including rotation, translation, uniform scaling and vertex re-ordering, as well as malicious attacks, including noise addition, smoothing, quantization and normal mesh editing. In addition to robustness, the proposed methods are able to offer acceptable embedding capacity without incurring noticeable visual distortion after embedding.

The efficiency of our proposed watermarking and steganography is evaluated in terms of embedding capacity, embedding distortion and robustness. We evaluate the embedding distortion between the original mesh and its corresponding marked one by root mean square error (RMSE) and Hausdorff distance. The robustness is verified according to the correlation coefficient between the original message string and the extracted one. Moreover, we further demonstrate the performance of the proposed steganographic and watermarking schemes via comparing them to the state-of-the-art ones.

The performance of steganalysis is measured by accurate detection rate. To do so, a big training dataset composed of a number of the clean or non-marked and watermarked 3D models is prepared for obtaining a classifier that is able to distinguish the non-watermarked and watermarked 3D models and a big test dataset composed of a number of the clean or non-marked and watermarked 3D models is used for validating the distinguishing capacity of the classifier and thus of the proposed steganalysis algorithm.

Concerning the limitations, the proposed LSB replacement data embedding method is very fragile and is unable to withstand attacks. Although they have been demonstrated to be robust against a wide range of attacks, the two proposed watermarking methods can be improved to achieve better performance. The proposed 3D steganalytic method only detects the messages hidden by those algorithms that embed information into the mesh geometry, not into mesh connectivity, or into the data redundancy of polygonal list files.

Chapter 2

Literature Review

In this chapter, we conduct a survey of previous work related to this thesis. In Section 2.1, we review the information hiding algorithms for digital images, including image steganography in Section 2.1.1 and watermarking in Section 2.1.2.

Next, we survey 3D information hiding techniques from these two perspectives, discussing 3D steganography in Section 2.2.1 and 3D watermarking in Section 2.2.2.

In Section 2.3, we briefly describe the information hiding approaches for other media, including video, audio and text.

In addition to information hiding, steganalysis methods are reviewed. Taking into account the fact that the previous steganalysis methods are mainly targeting digital images, we only cover the image steganalysis in Section 2.4.

2.1 Image Information Hiding

Over the past few year, we have witnessed remarkable advances in digital image processing and computational photography, resulting in sophisticated image-editing software systems. The ease of digital image manipulation has created a need for information hiding techniques that can, for example, achieve covert communication and protect copyright ownership. As such, there have been a great deal of research effort dedicated to the area of information hiding for digital images since 1990s. As there are many works in this area, for conciseness we will only review the most representative algorithms here.

2.1.1 Image Steganography

Among various image steganographic schemes, least-significant bit (LSB) steganography, including LSB replacement and matching embedding, is one of the fundamental embedding techniques. The two methods are simple and easy to implement, with the capability of hiding a large amount of secret information in a cover image without incurring noticeable embedding distortion [51]. They both work by hiding message bits into the LSBs of pixel values of a cover image. The slight difference between them is that the first method simply replaces the LSB plane of image pixels with message bits, while the second method, at random, increments or decrements a pixel value when the message bit is different from its LSB, or otherwise keeps the pixel value unchanged. Based on the original LSB steganography, subsequent research effort has led to certain variants of the technique [52, 53].

LSB-based steganography has also been modified so that the message bits can be inserted into multiple bit planes rather than the least-significant bit plane for the purposes of higher embedding capacity and/or stronger robustness. A generalization of the LSB steganographic method has been proposed in [54] as a lossless (reversible) data-embedding technique. In the embedding phase, the lowest L levels of the original pixel values are replaced with the payload vector of L-ary symbols by quantizing the pixels at L level followed by watermark addition. During extraction, the watermark payload is extracted by simply reading the lowest L levels of the watermarked pixel values.

Uniform replacement of the lowest L levels of the original pixels with watermark payload could incur perceptual embedding distortions in some image regions, due to the characteristics of the HVS. That is, HVS has a non-linear response to luminance, so changes that are perceptually unnoticeable or undetectable in one image area could become visible in another area. That means that we would like to introduce varying amounts of embedding distortion over different image blocks, depending on the distortion tolerance, in order to avoid unacceptable quality degradation. As a result, researchers have proposed some adaptive steganographic approaches that take into account the HVS features. Yang et al. [55] have extended the original L least significant bits substitution scheme by using a changeable L rather than a fixed L. This approach allows one to adaptively overwrite the insignificant bits with message bits, thus ensuring that the resulting distortion is globally acceptable.

Instead of inserting the messages into the image pixels directly, the differences between the pixel values have been used to carry secret information as well. From another point of view, the difference can be considered as a kind of prediction error. In the pixel value differencing (PVD) steganography, a cover image is generally partitioned into non-overlapping and consecutive groups of several neighboring pixels. Tian [56] creates a set of pairs, each of which is composed of two neighboring pixels, and then embeds the data into the difference of the two pixels within each group. Actually, this embedding results in an expansion of the prediction difference, or error, between two pixels. The idea of difference expansion has greatly influenced the development of PVD-based steganography, and in particularly is widely regarded as a remarkable breakthrough in reversible data-hiding schemes where the original data and the embedded data can be completely restored after extraction.

Alattar [57] extends Tian's algorithm using difference expansion of N-sized vectors ($N > 2$), instead of 2-sized pairs, to increase the hiding ability and the computation efficiency. Kim et al. [58] improve Tian's algorithm in terms of the size of the location map indicating the locations of all pixel pairs that have been selected for difference expansion so as to convey message bits.

There exist some steganography operating in frequency domain rather than spatial domain. Quantization index modulation (QIM) is a commonly employed data embedding technique that embeds data into frequency coefficients. Noda et al. [59] have proposed two JPEG steganographic methods using QIM in the discrete cosine transform (DCT) domain. The two methods approximately preserve the histogram of quantized DCT coefficients, so that they are able to defend against histogram-based attacks.

2.1.2 Image Watermarking

In late 1990s, digital watermarking dominated the research in information hiding due to its wide range of practical applications in, for instance, digital rights management, secure media distribution and authentication [60]. Generally, the existing watermarking methods can be classified into two categories: *spatial-based* and *frequency-based*. The former type of approaches works by modifying the spatial data, i.e., the image pixel values of an image to embed watermark. The latter type operates by transforming a host image into frequency domain, followed by altering the frequency-based data to insert watermark.

Spatial-based Watermarking: One of the early image watermarking approaches working in the spatial domain is the work by Nikolaidis et al. [38]. They have presented an additive method that adds a positive integer to the pixels corresponding to the 1-valued pixels of a binary watermark pattern. The decision on whether the image is watermarked or not is carried out using hypothesis testing. Based on the least-squares (LS) prediction error sequence of the cover image, Karybali et al. [61] have proposed a spatial watermarking, which is robust against linear filtering and noise attack. The LS prediction error sequence matches quite well the characteristics of the HVS, in that the errors are expected to be smaller in smooth areas than in edges and textured areas.

Frequency-based Watermarking: For a good tradeoff between watermark robustness and invisibility, some image transformations, such as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) and Discrete Wavelet Transform (DWT), have been exploited in previous image watermarking.

Watermark embedding in the Fourier domain has the advantages of being robust against geometric attacks, such as scaling, rotation and translation etc. Nevertheless, there are few Fourier-based watermarking algorithms as yet. One early work is by Solachidis et al. [62], where the authors insert the watermark over the middle frequencies of the DFT image domain. The reasons of inserting watermark into the middle frequencies are based mainly on two considerations. On the one hand, modifications in the low frequencies of the Fourier transform are likely to cause visible changes in the spatial domain. On the other hand, image compression could remove the high Fourier frequencies. Hence, to survive compression and meanwhile maintain invisible, the watermark should be added in the middle frequency range. The watermarking in [40, 63] also modifies the middle Fourier frequencies so as to hide the watermark.

DCT domain watermarking can be classified into *global* DCT watermarking and *block* DCT watermarking. Cox et al. [14] have presented a global robust watermarking that embeds the watermark into the large coefficients of the DCT of a cover image. As the large coefficients represent more perceptually significant components of the image spectrum that likely survive after common image processing operations, such an embedding strategy can achieve higher robustness. Given the watermarked and the original images, the watermark can be retrieved from the DCT coefficients of the two images. The method is robust against various image operations, including image scaling, JPEG compression and dithering. However, it is a non-blind method, because it requires the availability of the original image during the extraction of watermark.

Another category of DCT watermarking is block-based, taking advantage of the local spatial correlation property of images. In block-based watermarking, the host image is generally divided into several non-overlapping blocks (the commonly used block size is 8×8) and then watermark is independently embedded into the DCT coefficients of each image block. Huang et al. [64] have introduced a block-based watermarking scheme in DCT domain, where spatial masking (both luminance and texture masking) is taken into account during embedding, making the watermarking an adaptive algorithm. The DC (Direct Current) coefficients rather than the AC (Alternating Current) ones are employed to carry watermark information for improved robustness of watermark. Several other block-based DCT watermarking algorithms can be found in [4,65].

DWT has been used in digital image watermarking more frequently as compared to DFT and DCT, due to its excellent spatial localization and multi-resolution characteristics, which are similar to the theoretical models of the HVS. Zhu et al. [66] have proposed proposes a unified approach to watermarking images and videos based on using the two- and three-dimensional DWT. This method adds the watermark to all high-pass bands in the wavelet domain, but it does not consider the features of the HVS. Kaewamnerd and Rao [67] improve the technique in [66], taking into account the HVS features. Ganic et al. [68] have proposed a hybrid watermarking technique based on DWT and Singular Value Decomposition (SVD). After decomposing the cover image into four bands, the SVD is applied to each band, followed by embedding the watermark data through modifying the singular values. By exploiting the significant difference of wavelet coefficient quantization, Lin et al. [69] have presented a blind watermarking algorithm. The maximum wavelet coefficient are quantized so that their significant difference exhibits a large energy difference which is used for watermark extraction.

Embedding into multiple frequency domains could have combined advantages derived from each single domain, so researchers have investigated and presented some interesting multiple domain watermarking techniques. Zhao et al. [70] have presented a dual domain watermarking technique for image authentication and compression. They use the DCT domain for watermark generation and DWT domain for watermark insertion. The embedded watermark is comprised of two components: a soft-authenticator watermark for authentication and tampering assessment of the given image, and a chrominance watermark for improved efficiency of compression. A DWT-DFT based image watermarking algorithm has been proposed in [71], where a spread-spectrum-based informative watermark is embedded in the coefficients of the LL subband in the DWT domain while a template is embedded in the middle frequency components in the DFT domain. Notice that after applying DWT to an image, we obtain a set of four subbands, namely, LL, LH, HL and HH. The LL subbands represent the approximated version of the original at smaller resolution. The LH blocks contain vertical edge components. Similarly, the HL and HH blocks contain horizontal and diagonal details, respectively.

2.2 3D Information Hiding

3D information hiding has received relatively less attention and thus fewer schemes have been reported for hiding data over 3D models, as compared to digital images. However, recent advances in 3D hardware, data acquisition and processing imply the advent of 3D as a mainstream communications medium over the next years. As pointed out in [72], some of the potential applications of 3D computer graphics models are in digital archives, entertainment, Web3D, MPEG4, and game industry.

The expected extensive use of 3D content in practical applications means that immediate research attention should be paid to 3D steganography and watermarking. In what follows, we shall review the recently proposed data embedding strategies for 3D models.

2.2.1 3D Steganography

Regarding 3D steganographic algorithms, they achieve embedding of information based either by modifying the geometry, or by changing the connectivity and topology, or on taking full advantage of the redundancy in the indexed representation of a host 3D model. Since steganography favors large embedding capacity, the majority of the existing 3D steganographic algorithms use as the message carrier the geometry rather than the connectivity or the representation which usually has lower embedding capacity.

Geometry-based Steganography: This type of 3D steganographic algorithm works by means of inserting messages into the geometry of a host 3D model. As mentioned before, most of the previous 3D steganographic schemes belong to this category.

The first data embedding technique for 3D geometric objects is by Ohbuchi et al. [73]. Ohbuchi et al. have described the background and requirements for 3D data hiding and then proposed two methods: triangle similarity quadruple embedding and tetrahedral volume ratio embedding.

Cayre et al. [5] have proposed a blind data hiding scheme. The algorithm consists of two steps: 1) establishing a list of triangles of the 3D mesh that will be used to carry message bits and 2) modifying each admissible triangle in the list according to the message bit it will carry. The principal idea behind the method is to consider each triangle as a two-state geometric object, that is, 0 or 1, depending on the position where the triangle summit vertex is orthogonally projected to its opposite edge. In order to insert a message bit, they move the projection of a vertex towards the nearest correct interval of its opposite edge. Inspired by the ideas by Benedens [74, 75], they deal with the synchronization problem by considering geometrical/topological properties and applying PCA. More specifically, the starting

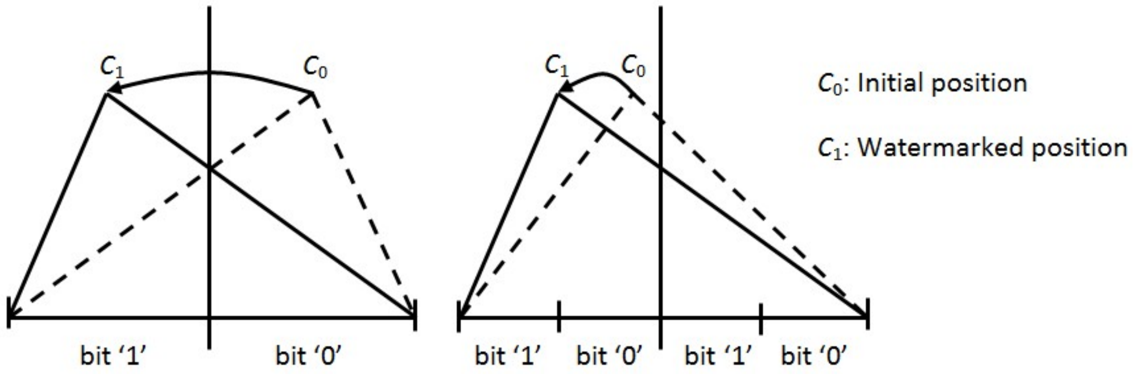


Figure 2.1: Illustration of Cayre et al.'s method [5] that embeds the message bit through moving the projection of a vertex towards the nearest correct interval. (a) the opposite edge is divided into two intervals and (b) the opposite edge is divided into four intervals. In both cases, the bits being embedded are all "1".

triangle in the triangle list is regarded as the triangle with lowest or greatest area. Alternatively, the initial triangle is one of the six triangles that intersect with the three principal axes centered at the gravity center of the host 3D mesh. Fig. 2.1 illustrates the embedding process. The use of PCA make the data hiding method robust against 3D content-preserving operations, but it offers weak robustness against malicious attacks, such as 3D simplification and re-meshing. In addition, the algorithm has other shortcomings. For instance, it is exclusively suitable for hiding data over 3D triangle meshes, but not appropriate for data embedding into other non-triangular meshes with arbitrary topology, because it can be hard to define the opposite edge of a vertex on a general polygonal mesh. Another drawback lies in its low embedding capacity, whose upper limit is 1 bit per vertex.

The work by Cayre et al. [5] has greatly influenced the development of 3D steganography. Based on Cayre et al.'s method, Wang et al. [11] have successfully increased the embedding capacity to be approximately 3 bits per vertex by using a multi-level embedding procedure. The improved algorithm decreases visual degradation since it takes into account the characteristics of HVS when executing message insertion. Moreover, the algorithm is able to carry out data embedding and extraction quickly, making it suitable for data embedding over large 3D datasets. Wang

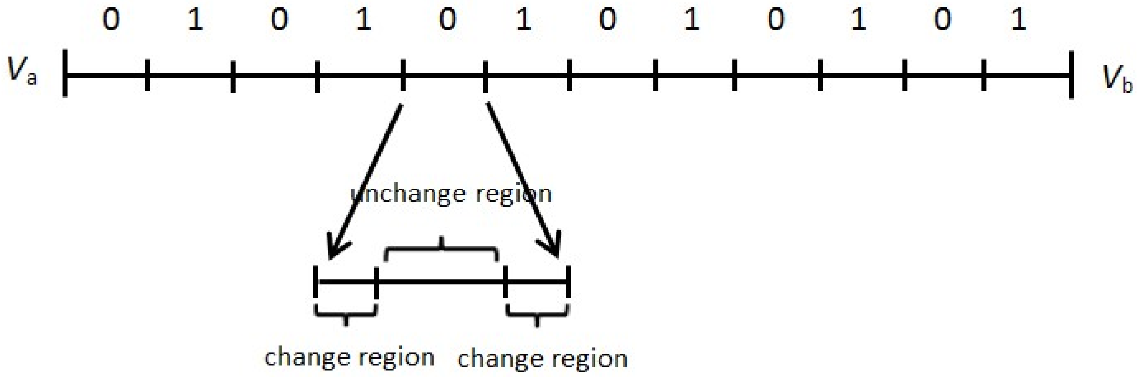


Figure 2.2: Illustration of Chao et al.'s state classification [6].

et al. have extended their work in [76], achieving even higher embedding capacity and lower distortion. The extension has taken into consideration the relationship between the HVS and the size of the payload in the phase of embedding, thus making it an adaptive algorithm. Their main observation is that it is easier to notice the changes on the smooth surfaces or regions than to observe the modifications on the rough surfaces. Based on this, it is reasonable to vary the embedding strength in different surface areas. For the embedder to estimate the degree of smoothness or roughness, the approach exploits the correlation between neighboring polygons with respect to the HVS. It is worth mentioning here that the method is the first 3D steganographic scheme that can achieve adaptive message embedding. Additionally, it is robust against affine transformations and it has high capacity and low distortion. The main limitation is its poor performance when inserting messages into models with a smaller number of vertices. This limitation is due to machine precision errors.

Instead of deeming each triangle as a two-state object, Chao et al. [6] construct a new coordinate system using the three principal axes obtained by applying PCA to the cover 3D model. Then, they uniformly divide the line segment, whose two extreme end points V_a and V_b are the furthest projections of the mesh vertices on the new x, y or z axis into a set of intervals and assign each segment a two-state object in an interleaved manner. Fig. 2.2 illustrates the state classification process on the line $\overrightarrow{V_aV_b}$. Next, each segment is further divided into *change region* and *un-change*

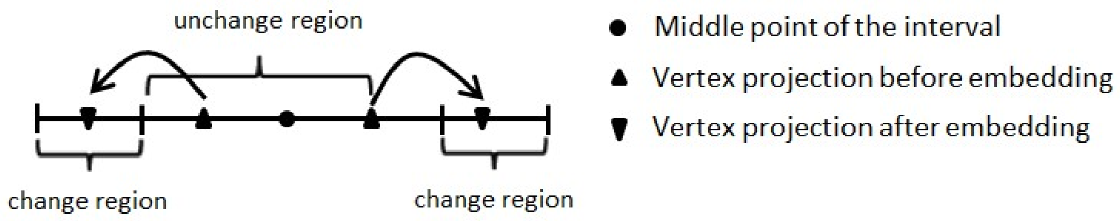

unchange region
change region
change region
● Middle point of the interval
▲ Vertex projection before embedding
▼ Vertex projection after embedding

Figure 2.3: Illustration of Chao et al.'s moving direction for embedding [6].

region and the embedding proceeds with either keeping the to-be-embedded vertex intact, or moving it into the change region of an interval, depending on the message bit to be inserted and the state of the interval into which the vertex is projected. To minimize distortion, the moving direction depends on the projected position on the interval. That is, if the projection is on the left-hand side of the middle of the interval, the vertex will be moved towards the left into the change region, and vice versa (see Fig. 2.3). The method achieves high embedding capacity with low embedding distortion, but it can weakly only resist similarity transform attacks and is unable to withstand malicious attacks.

Through the evaluation of the algorithm, the embedding capacity of 3D steganography has increased from nearly 0.5 bit per vertex in [77], to approximately 39 bits in [6] and about 50 bits per vertex in [10]. In particular, the steganography by Yang et al. [10], which will be presented in Chapter 4, appears to outperform any previous state-of-the-art steganographic methods in terms of embedding capacity and distortion control.

Other embedding primitives have also been successfully used to carry the messages. For instance, Wu et al. [78] have proposed an embedding algorithm that carries out information embedding through modifying

$$\mathbf{d_i} = \mathbf{v_i} - \mathbf{c_i} = (v_{ix} - c_{ix}, v_{iy} - c_{iy}, v_{iz} - c_{iz}) \tag{2.2.1}$$

with

$$\mathbf{c_i} = \frac{1}{N_i} \sum_{j=1}^{N_i} \mathbf{n}_i^j \tag{2.2.2}$$

where $\{\mathbf{n}_i^j \,|\, 1 \leq j \leq N_i\}$ is a set of N_i vertices $\mathbf{n}_i^j$ directly connected to the vertex

$\mathbf{v_i}$ and (v_{ix}, v_{iy}, v_{iz}) and (c_{ix}, c_{iy}, c_{iz}) are, respectively, the coordinates of $\mathbf{d_i}$ and $\mathbf{c_i}$ in $\mathbb{R}^3$. Wu et al. [79] have inserted the information into the distances from the mesh vertices to the centroids of their respective neighboring vertices. As those distances are invariant to similarity transformations, the method can withstand similarity transformations.

Topology-based Steganography: This category contains 3D steganographic algorithms based on modifying the connectivity or topological features of a host 3D model to embed data. Their main drawback is that the connectivity can carry limited information only. Presently, there are only a few 3D connectivity-based steganographic methods reported in the literature.

Mao et al. [80] have developed a method achieving data hiding through triangle subdivision. Specifically, they add a new vertex on each of the three edges of a triangle and then divide the triangle into four sub-triangles after connecting any two new vertices. This process is equivalent to replacing a longer triangle edge with two smaller edges. The ratio between the lengths of the two newly introduced edges has been used to carry a message bit. The approach is blind because it does require the availability of the original cover model during the message extraction process.

Using minimum spanning tree (MST) and connectivity modification, Amat et al. [81] have proposed a lossless 3D steganography. The method consists of three steps. The first step is to construct a MST that covers the vertices of the host 3D model. Next, the second step necessitates a scanning of the constructed MST to find and synchronize particular mesh areas for embedding the data. They have used PCA in the second step, which guarantees that the mesh reorganization does not disturb the location of the starting vertex and that the starting vertex is dependent only on the secret key. The final step involves the embedding of the message, which is carried out via modifying the connectivity of triangles in the selected areas. The method is distortion-free, in the sense that it does not modify the positions or coordinates of the mesh vertices during embedding.

Representation-based Steganography: This category of 3D steganographic algorithms carries out data hiding by exploiting the redundancy in the mesh representation. In general, representation-based steganographic methods are distortion-free, because they employ representation redundancy for embedding while keeping the geometry and connectivity of the 3D mesh intact. Similar to connectivity-based steganography, only few 3D steganographic techniques based on representation have been proposed. Moreover, representation-based steganography has a more limited scope than those embedding information on the mesh geometry.

Chen et al. [82] have presented a 3D polygonal steganographic approach that uses the representation information to embed messages. They embed messages by modifying the vertex representation order and the polygon representation order, with respect to the traversal orders. This method is lossless because changing the order of vertices and polygons arbitrarily in the vertex list and the polygonal list of the mesh file does not distort the mesh geometry.

Bogomjakov et al. [83] have presented a distortion-free steganography that hides a message in the indexed representation of a mesh by permuting the order in which faces and vertices are stored. The permutation is relative to a reference ordering that encoder and decoder derive from the mesh connectivity in a consistent manner. Although the permutation of vertices and faces does not affect the geometry of the mesh, it will affect the rendering performance. Tu et al. [84] have improved the efficiency in the original mapping of [83], further increasing the average embedding capacity up to 0.63 bits per primitive. The original mapping in [83] encodes interior nodes as well as their leaf nodes, so the prefixes of the corresponding bitstreams are represented twice. By contrast, Tu et al. have overcome this inefficiency by changing the mapping such that only leaf nodes are encoded.

2.2.2 3D Watermarking

Similarly to image watermarking, 3D mesh watermarking methods can be generally classified into two categories: *spatial-based* and *transform-based*. The methods in the first category achieve the embedding of the watermark through modifying the spatial information, such as the geometry and topology of a 3D cover model. The methods in the second category work by first transferring a 3D model into the spectral domain and then inserting the watermark into some frequency-based primitives.

Spatial-based Watermarking: During the early stages of the development of

3D watermarking, fragile watermarking schemes have been proposed for the purposes of authentication and attack localization. In general, a fragile watermarking should have the following two properties. The first one is that it should be sensitive even to slight modifications. The second one is that it should be able to locate and identify the endured attacks. Although fragile watermarking techniques are not required to be robust against malicious attacks, it is also expected that they are able to survive the content-preserving operations. As stated in [85], two problems frequently arise in the embedding stage: *causality* and *convergence*. The causality problem arises while the neighboring relationship of a formerly processed vertex is influenced by the perturbation of the subsequently processed neighboring vertices. It results in the extracted bits being different from the original ones, even in the absence of attacks. The convergence problem means that the original model may have to be heavily distorted before some vertices reach a predefined relationship.

In one of the earliest approaches, Yeo et al. [86] have first proposed a fragile watermarking method to verify 3D meshes. The watermarking method perturbs each vertex to ensure that two predefined hash functions have the same value on it. Verification is achieved by a comparison of the values computed by the two hash functions. One drawback of this method is the causality problem, caused by its heavy dependence on the order of traversal of the vertices. The proposed scheme is not able to localize the changes or distinguish malicious attacks from incidental data processing. The method has both the causality and convergence problems.

To trade off the causality problem in Yeo et al.'s method, Lin et al. [87] have introduced a new fragile watermarking that is immune to certain shape-preserving data processing. Their method copes with the causality problem by applying two different hash functions on the vertex coordinates, without taking into account the neighbors of a vertex. The method is independent of the order of vertices, so the hidden watermark is immune to vertex order-dependent attacks, such as vertex reordering. A 3D mesh authenticating watermarking scheme has been proposed by Wu et al. in [88]. Their method embeds the watermark bits by adjusting the positions of the centroids of the mesh faces. The watermark embedded with this method survives translation, rotation and uniform scaling, but it is sensitive to other operations.

Chou [85] have overcome the causality and convergence problems by introducing a multi-function vertex embedding method and a vertex-adjusting method. In the former method, the three coordinate components of a marked vertex are assigned and embedded with different functions of watermark in the watermark embedding stage. The x coordinate of a vertex is modulated to indicate if it is a marked vertex, while the y and z coordinates are respectively used to carry the watermark information and the hash value of the watermark. The latter method aims at overcoming the afore-mentioned two problems, by means of keeping the barycenters of the marked vertices unchanged. Another feature of the method is its ability to control the average distortion as a result of the watermark embedding.

Recently, a semi-fragile watermarking algorithm for the authentication of 3D models has been proposed by Wang et al. [89] based on integral invariants. For embedding, the method modifies the integral invariants of some of the vertices through shifting the positions of a vertex and its neighbors. It can survive under rigid transforms and certain noise attacks. More 3D fragile watermarking algorithms can be found in [90, 91].

As mentioned earlier, the literature is primarily concerned with the robustness of watermark. To increase robustness, rather than inserting the watermark into a single vertex, Yu et al. [92] have proposed to embed it into a group of mesh vertices. Specifically, they scramble and divide the vertices into a set of groups according to a secure key. Each group carries only one watermark bit. The watermark bit is inserted with a simple additive method that modifies the distances between the vertices of each group to the center of the model. The watermark strength varies taking into account the local feature of the mesh, thus making the embedded watermark less visible, but more robust. The watermarking is non-blind since the original 3D model is required during watermark extraction. It is worth mentioning here that the method is the very first attempt to insert a watermark in a global and essentially geometric characteristic of a 3D mesh; the characteristic used here is the distances from the mesh vertices to the mesh center.

Since the Cartesian coordinates (x, y, z) appear to be sensitive under attacks,

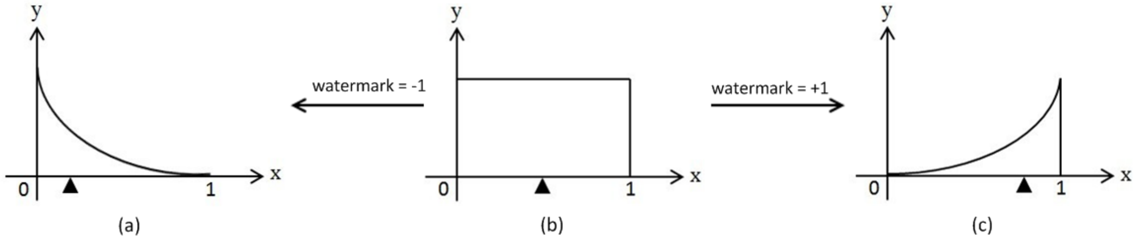


Figure 2.4: Illustration of Cho et al.'s watermarking [7] that embeds the watermark through modifying the mean value of the histogram in a bin: (a) the mean value is decreased to embed a bit "-1"; (b) the assumed uniform distribution in a bin; and (c) the mean value is increased to embed a bit "+1". The x- and y- axes stand for the normalized distances from vertices to the mesh barycenter and the occurrence probability, respectively.

some researchers have recently attempted to insert watermarks over the spherical coordinates (ρ, θ, ϕ). The ρ is the radial component, representing the Euclidean distance from a mesh vertex to the barycenter of the 3D model. Several properties of ρ make it a promising candidate as watermark carrier. One of them is its invariance under various operations, such as 3D rotation and translation. A second property is that the embedding of ρ is supposed to be robust, in the sense that the values of ρ approximately represent the shape of the 3D mesh.

Cho et al. [7] have proposed two watermarking algorithms based on modifying the radial component ρ of the spherical coordinates. The basic ideas of two methods are almost the same, but the difference is that the first changes the mean of ρ in one group, while the other alters the variance. Both build a histogram using $\{\rho_i | 1 \leq i \leq N\}$, where ρ_i is the Euclidean distance from the i-th vertex to the mesh barycenter and N is the number of vertices of a host 3D model. Then, they normalize all the ρ_i in each histogram bin, so that the normalized ρ'_i is $0 \leq \rho'_i \leq 1$. The embedding of watermark is based on the assumption that the mean of the normalized distances ρ'_i in each bin is approximately 1/2 and the variance is about 1/3. In other words, they increase or decrease ρ'_i depending on the value of watermark bit to be embedded, making the mean (or variance) of the updated ρ'_i in each bin smaller or

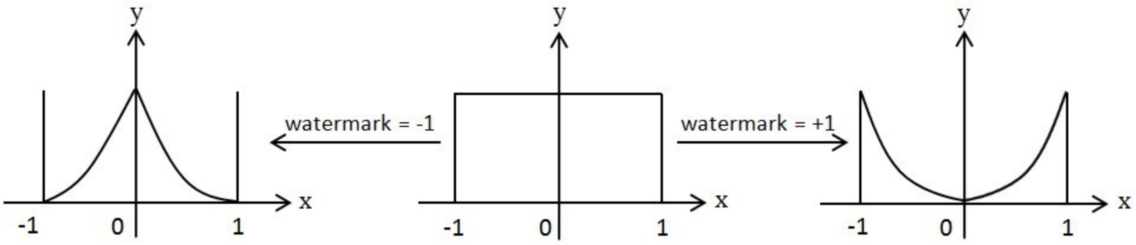


Figure 2.5: Illustration of Cho et al.'s watermarking [7] that embeds the watermark through modifying the variance of the histogram in a bin: (a) the variance is decreased to embed a bit "-1"; (b) the assumed uniform distribution in a bin; and (c) the variance is increased to embed a bit "+1". The x- and y- axes stand for the normalized distances from vertices to the mesh gravity center and the occurrence probability, respectively.

greater than 1/2 (or 1/3). Fig. 2.4 and 2.5 illustrate the watermarking processes for the two approaches.

Similarly, Zafeiriou et al. [12] have proposed two watermarking schemes in the spherical coordinate system. Before watermark embedding, they calculate the barycenter, then perform principal axis alignment using PCA and convert the 3D model from the Cartesian coordinate system into the spherical coordinate. The watermark embedding is carried out by altering the radial component ρ.

As demonstrated by the experimental results, the watermarking schemes in [7, 12] are resistant to a number of both non-malicious and malicious attacks. Other methods that exploit spherical coordinates as watermark carriers include those by Ashourian et al. [93] and Darazi et al. [94].

Regarding other spatial-based robust watermarking techniques that modify the geometry for watermark insertion, some are particularly worthy of being mentioned here, even though they could have relatively weaker robustness. Bors [95] has proposed two blind watermarking algorithms that use a neighborhood localized measure to find vertices that give small embedding distortion and then watermarks these vertices by local geometric perturbations. Unlike most conventional 3D object watermarking techniques, where both watermark insertion and extraction are per-

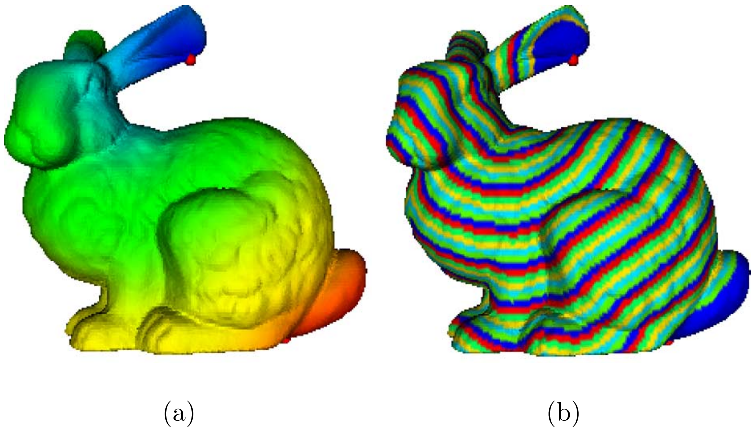


Figure 2.6: The geodesic distances, computed on the *Bunny* object with respect to the single source point (indicated by a small red circle on *Bunny* ear), are highlighted using pseudo-color. (a) the Geodesic map, where the pseudo-color varies from blue to red, according to the geodesic distance. (b) Iso-geodesic mesh strip generation. Each trip defines the region of almost equal geodesic distances and is used for embedding a single bit.

formed on the 3D object itself, Bennour et al. [96] have proposed a new framework for watermarking 3D objects via their contour information. More recently, Luo et al. [97] have presented a new statistical approach that watermarks geodesic distances calculated with respect to a reference location on the mesh. The embedding and extraction processes are very similar to those proposed in [7]. The principal idea behind this type of watermarking is that it changes the mean or variance of the geodesic distances that are grouped into each region. Fig. 2.6 shows the geodesic map, which was computed on the *Bunny* model, as well as the regions or trips, each of which contains almost the same geodesic distances.

A number of 3D watermarking approaches succeed in resisting common processing operations, such as smoothing and noise addition. However, they are unable to survive mesh editing operations, which have been recently considered as being more realistic than other operations. Here, mesh editing means an operation that alters the global shape of the 3D model. An example showing the mesh editing is illustrated in Fig. 2.7.

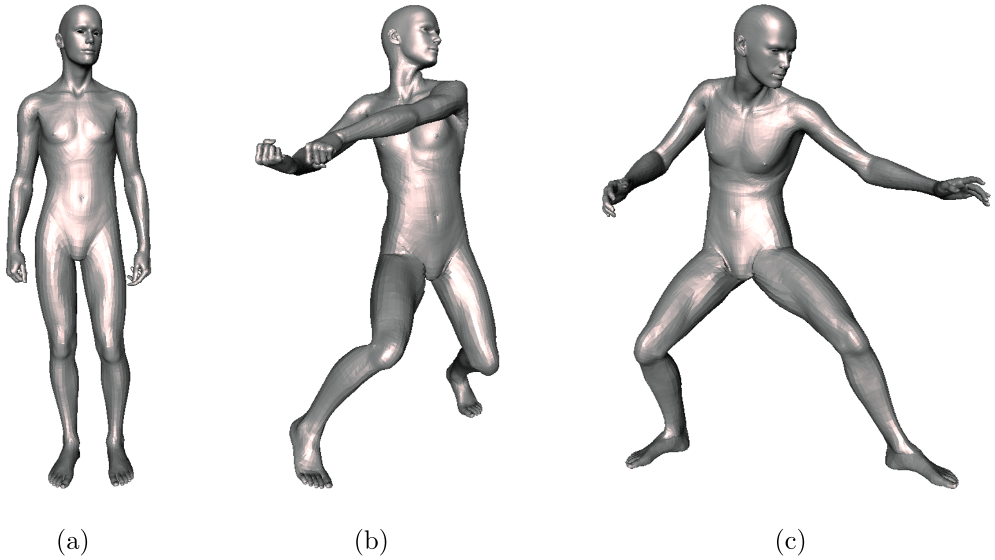


Figure 2.7: Illustration of the *David* model editing. (a) Original model, (b) and (c) edited versions. The models are from the TOSCA database [8].

Considering the situation above, researchers have recently proposed some watermarking schemes that are robust against mesh editing. One of them is by Lin et al. [98], who embed watermark over the significant mesh patches with high curvature values. Embedding into those regions minimizes the perceptible distortion, because the HVS is more sensitive to changes in the low-curvature regions. Instead of using Euclidean distances, geodesic distances are used, which would not be altered significantly under mesh editing. This watermarking can survive a range of attacks, but offers week robustness to non-uniform scaling and shearing. Moreover, it is a non-blind method that requires some side information for watermark extraction.

Transform-based Watermarking: Another category of robust 3D mesh watermarking methods uses spectral analysis to achieve significant improvements in both watermark robustness and imperceptibility according to the spread spectrum communication principle [14]. Multi-resolution analysis, such as DFT, DCT and DWT, has been widely utilized in early image watermarking with the watermark inserted into the coefficients obtained by applying these transformations to the host image. Nevertheless, an efficient and robust spectral analysis tool has not been proposed for 3D models as yet, and moreover, there is still a lack of a natural parametrization for frequency-based decompositions of arbitrary 3D models. As a result, generalizing the spread spectrum-based watermarking approach to the case of arbitrary triangle meshes is indeed an acute challenge [99], and it is some recent research work in analyzing meshes that has resulted in multi-resolution surface representations possessing similar properties as traditional wavelet transforms.

By taking advantage of the edge collapse-based multi-resolution decomposition proposed by Hoppe [100], Praun et al. [50] have presented a robust, non-blind 3D watermarking. A multi-resolution set of scalar basis function over the original mesh is constructed after applying Hoppe's method. The basis functions that correspond to large perceptually significant features of the 3D model have been used to carry the watermark enhancing the watermark robustness. This embedding idea is similar to image watermarking approaches, where, for example, DCT basis functions with large amplitudes have often been employed as watermark carriers. This method has been shown to be robust against various real-world attacks; however, it is a non-blind method, requiring 3D mesh registration and re-sampling.

Based on subdivision surfaces, Lounsbery et al. [101] have established a theoretical basis for applying wavelet analysis to surfaces of arbitrary topological type. Using this wavelet analysis, Kanai et al. [102] proposed a non-blind watermarking method by modifying the wavelet coefficients to embed watermark, while Uccheddu et al. [103] have extended this non-blind approach to be a blind one. One constraint to these methods is, however, that the input 3D mesh should be semi-regular.

Using the wavelet analysis scheme [104] for irregular meshes, Kim et al. [105] have designed a watermarking technique that can be applied to both regular and irregular 3D triangular meshes. In this proposal, the watermark is embedded by modifying the L^2-norm of the wavelet coefficients.

In addition to robust watermarking, wavelet multi-resolution analysis has been employed to design fragile watermarking algorithms. In order to authenticate semi-regular meshes, Cho et al. [106] have proposed a wavelet domain-based fragile method, where the facets in the coarser mesh, which is obtained from applying wavelet decompositions to the original triangular mesh, are taken as authentication primitives. This method can withstand similarity transformations. However, as pointed out in [99], it may have two drawbacks: causality problem and weak localization capability. Wang et al. [107] have embedded the authentication watermark by slightly modifying the norms and orientations of the wavelet coefficient vectors. Their method is robust to the content-preserving attacks, but vulnerable to others attacks such as local and global geometric modifications and re-meshing.

Several other multi-resolution analysis tools have been utilized to design robust 3D mesh watermarking algorithms too. Yin et al. [108] first construct a Burt-Adelson pyramid using [109], followed by the watermark insertion into a suitable coarser mesh representation. Registration and re-sampling are required during watermark detection to bring the attacked 3D mesh model back into its original location, orientation, scale, topology and resolution level, thus making this method non-blind.

Several frequency-based watermarking methods have been proposed. Ohbuchi et al. [13] have proposed a frequency-based robust watermarking method. This method is considered a classic in the frequency-based watermarking. Before watermark embedding, the original cover mesh is divided into several patches, each of which is generated around a seed vertex by incrementally adding all the neighboring vertices within a given topological distance from the seed vertex. The distance used in this method is computed by the standard Dijkstra's algorithm. After patch generation, the watermark is embedded via additive modulation of the spectral coefficients of each patch, which are computed from the *Kirchhoff* matrix [110]. To further improve robustness, each watermark bit is repeatedly inserted. The method is able to survive various attacks, including cropping, mesh simplification and smoothing. It is a non-blind method and thus requires the original mesh at the stage of watermark extraction. Notice that Taubin et al. [111]'s spectral decomposition has also been used by Karni et al. [112] for geometry compression.

Cayre et al. [113] have applied this spectral analysis to the watermarking of 3D triangle mesh geometry. The embedding of the watermark is achieved through modifying the spectral coefficients, obtained from the *Laplacian* matrix. More recently, Abdallah et al. [114] have presented a spectral graph-theoretic approach to 3D mesh watermarking, where they partition the host 3D mesh into a set of smaller sub-meshes, compress each sub-mesh using spectral compression [112] and finally modify the spectral coefficients of each compressed sub-mesh to insert the watermark.

One obvious shortcoming of spectral analysis is its high computational complexity, since it involves the computation of the eigenvalues and eigenvectors of a probably large *Laplacian* or *Kirchhoff* matrix. This computation is extremely time-consuming, especially for large 3D models. To settle this issue, the researchers have proposed to divide the original 3D mesh into several mesh segments and then insert watermarks into each segment.

Other spectral algorithms that have been proposed and used for robust 3D watermarking include Wu et al. [115]. There, they have presented a fast watermarking algorithm based on the orthogonalization of a small set of radial basis functions, trying to avoid the complicated numerical computations of previous methods during multiresolution or spectral mesh analysis. This method can efficiently watermark very large 3D models.

Yang et al. [9] have proposed a robust watermarking algorithm, aiming at the robustness against mesh editing. The method will be presented in Chapter 7.

2.3 Video, Audio and Text Information Hiding

This section will provide a brief review of the information embedding techniques for video, audio and text. We start introducing information hiding schemes for video, then for audio and finally for text.

Video Watermarking: In general, video signals have large amounts of data and inherent redundancy between frames, that is, many successive frames are visually similar to each other. These properties result in video signals being highly susceptible to pirate attacks, such as frame averaging, frame dropping and interpolation, which are not applicable to digital images. Therefore, video watermarking is not a simple extension of still image watermarking and designing a video watermarking that is able to survive a wide range of practical attacks is a particularly challenging task.

Hartung et al. have first launched the research on video watermarking and proposed two methods for raw and compressed videos [116]. In the case of raw video watermarking, after modulating a binary watermark sequence with a binary pseudo-noise sequence, they add the modulated signal to the line-scanned digital video signal, yielding a watermarked video signal. In the case of compressed video watermarking where the video data is represented in a bitstream, they obtain a set of 64-sized vectors after applying DCT to the 8×8 blocks of the watermark and doing a zig-zag-scan to the DCT coefficients of each watermark block. Then, they perform an additive operation, adding each 64-sized vector to its corresponding 64-sized vector of DCT coefficients of each 8×8 video block, obtaining a marked video. The two approaches are, in general, sensitive to frame cutting and exchange.

Swanson et al. [117] have presented an object-based watermarking procedure to embed copyright protection into raw video sequences. They extract objects from the given host video and embed identical or different watermarks into visually similar and distinct objects, respectively. In particular, each similar object is associated with a watermark. As a result, the watermark for each frame changes according to perceptual characteristics, while simultaneously protecting objects against statistical analysis and averaging. In [118], they have extended their work by adding the wavelet transform to the previous embedding framework. To be more precise, instead of working on the original video frames [117], they apply wavelet transform to the original frames, producing wavelet coefficient video frames, and embed the watermark into the resulting wavelet-based frames. Both methods are blind, but one common drawback is that they consume additional storage space to maintain an object database that is composed of the objects extracted from the videos. The database is later used for determining the presence of watermark. In other words, given a potentially pirated video sequence, they extract an object from it, obtain a difference signal by subtracting the object from its corresponding similar object from the database and finally draw a watermark existence decision based on the similarity between the extracted signal and the original watermark associated with the object from the database. Another interesting object-based video watermarking has been proposed by Bas et al. [119], where the authors use the PCA to orientate the signature, taking into account object manipulations, such as rotations, translations and Video Object Plane (VOL) modifications.

There exist some algorithms that only watermark specific video formats, for instance, MPEG-4 watermarking in [120,121] and H. 264 watermarking in [122,123]. A survey regarding the current approaches for video content protection systems and recent advances in the tools and methods for providing security in such systems can be found in [124,125].

Audio Watermarking: Since the early development stage of audio watermarking, spatial/temporal and frequency masking approaches have been considered for audio watermarking algorithms, aiming at the inaudibility of the embedded watermark.

Tilki et al. [126] have presented a method of encoding a hidden digital signature of about 35 bits in length into the audio component of a television signal. Based on the fact that humans are much more sensitive to the lower frequencies, they add the watermark signal to the Fourier transform coefficients over the middle frequency bands. Aiming at watermarks that are both inaudible and robust, properties of the human auditory system (HAS) have been exploited by Swanson et al. [127]. The method computes the temporal masking and the Fourier-based frequency masking and uses the two maskings together with the author signature to create a watermark. The watermark signal is subsequently added to the original audio signal, producing the watermarked audio signal. The watermark detection requires the availability of the original audio and is accomplished via hypothesis testing. In other words, the watermark-like signal extracted from a received audio signal is measured against the original watermark with respect to a threshold that defines their similarity.

Similarly to image watermarking, DCT and DWT have been successfully used for audio watermarking too. After applying DCT to the audio signal, Yeo et al. [128] insert watermark by modifying the resulting DCT coefficients. Wang et al. [129] have proposed a synchronization invariant audio watermarking, based on both DWT and DCT. To avoid the synchronization problem, a synchronization code, in addition to a binary watermark image, is inserted into the audio signal. That is, they divide each segment of a host audio into two sections; one is used to carry the synchronization code, while the other is employed to convey the real watermark signal. Another interesting self-synchronized watermarking scheme has been introduced in [130], where the authors combine the synchronization codes and watermark together into a binary data sequence and then insert the combined binary string into the low-frequency DWT coefficients of each audio segment. The combination used in the method enables the hidden data to be equipped with the self-synchronization ability.

There are several other excellent audio watermarking approaches reported in the literature. Lee et al. [131] use a technique analogous to spread spectrum communications to insert the watermark in the cepstral domain of the host audio signal, while Bassia et al. [48] embed the watermark in the time domain.

Text Watermarking: Thus far, there have been relatively fewer works on text watermarking compared to images and videos watermarking.

Brassil et al. [132] have proposed the line-shift and word-shift text watermarking algorithms. The first method works by moving the second line up or down depending on the binary signal to be inserted and keeping the line immediately above or below (or both) intact. By contrast, the second method performs embedding by moving a word left or right and keeping the immediately adjacent words unmoved. In both approaches, the unmoved components serve as reference locations in the decoding process.

Atallah et al. [133] have proposed the idea of using the semantics and syntax of the text for inserting the watermark and also a watermarking algorithm that uses the binary encodings of the words to insert information into the text through performing lexical substitution in synonym sets. Later, they have presented another two algorithms [134, 135], where the information is inserted into the tree structure of the text, rather than directly into the text. The difference between the two algorithms is that the first one modifies syntactic parse trees of the cover text sentences, while the second one uses semantic tree representations. The selection of sentences that carry the watermark depends only on the tree structure. After selecting the sentences to embed the watermark, the bits are stored by applying either syntactic or semantic transformations. Semantic transformations in that work were designed for preserving the meaning of the overall text, but not necessarily preserving the meaning of every individual sentence.

There are some watermarking approaches aiming at inserting watermark into language-specific texts. For instance, Sun et al. [136] and Liu et al. [137] concentrate on watermarking Chinese texts, while Meral [138] focus on marking Turkish texts.

2.4 Image Steganalysis

It is not surprising that the advancements in steganography and watermarking have impelled the development of their counterpart, steganalysis. Thus far, numerous excellent steganalysis algorithms have been proposed, and they can generally be classified into two categories: *embedding specific* and *universal*. The former aim at the detection of a message associated with particular steganographic methods, while the latter aim at the message detection regardless of the embedding algorithm.

2.4.1 Specific Steganalysis

Specific steganalysis is likely to achieve higher detection accuracy than universal steganalysis because specific methods concentrate on the identification and extraction of those patterns that are sensitive to a specific or known embedding algorithm only.

Among various image embedding techniques, least-significant bit (LSB) steganography, including LSB replacement and matching embedding, is of great interest to steganalysis experts. Fridrich et al. [139] have introduced a powerful steganalytic method for detection of LSB replacement. The method is based on the observation that the LSB plane in a typical cover image can be predicted to some extent from the remaining 7 bit-planes, while this prediction becomes less reliable as the LSB is randomized. The experimental results show that an upper bound for safe LSB embedding is 0.005 bits per sample for high-quality images and that any LSB-based hiding method with embedding rate above that bound could be easily detected. By modeling a finite state machine whose states are selected multisets of sample pairs called trace multisets, Dumitrescu et al. [140] have presented an alternative approach to detecting least LSB replacement steganography.

Thus far, fewer detection methods for LSB matching have been proposed. Among them, one well-known steganalytic algorithm is by Harmsen et al. [141], who have proposed using the center of mass (COM) of the histogram characteristic function (HCF). Ker [142] has achieved higher detection rate by using down-sampled image for calibration and by using the adjacency histogram instead of the usual histogram.

Most of the steganalysis schemes against LSB steganography work only for detecting information hidden in the least-significant bit plane of an image. Yu et al. [143] have designed a image steganalysis approach that can detect the message bits embedded not only in the least-significant bit plane, but also in other less significant bit planes.

There exist some steganalysis approaches aiming at other than LSB steganographic methods. For instance, Fridrich et al. [144] have been focused on the JPEG steganographic algorithm F5. The key element of their method is the estimation of the cover-image histogram from the stego-image, which is done by decompressing the stego-image, cropping it by four pixels in both directions to remove the quantization artifact in the frequency domain, and re-compressing it using the same quality factor as the stego-image. Li et al. [145] have concentrated on a recent steganographic algorithm, i.e., the YASS algorithm, which was designed to resist blind steganalysis via embedding data in randomized locations. Their steganalysis is based on the observation that the embedding locations chosen by the YASS embedding scheme are not randomized enough, making the YASS detectable.

2.4.2 Universal Steganalysis

A major challenge in developing universal steganalytic algorithms is the identification of the features that are modified by watermark embedding. Learning-based steganalysis has been demonstrated as a promising strategy that can ensure universality. Generally, universal steganalysis follows a two-step framework. The first step involves feature training to obtain a classifier, where the feature data to be trained are extracted from a training dataset of both marked and unmarked images. The second stage consists of computing the features from a test image within a test dataset, applying the classifier produced in the first phase to the feature vector and finally making a steganalytic decision.

Farid [30] has proposed a universal approach that uses a wavelet-like decomposition to build higher-order statistical models of natural images. For each image, Farid computes the mean, variance, skewness and kurtosis of the wavelet coefficients and cross-subband prediction errors of wavelet coefficients and creates a $24(n-1)$-sized feature vector by collecting these statistics. Here, n is the level of wavelet decomposition applied to images. Fisher linear discriminant analysis (FLD), a class specific method for pattern recognition, is applied to the feature vectors of a training set of images with and without hidden messages, resulting in a classifier that distinguishes between unmarked and marked images. Once the classifier has been computed, the steganalysis process is straightforward. Given a test image, we just compute its feature vector, apply the classifier to the feature vector and assign it to the marked or unmarked category. Lyu et al. [146] have proposed a steganalytic algorithm using higher-order image statistics. They have extended [30] by extracting a $72(n-1)$-dimensional feature vector from each image and employing support vector machines (SVM) rather than FLD for building the classifier.

Rather than using empirical probability density function (PDF) moments as in [30], Xuan et al. [147] use the empirical characteristic function (CF) moments of the wavelet characteristic functions for steganalysis. Wang et al. [23] extract features from wavelet coefficients and use the informative features from empirical PDF and CF moments of subband images for universal steganalysis. These two kinds of moments have been widely used as features in the steganalysis approaches, but coefficients from other domains have also been used. Lie et al. [148] analyze the statistical properties of the spatial and DCT domains to determine the existence of hidden messages in an image.

2.5 Summary

By searching the literature we realized that several excellent data hiding and steganalysis algorithms have been proposed. Most of them have concentrated on text, image, audio or video and relatively little work has been reported on 3D models, despite their increased prevalence in practical applications. We also notice that despite the sophistication of the existing text/image/audio/video data embedding and detection techniques, they cannot be applied to mark 3D objects directly. We concluded that research into 3D information embedding and detection can be timely and have immediate practical implications.

In addition, most of the 3D existing steganographic approaches have low embedding capacity and are not able to control the embedding distortion resulting from steganography, which could cause unpleasant visual quantity degradation. Regarding 3D watermarking, some previous watermarking algorithms have been demonstrated to be robust against common attacks, such as geometric transformations, noise addition and mesh smoothing, but they are not resistant to more realistic processing attacks, for instance, mesh editing attacks.

As for steganalysis, a number of steganalysis algorithms have been proposed for detecting the hidden message in digital images; however, to the best of our knowledge, there is no steganalytic algorithm yet reported for 3D computer graphics models.

Given the above observations, the main motivation for this thesis is the development of 3D information hiding and steganalysis techniques, introducing new insights into how the embedding distortion and embedding capacity are related, and proposing novel algorithms.

Chapter 3

Normal Degradation of Triangle Meshes: An Empirical Study

As any digital information, the vertex coordinates of a triangle mesh can be seen as real numbers quantized at a level l, with typical values $l = 32$ bits (floats), or $l = 64$ bits (doubles). The choice of the appropriate l is a trade-off between efficiency and quality. A small l may lead to a significant loss of geometric information while, on the other hand, a large l may lead to mesh representations with a lot of redundancy, resulting to unnecessarily large files and consequently, to unnecessarily expensive computations.

Modifications of these discrete coordinates, as a result for example of applying a steganographic algorithm, can change the appearance of the rendered mesh in two ways. Either directly, in the form of a spatial perturbation of the vertices, or indirectly, by changing triangle normals used by the rendering algorithm. Generally, the induced normal perturbations are much more distractive to the human eye than the spatial perturbations, see Fig. 3.1. That means that although most mesh processing operations manipulate spatial information, we are mainly interested in the indirect effects the spatial manipulation has on the normal information.

This chapter studies empirically the effect of modifying the coordinates of the vertices of 3D triangle meshes on the face normals. More specifically, we propose a logistic model to predict the degradation of the face normals as the level of quantization decreases. The mesh is degraded by the randomization of each vertex coordinate

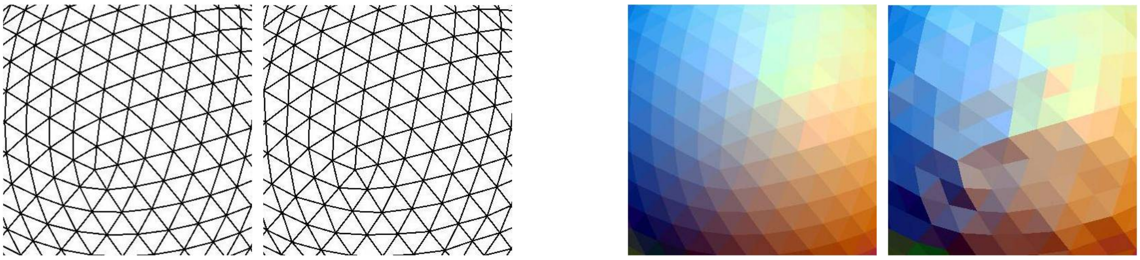


Figure 3.1: **Left:** The wireframe renderings of a smooth and a noisy mesh. **Right:** The flat shaded renderings of the same meshes.

after its t-th significant bit. Notice that we assume that any coordinate bit after the chosen level of quantization l has a random value, which is equivalent to applying to them a dithered quantizer rather than the commonly used quantizer that puts any less significant bit to zero. The normal degradation is computed as a weighted average of the angle differences between the normals of the original triangles and the corresponding degraded triangles.

The main contributions of this chapter are:

- A logistic model describing the degradation of the normal information of a triangle mesh as the quantization level decreases.
- A method for computing an appropriate level of mesh quantization when a tolerance for the accuracy of the normals is given.

The main limitation of our approach is the assumption that our meshes have no significant amount of noise. We make this assumption implicitly, by regarding the normals at the highest level of quantization as the most accurate. Thus, even though geometric noise can be estimated [149], and the effects of noise at different levels of mesh quantization have been empirically studied [150], this chapter focuses on the effects of quantization on clean, high quality meshes.

As an application of the proposed normal degradation model, we will compute appropriate levels l of quantization, given a tolerance for the average accuracy of a triangle normal, possibly weighted by geometric characteristics of the triangle, such as its area, or its dihedral angles. Finally, we will demonstrate by several examples that the claimed optimization is visually meaningful.

The rest of this chapter is organized as follows. Section 3.1 reviews the previous related work. In Section 3.2, we describe a logistic model for normal degradation and show how it can be empirically computed for a given triangle mesh. In Section 3.3, we experimentally validate the proposed model. In Section 3.4, we use the degradation model to compute appropriate quantizations for triangle meshes and experimentally show that the claimed optimization is visually meaningful. We briefly conclude in Section 3.5. Material in this chapter has been published in [151].

3.1 Prior Work

The problem of finding appropriate quantization levels for a mesh has been encountered in the classic predictive mesh compression algorithms of Touma and Gotsman [152] and Alliez and Desbrun [153]. In predictive encoding, the importance of removing redundant bits is further magnified by the fact that the predictions of the least significant bits are less accurate and thus more difficult to compress. Nevertheless, in all existing work the choice of quantization level is left to the user.

Face or vertex normals are used by most rendering algorithms, from the classic Gouraud and Phong algorithms, to the more computationally intensive BRDF based rendering by Walter et al. [154]. Vertex normals can be computed from face normals in various ways, typically as a weighted mean of the face normals, such as Jin et al. [155]. If instead the vertex normals are separately encoded as a part of the mesh file, they are usually represented by vectors with three 32 bit coordinates. Meyer et al. [156] study the quantization error introduced by such representations and propose efficient normal encoding methods.

The randomization of the least significant bits of the vertex coordinates adds a high frequency stochastic component to the geometry of the mesh. Uncertainty in polygonal meshes has been studied by Pauly et al. [157] and Kalaiah and Varshney [158]. Yoon et al. [149] propose methods for noise estimations on 3D point sets, while Sun et al. [159] study laser scan noise. In [160], Sorkine et al. discuss the visual impact of the high frequency noise introduced by spatial quantization and an alternative quantization method based on the mesh Laplacian is proposed.

As Schuchman [161] and Gray and Stockman [162] show, dithering is a technique with strong theoretical foundations and is commonly used in audio and image processing, see for example Roads [163] and Akarun et al. [164]. The purpose of dithering is to avoid coarse quantization artifacts, that is, unwanted regular patterns that may distract the eye or the ear.

The choice of quantization level for the vertex coordinates can be seen as a choice of level-of-detail. The low level quantizations correspond to coarse meshes with few triangles and the high level quantizations, which contain more geometric detail, correspond to fine meshes. In multi-resolution techniques, the level-of-detail of a mesh can be controlled either by a subdivision algorithm, see for example Kobbelt et al. [165] and Guskov et al. [166], or by a sequence of unitary mesh editing operations, such as edge collapses. The latter approach has been successfully applied into adapting the resolution of a mesh to the camera view, see for example Hoppe [167], Pasman and Jansen [168] and Hu et al. [169].

3.2 A Logistic Model for Normal Degradation

Given a triangle mesh $\mathcal{M}$, we model the quality of its normal information as a function of the level of quantization t by

$$\mathsf{D}_{\mathcal{M}}(t) = C/(1 + e^{-a-bt}), \quad t \geq 0. \tag{3.2.1}$$

$\mathsf{D}_{\mathcal{M}}$ is the expected average change of the triangle normals, possibly weighted by geometric characteristics of the triangles, when the t most significant bits of each vertex coordinate are retained and the less significant bits are randomized. Under our assumption of a clean, high quality original mesh $\mathcal{M}$, $\mathsf{D}_{\mathcal{M}}$ is seen as the normal error resulting from the vertex quantization. We notice that we can see Eq. 3.2.1 as an abstract degradation model and think of t as a real number; however, in practice, t represents bit positions and thus we are interested in the integer values of t between 0 and 64.

The curve of Eq. 3.2.1 has an inverse 'S' shape. Small values of t correspond to coarse quantizations and large expected normal error, while large values of t correspond to fine quantizations and small expected normal error. The exact shape of the curve depends on the three constants a, b and C. The value of a represents a quantization threshold, after which some of the normal information of the original mesh is retained, and b represents the rate at which normal information is retained. C is a scaling factor controlling the maximum value of $\mathsf{D}_{\mathcal{M}}$. We notice that the maximum of $\mathsf{D}_{\mathcal{M}}$ should be obtained at $t = 0$, that is, when all spatial information is random, in which case the expected average normal error should reach its theoretical maximum of $\pi/2$. For the usual range of values of a and $t = 0$, we have found experimentally that $(1 + e^{-a-bt}) \approx 1$ and thus $C \approx \pi/2$.

Eq. 3.2.1 is a member of the family of the *logistic functions*. These functions were initially introduced to model population growth, and have since found numerous applications in fields ranging from social sciences to engineering. In some of these applications, logistic functions have been used as degradation models, describing for example the transition of the state of a machine from working perfectly to total failure [170].

3.2.1 Logistic Curve Fitting

For a given mesh $\mathcal{M}$, its degradation model, that is, the values of the constants a, b and C, is computed empirically. Specifically, for several integer values of t between 0 and 64, we randomize all vertex coordinates after their t-th bit and compute the weighted average change of the face normals. This weighted average is considered a sample from the logistic curve at parameter value t. The curve itself is computed by applying logistic curve fitting on samples computed at several values of t.

To describe the above fitting process more formally, let the i-th vertex of a given triangle mesh $\mathcal{M}$ be given in Cartesian coordinates by $v_i = (x_i, y_i, z_i)$ $(1 \leq i \leq N, i \in \mathbb{N})$, where N is the number of vertices of $\mathcal{M}$. As the coordinates are assumed to be float-point numbers, they can be represented in double precision format (64-bit long). After this format conversion the mesh vertices have the form $\hat{v}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$, where $\hat{x}_i, \hat{y}_i$ and $\hat{z}_i$ are binary strings of 64 bits.

Next, we randomize the least significant bits. Specifically, for each $\hat{v}_i = (\hat{x}_i, \hat{y}_i, \hat{z}_i)$, we retain the t $(0 \leq t \leq 64, t \in \mathbb{N})$ most significant bits of each of $\hat{x}_i$, $\hat{y}_i$ and $\hat{z}_i$ and replace the $64 - t$ least significant bits with randomly generated bits. The result is a new set of coordinates $\check{v}_i = (\check{x}_i, \check{y}_i, \check{z}_i)$, which are converted into the floating point coordinates $v'_i = (x'_i, y'_i, z'_i)$ of the degraded mesh $\mathcal{M}^t$.

Next, we compare the triangle normals of $\mathcal{M}$ and $\mathcal{M}^t$ and the normal degradation is measured as the weighted average of the normal distortion

$$\mathsf{dis}(\mathcal{M}, \mathcal{M}^t) = \frac{\sum_{i=1}^{M} w_i \cdot \mathsf{angle}(\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_i^t)}{\sum_{i=1}^{M} w_i}, \tag{3.2.2}$$

where M is the number of triangles in $\mathcal{M}$, $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_i^t$, are the normals of the i-th triangle of $\mathcal{M}$ and $\mathcal{M}^t$, respectively, $\mathsf{angle}(\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_i^t)$ is the smaller angle between $\hat{\mathbf{n}}_i$ and $\hat{\mathbf{n}}_i^t$ expressed in radians.

If we put $w_i = 1$ for $i = 1, 2, \ldots, M$ we get the mean average of the normal distortion. Depending on the application, we might want to weight the average in Eq. 3.2.2 by the area A_i of the triangles, that is, $w_i = A_i$ for $i = 1, 2, \ldots, M$. In this case, the normal distortion of the larger triangles, which dominate the rendering process, has a larger weight. A third possibility is to use larger weights for triangles with small dihedral angles. Such triangles represent the flat areas of the surface where even very small normal distortions can be immediately perceived as noise. For a dihedral angle weighted average we used the Gaussian weights

$$w_i = \frac{1}{\sqrt{2\pi\sigma^2}}\, e^{-\dfrac{(x_i - \mu)^2}{2\sigma^2}} \tag{3.2.3}$$

where x_i is the smallest of the three dihedral angles of the i-th triangle. In the experiments, we fixed $\mu = 0$ and $\sigma = 3.5$, which gave reasonable results.

Regarding the properties of $\mathsf{dis}(\mathcal{M}, \mathcal{M}^t)$, from

$$0 \leq \mathsf{angle}(\hat{\mathbf{n}}_i, \hat{\mathbf{n}}_i^t) \leq \pi, \quad 0 \leq t \leq 64, \tag{3.2.4}$$

we get

$$0 \leq \mathsf{dis}(\mathcal{M}, \mathcal{M}^t) \leq \pi, \quad 0 \leq t \leq 64. \tag{3.2.5}$$

While it seems quite difficult to improve these deterministic bounds, nevertheless, regarding the expectations for $\mathsf{dis}(\mathcal{M}, \mathcal{M}^t)$, we can easily see that the expectation $E(\mathsf{dis}(\mathcal{M}, \mathcal{M}^t))$ is a decreasing function of t and that $E(\mathsf{dis}(\mathcal{M}, \mathcal{M}^0)) \approx \pi/2$ as already discussed. Moreover, all our experiments with commonly used triangle meshes confirm that $\mathsf{dis}(\mathcal{M}, \mathcal{M}^t) \approx 0$ as t approaches 64.

Finally, the last step of the process is to use logistic curve fitting and fit the logistic model of Eq. 3.2.1 to the samples computed by Eq. 3.2.2.

3.2.2 Estimation of Appropriate Quantization Levels

The proposed logistic model can be used to find the appropriate level of quantization when a tolerance for the expected normal error is given. Indeed, by solving Eq. 3.2.1 we get

$$t = -\Big(\ln \frac{C - \mathsf{D}(\mathcal{M}, \mathcal{M}^t)}{\mathsf{D}(\mathcal{M}, \mathcal{M}^t)} + a\Big)/b \tag{3.2.6}$$

The parameters a, b and C are experimentally computed as described in Section 3.2.1. We substitute the normal error predicted by the model, i.e. $\mathsf{D}(\mathcal{M}, \mathcal{M}^t)$, with the given tolerance and compute the appropriate level of quantization t from Eq. 3.2.6.

Eq. 3.2.6 can be further simplified by assuming $C = \pi/2$ and a fixed tolerance that would be acceptable for all intended applications. For example, for a normal error tolerance of $\epsilon = 1^\circ$ (≈ 0.01745 radians), which is acceptable in most visualization applications, after using the ceiling function to convert t to an integer, Eq. 3.2.6 becomes

$$t = \lceil -(4.489 + a)/b \rceil \tag{3.2.7}$$

Thus, using the proposed logistic model, we are able to figure out the suitable quantization level easily, once the normal error $\mathsf{D}(\mathcal{M}, \mathcal{M}^t)$ and the two parameters a and b are given.

3.3 Validation and Quantization Levels

In this section we experimentally validate the proposed logistic model and compute the appropriate levels of quantization based on this model. The former is to test the accuracy of the model, that is, to measure the difference between the prediction of the model $\mathsf{D}(\mathcal{M}, \mathcal{M}^t)$ and the observed average normal distortion $\mathsf{dis}(\mathcal{M}, \mathcal{M}^t)$. By contrast, the latter is to find out the appropriate level of quantization, subject to a given normal degradation.

The validation experiment was conducted on a set of synthetic and natural 3D triangle mesh models consisting of the *Fandisk*, *Bunny*, *Dragon*, *Lucy* and *MPII*

Table 3.1: Mesh details and the results of the logistic model fitting. For each of the mean average $\mathsf{dis}^{av}(\mathcal{M},\mathcal{M}^t)$, area weighted average $\mathsf{dis}^{ar}(\mathcal{M},\mathcal{M}^t)$ and dihedral angle weighted average $\mathsf{dis}^{an}(\mathcal{M},\mathcal{M}^t)$, the **top**, **middle** and **bottom** rows show the values of a, b and C, respectively.

	Fandisk	*Bunny*	*Dragon*	*Lucy*	*MPII Geometry*
M	12946	69666	100000	525814	70761
$\mathsf{dis}^{av}(\mathcal{M},\mathcal{M}^t)$	22.863	22.300	18.865	23.503	14.628
	-1.168	-1.154	-0.927	-1.081	-0.685
	1.573	1.570	1.571	1.571	1.569
$\mathsf{dis}^{ar}(\mathcal{M},\mathcal{M}^t)$	23.108	22.127	20.874	21.887	11.600
	-1.184	-1.148	-1.077	-1.028	-0.703
	1.573	1.570	1.572	1.571	1.568
$\mathsf{dis}^{an}(\mathcal{M},\mathcal{M}^t)$	22.864	22.315	19.095	23.307	14.624
	-1.168	-1.153	-0.939	-1.071	-0.684
	1.573	1.570	1.569	1.572	1.571

Geometry. The mesh details and the results of the fitting process are summarized in Table 3.1. We notice that in all cases the value of the threshold a is relatively large, meaning that 8-bit or even 12-bit mesh vertex quantizations result to the loss of almost all normal information. This is in contrast to the resiliency of the volumetric properties of the corresponding shapes, given that in 12-bit, or even 8-bit voxelizations, the shapes are still clearly recognizable. We also notice that the large value of a means that $C \approx \pi/2$ for each of the test meshes, as expected.

The comparisons between the predictions of the logistic model $\mathsf{D}(\mathcal{M},\mathcal{M}^t)$ and the corresponding experimental observations $\mathsf{dis}(\mathcal{M},\mathcal{M}^t)$ are shown in Fig. 3.2. The logistic model fits five data points computed at $t = 12$, 16, 20, 24 and 28 (see Eq. 3.2.2). From Fig. 3.2, we see that the red and blue curves of $\mathsf{D}(\mathcal{M},\mathcal{M}^t)$ and $\mathsf{dis}(\mathcal{M},\mathcal{M}^t)$, respectively, are almost identical. That means that the proposed logistic model can successfully predict the quality of the triangle normals. The difference between the two curves are generally small, except at the high slope part of the curves. We notice

Table 3.2: Appropriate quantization levels. For each of the mean average $\mathsf{dis}^{av}(\mathcal{M}, \mathcal{M}^t)$, area weighted average $\mathsf{dis}^{ar}(\mathcal{M}, \mathcal{M}^t)$ and dihedral angle weighted average $\mathsf{dis}^{an}(\mathcal{M}, \mathcal{M}^t)$, the **left**, **middle** and **right** columns correspond to $\epsilon = 1°$, $\epsilon = 5°$ and $\epsilon = 10°$, respectively.

	$\mathsf{dis}^{av}(\mathcal{M}, \mathcal{M}^t)$			$\mathsf{dis}^{ar}(\mathcal{M}, \mathcal{M}^t)$			$\mathsf{dis}^{an}(\mathcal{M}, \mathcal{M}^t)$		
Fandisk	24	22	22	24	22	22	24	22	22
Bunny	24	22	22	24	22	22	24	22	22
Dragon	26	24	23	24	22	22	26	24	23
Lucy	26	25	24	26	25	24	26	25	24
MPII Geometry	28	26	25	23	21	20	28	26	25

that the higher error at the high slope part of the curve was predictable, given that small misalignments at the horizontal direction can cause large discrepancies at the vertical direction. We also notice that quantizations corresponding to that part of the curve are not of direct interest in practical applications because their normal error is very large.

Fig. 3.2 shows that the logistic functions fit nicely the data from the five test models, except for the MPII Geometry model for some values of t for the case of area weighted average. The problem with the MPII Geometry model is caused by the high variance of the normal degradation, which as a result cannot be modeled effectively. In particular, for some values of t, the normal degradation variance of the very large triangles is also very large and dominates the area weighted average. We notice that the MPII model is the only one of our test models with very large triangles, as it contains triangles spanning whole sides of the building. Fig. 3.3 shows the variance computations for the test models.

As mentioned before, the proposed model is quite efficient in estimating the suitable mesh quantization level, given a normal degradation. Indeed, the appropriate levels of quantization t can be yielded by simply rounding t in Eq. 3.2.6 to the nearest integers towards infinity. For example, for $\epsilon = 1°$, the levels t for the five models of the validation experiment, computed according to Eq. 3.2.7, are shown in Table 3.2.

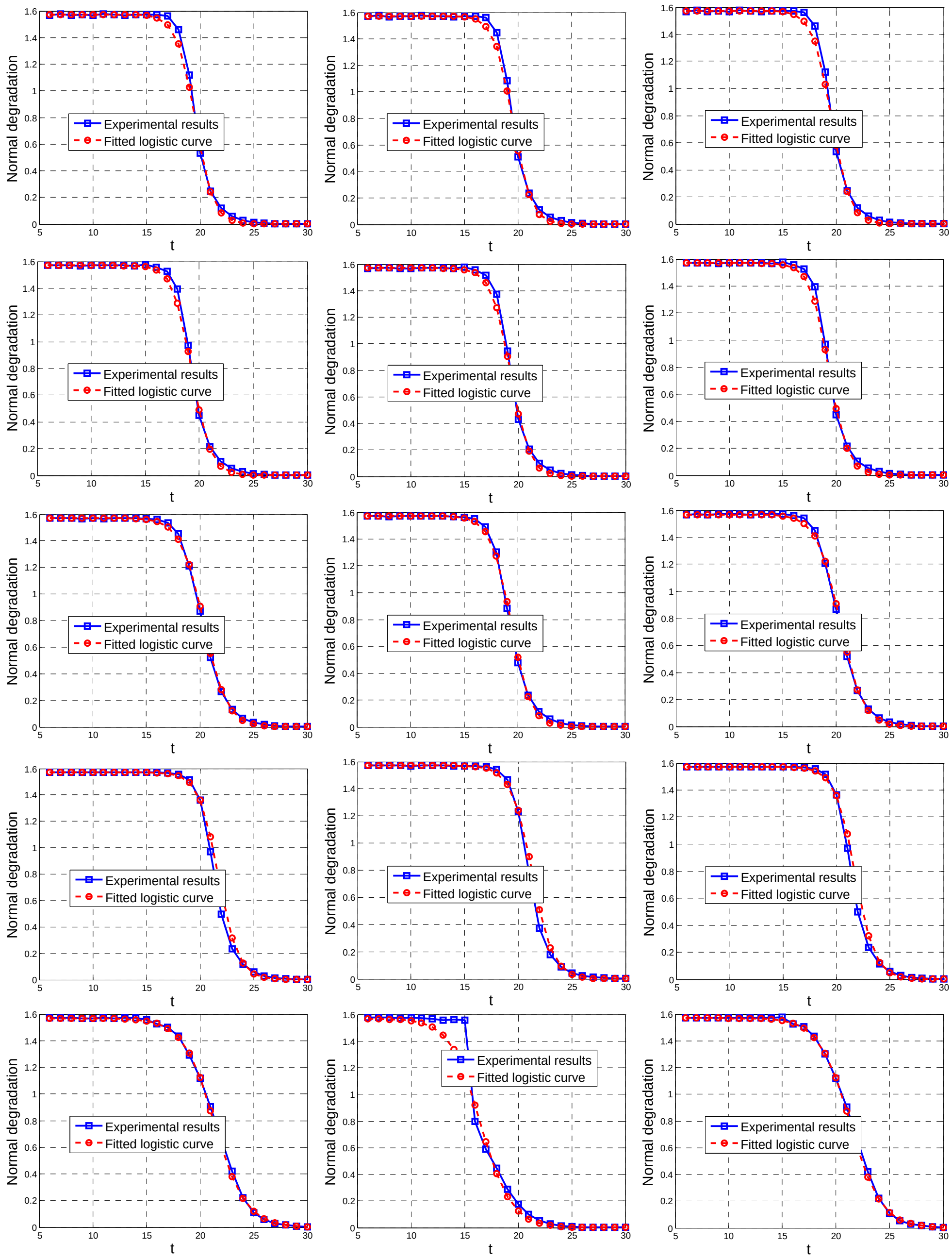


Figure 3.2: **Left:** Comparison between the average normal distortion observations $\mathsf{dis}^{av}(\mathcal{M}, \mathcal{M}^t)$ and the logistic model predictions. **Middle:** Comparison between the area-weighted average of normal distortion observations $\mathsf{dis}^{ar}(\mathcal{M}, \mathcal{M}^t)$ and the logistic model predictions. **Right:** Comparison between the dihedral angle-weighted average of normal distortion observations $\mathsf{dis}^{an}(\mathcal{M}, \mathcal{M}^t)$ and the logistic model predictions. The models used here are *Fandisk*, *Bunny Dragon*, *Lucy* and *MPII Geometry* (from top to bottom).

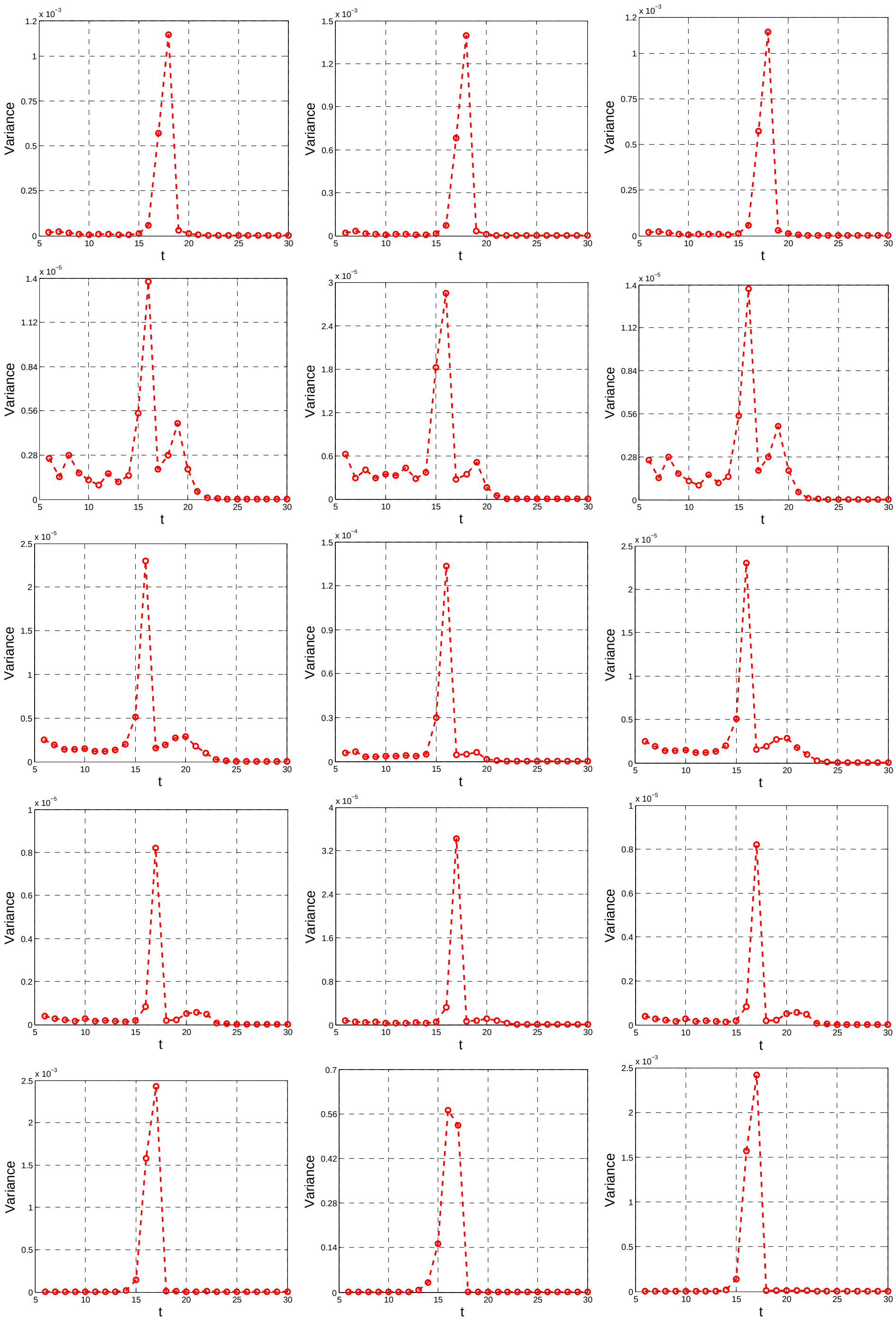


Figure 3.3: **Left:** Variances of the average normal distortion observations $\mathsf{dis}^{av}(\mathcal{M}, \mathcal{M}^t)$. **Middle:** Variances of the area-weighted average of normal distortion observations $\mathsf{dis}^{ar}(\mathcal{M}, \mathcal{M}^t)$. **Right:** Variances of the dihedral angle-weighted average of normal distortion observations $\mathsf{dis}^{an}(\mathcal{M}, \mathcal{M}^t)$. The models used here are *Fandisk*, *Bunny Dragon*, *Lucy* and *MPII Geometry* (from top to bottom).

The appropriate levels of quantization for $\epsilon = 5°$ and $10°$ can be computed analogously. We notice that, as expected, different models may have different appropriate level of quantization. For example, when the tolerance is $\epsilon = 1°$ and we consider the mean average of the normal distortion, the relatively simple *Fandisk* model can be represented by 24 bits per vertex coordinate without significant loss of normal information, while the more complex *Lucy* and *MPII Geometry* require 26 and 28 bits, respectively. We also notice that, in most models, the appropriate quantization level does not depend on the chosen averaging method. A notable exception is the *MPII Geometry* which consists of some extremely large and some extremely small triangles and the results of the area weighted averaging method diverge.

To judge the visual significance of the claimed optimisation of t, the left column of Fig. 3.4 shows renderings of the original meshes quantized at 64 bits per vertex coordinate and the middle column renderings of their respective appropriate quantizations for a small tolerance of $\epsilon \approx 1°$ when the mean averaging method is used. We notice that in all cases the appropriately quantized meshes are almost indistinguishable from the originals, and any possible degradation has been kept to visually acceptable levels.

To further demonstrate the relevance of our results, the right column of Fig. 3.4 shows renderings of the same meshes quantized at a coarser level corresponding to a tolerance of $\epsilon \approx 10°$. In practice that means that they are quantized at a level that is 2 or 3 bits coarser than the ones on the middle column. We notice that in all cases except the *Dragon* the degradation of the normal information is visually significant. The only exception is the *Dragon* model which does not show significant signs of degradation as a result of the coarser than appropriate quantization. The reason is that the original *Dragon* model is already noisy and thus, its original normal information has already a large random element which is not affected by the quantization. This salient point is further illustrated in Fig. 3.5 where close ups of the *Fandisk* and the *Bunny* at appropriate and suboptimal levels of quantization are shown. Finally, we notice that a given level of quantization ($t = 24$) may be appropriate for one mesh (*Fandisk*) and significantly suboptimal for another (*Lucy*). Fig. 3.5 shows close-ups of the models in Fig. 3.4.

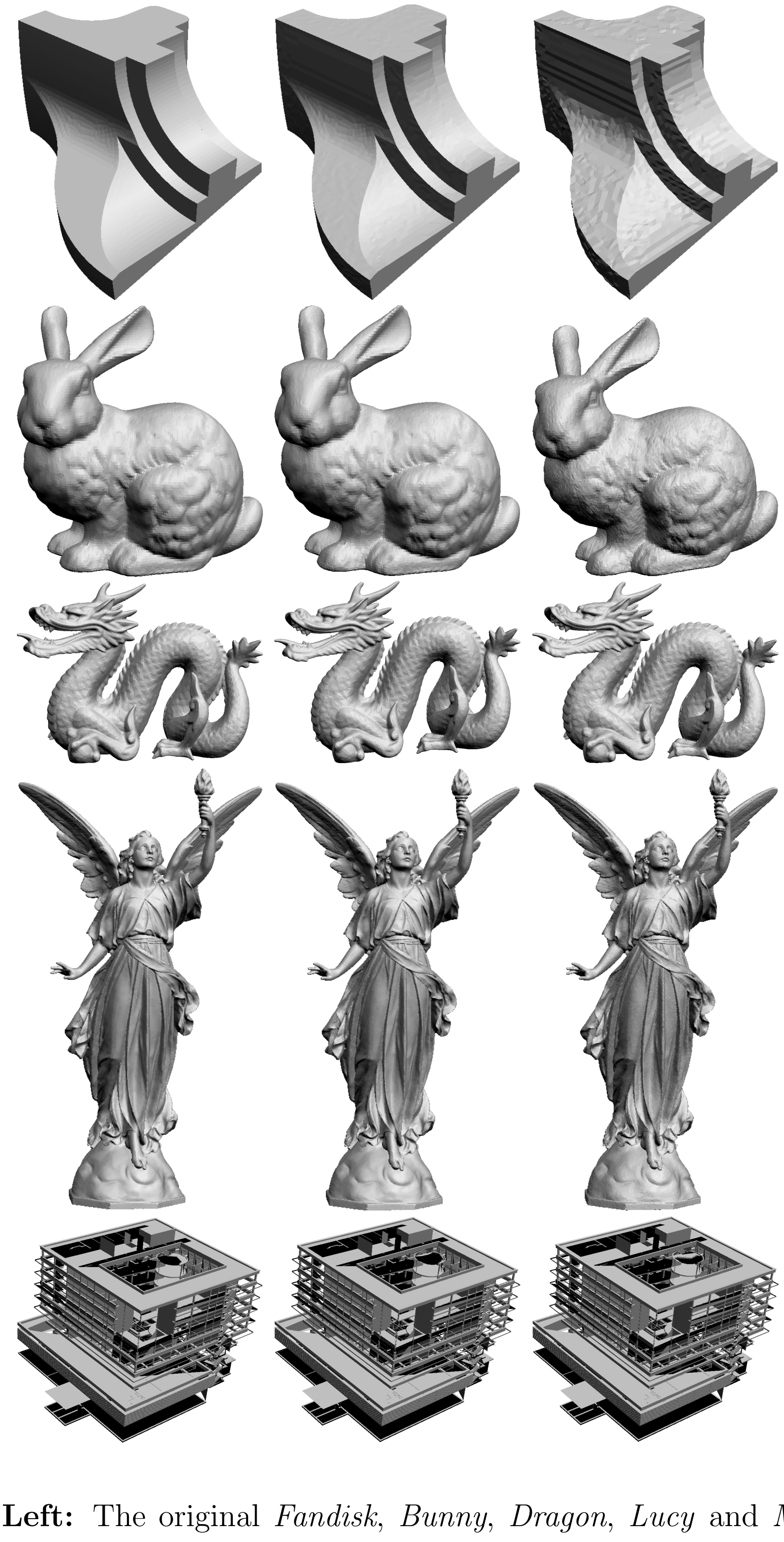


Figure 3.4: **Left:** The original *Fandisk*, *Bunny*, *Dragon*, *Lucy* and *MPII Geometry* meshes quantized at 64 bits per vertex coordinate. **Middle:** Finely quantized meshes at 24, 24, 26, 26 and 28 bits per vertex coordinate, respectively ($\epsilon \approx 1°$). **Right:** Coarsely quantized meshes at 22, 22, 23, 24 and 25 bits per vertex coordinate, respectively ($\epsilon \approx 10°$).

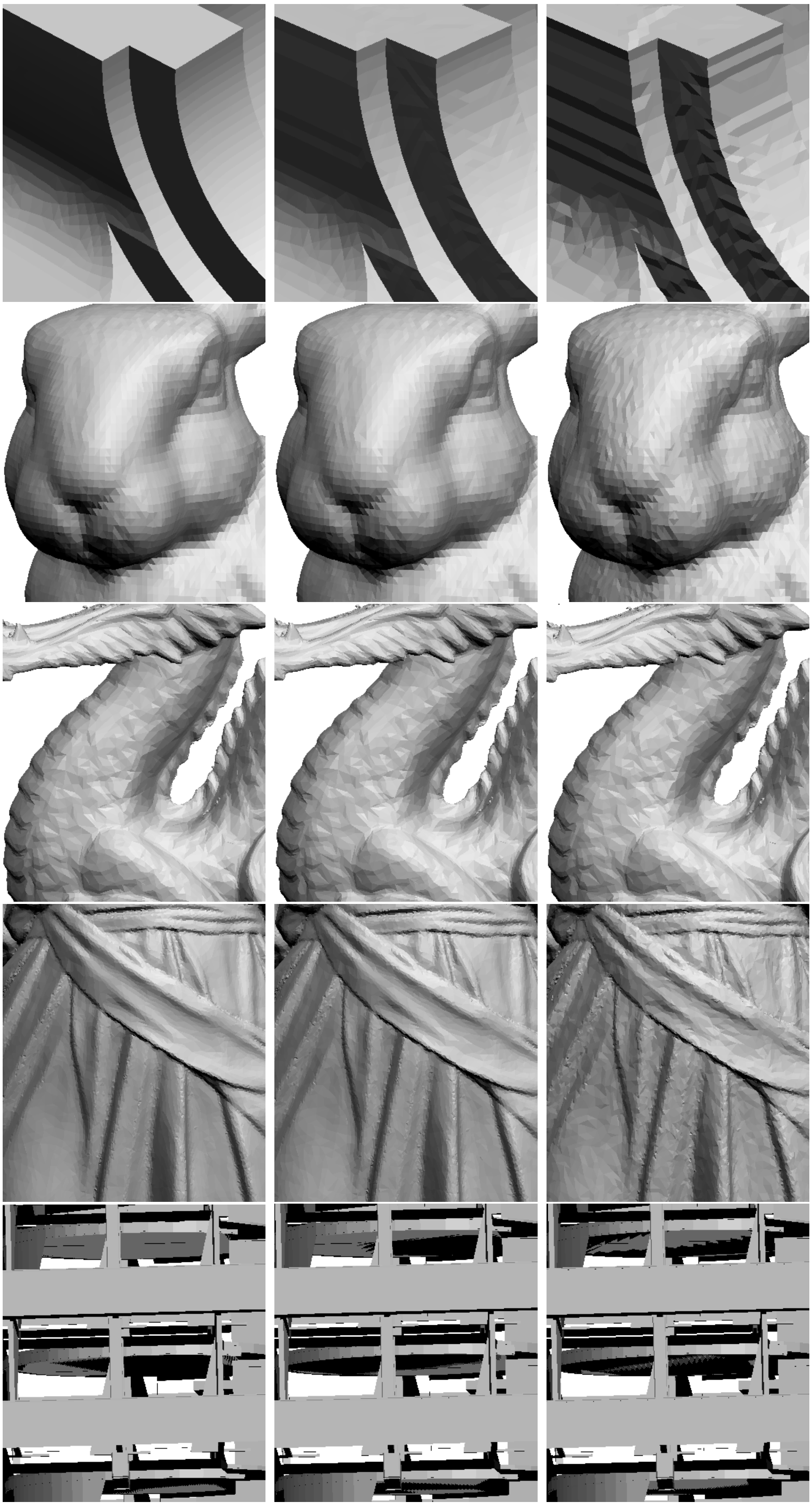


Figure 3.5: Close-ups of the models in Fig. 3.4.

3.4 Practical Applications

In this section we briefly discuss how the proposed logistic model for mesh degradation can be used to enhance existing mesh steganography/watermarking and mesh compression algorithms, help to evaluate such algorithms and inform their further development.

The goal of steganography is to embed a confidential message on a carrier signal, here a mesh model, in such a way that no one apart from the sender and the intended recipient can detect the existence of the hidden message. The challenge in designing a good steganographic algorithm lies in balancing the two conflicting requirements of high *embedding capacity* and low *embedding distortion*. That is, one has to maximise the length of the message that can be embedded on a given mesh and simultaneously minimize the visual impact of that embedding.

We notice that the proposed logistic model describes a trade-off between embedding capacity, in the form of unused bits in the vertex coordinates, and distortion, in the form of normal degradation. Therefore, it can directly be used to inform least significant bit steganographic algorithms about the embedding capacity of the carrier. More specifically, for a given carrier mesh $\mathcal{M}$, we first compute the parameters a, b of the logistic degradation model, as described in Section 3.2. Then, for a certain distortion tolerance $\mathsf{D}(\mathcal{M}, \mathcal{M}^t)$ we compute the appropriate level of quantization t using Eq. 3.2.6. Finally, the $64 - t$ least significant bits of each vertex coordinate are replaced with the message bits to be hidden.

The proposed logistic model can also be used in conjunction with mesh compression algorithms, as discussed in the Introduction. In particular, it can be used to inform the user's choice of level of quantization t for the vertex coordinates. After t has been determined, the compression algorithm will only keep the t most significant bits of each coordinate and the coarse mesh will be compressed and transmitted. At the receiver end, the coarse mesh will be decoded and $64 - t$ random bits will be appended to the t most significant bits of each coordinate. Such a process would be completely analogous to dithering, as used in signal encoding and compression.

3.5 Summary

We have proposed a logistic model for estimating the degradation of face normals in 3D triangle meshes caused the quantization of vertex coordinates. As demonstrated by the validation experiments, the behaviour of the proposed model is satisfactory and its predictions for the normal degradation are good approximations of the respective experimental values. We have also discussed how the proposed model might be utilized in a number of applications, especially those requiring the estimation of an appropriate quantization level for vertex coordinates, as for example mesh steganography/watermarking and mesh compression. Regarding algorithmic complexity, the algorithm runs in linear time $\mathrm{O}\big(n\big)$ as its time execution is directly proportional to the number of vertices n, i.e. time grows linearly as the vertex number increases.

Given the simplicity of the logistic model, such formula might be simpler and easier to compute than the formulas that we will in Chapter 4, which does not assume a logistic model for normal degradation.

Chapter 4

Normal Degradation of Triangle Meshes: A Theoretical Study

While the previous chapter presents an empirical study of normal degradation of 3D triangle models based on fitting a logistic model to the curve of normal degradation, this chapter is based on direct computation approximating the normal degradation.

We first compute in closed form the expectation for the angle θ between the new and the old normal when uniform noise is added to a single vertex of a triangle. Next, we propose and experimentally validate an approximation and lower and upper bounds for θ when uniform noise is added to all three vertices of the triangle. In all cases, for small amounts of spatial noise that do not severely distort the mesh, there is a linear correlation between θ and simple functions of the heights of the triangles and thus, θ can be computed efficiently. The addition of uniform spatial noise to a mesh can be seen as a dithered quantization of its vertices. We use the obtained linear correlations between spatial and normal noise to compute the level of dithered quantization of the mesh vertices when a tolerance for the average normal distortion is given.

The main contributions of this chapter are:

- An exact closed-form formula, and a linear approximation of it, for the expectation of the angle between the old and the new normal when noise is added to a single vertex of a triangle.

- An approximation, and heuristic lower and upper bounds, for the expectation of the normal perturbation when noise is added to all three vertices of a triangle.

- The fast computation of the dithered quantization level of a vertex when a tolerance for the degradation of the normals is given.

- A data hiding algorithm that can claim maximum capacity for the given tolerance of normal degradation.

The main limitation of the first contribution comes from the error of the linear approximation, which is small but not negligible for small amounts of added noise. Moreover, the error increases significantly when the amount of added noise becomes large. The approximation of the error is also a limitation of the second contribution. Moreover, the upper and lower bounds are heuristic and we have no mathematical proof that they always hold.

Regarding the third contribution, the extra limitation is the assumption that adding uniform spatial noise with support the shape of a cube is equivalent to dithered quantization. This is only approximately true when the edges of the cube do not align with the axes of the coordinate system.

The main limitation of the high-capacity data hiding method is its fragility, that is, it cannot survive attacks without loss of information.

Concerning the application, we can compute optimal levels i of *dithered quantizations* of the mesh vertices using the developed theoretical model, when a tolerance for the normal degradation is given. Another application lies in 3D steganography, where the message bits could be inserted into the i less significant bits.

The rest of this chapter is organized as follows. In Section 4.1, we study the expected change of the normal when a small amount of noise is added to a single triangle vertex. Based on this result, we in Section 4.1 also derive heuristic bounds and an approximate formula for the expected normal change when a small amount of noise is added to all three vertices of the triangle. In Section 4.2, we validate the bounds and the approximation by comparing them to the average normal change on test meshes when actual noise is added to the mesh vertices and also measure

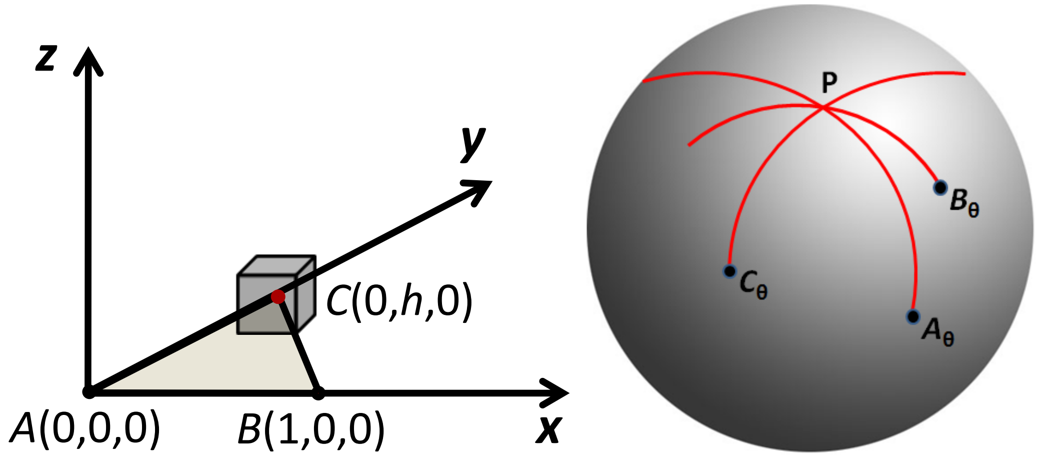


Figure 4.1: **Left:** Adding noise to a single triangle vertex. **Right:** Normal orbits on the Gaussian sphere.

the time performance of the proposed algorithm. In section 4.3, we present two potential applications. Material in this chapter has been published in [10].

4.1 Normal Noise Estimation

In our model, the noise added to a vertex P of the triangle mesh is described by a random variable u with distribution $\mathbf{p}(u)$, $u \in \mathbf{R}^3$. To measure the effect of the added random variable on the normal of a triangle T incident to P, we compute the expectation $\mathrm{E}(\theta(u))$ for the angle $\theta(u)$ between the normal of T and its normal after u has been added to P. The expectation is given by

$$\mathrm{E}(\theta(u)) = \int\limits_{\Omega} \mathbf{p}(u)\theta(u)du \tag{4.1.1}$$

where Ω is the support of $\mathbf{p}(u)$.

The expectation in Eq. 4.1.1 can be computed by standard numerical integration methods for any given probability distribution that may appear in practice, for example the Gaussian or the uniform distributions. In what follows, we study uniform noise with cubic shaped support. In practice, such types of noise are the result of the randomization of the least significant bits of the vertex coordinates and appear in applications such as dithered quantization and data hiding. In particular, we will compute the integral in Eq. 4.1.1 in a closed form and find local linear approximations of it. To simplify the notation we will write θ instead of $\theta(u)$ when the context removes any ambiguity.

4.1.1 Adding Uniform Noise To a Single Vertex

Let $T = ABC$ be a triangle. We will compute $\mathrm{E}(\theta)$ when u is added to the vertex C with $\mathbf{p}(u)$ the uniform distribution with support a cube of edgelength $2l$, centered at C with one face parallel to ABC, see Fig. 4.1 (left). In this case, $\mathbf{p}(u)$ is equal to the constant $1/8l^3$ inside the cube and zero outside.

By a basis change of the Cartesian xyz coordinate system, including scaling, we may assume without loss of generality that $A = (0,0,0)$, $B = (1,0,0)$ and C lies in the xy-plane. We notice that moving $C + u$ parallely to the x-axis by adding a displacement $t\vec{AB}$ does not change $\mathrm{E}(\theta)$ because the two triplets of points $(A, B, C + u)$ and $(A, B, C + u + t\vec{AB})$ define the same plane, hence have the same normal. That means that without loss of generality we may assume that $C = (0, h, 0)$, where h is the height of ABC at C, see Fig. 4.1 (left).

Intuitively, the above simplification of the problem is based on the fact that if we fix A, B and add the random variable u to C then $\mathrm{E}(\theta)$ depends on the distance h of C from AB only. Formally, the simplification is a consequence of the equality

$$\theta(u) = \theta(u + t\vec{AB}) \tag{4.1.2}$$

giving,

$$\int\limits_{\Omega+t\vec{AB}} (1/8l^3)\theta(u + t\vec{AB})du = \int\limits_{\Omega} (1/8l^3)\theta(u)du \tag{4.1.3}$$

Let $u = (x, y, z)$, we have

$$\theta = \begin{cases} \arctan(|z|/(h+y)), & \text{if } y > -h \\ \pi/2, & \text{if } y = -h \\ \pi + \arctan(|z|/(h+y)), & \text{if } y < -h \end{cases} \tag{4.1.4}$$

The simple form of the components of Eq. 4.1.4 allows the computation of $\mathrm{E}(\theta)$ in closed form. First, we notice that because of the invariance of θ when u is reflected

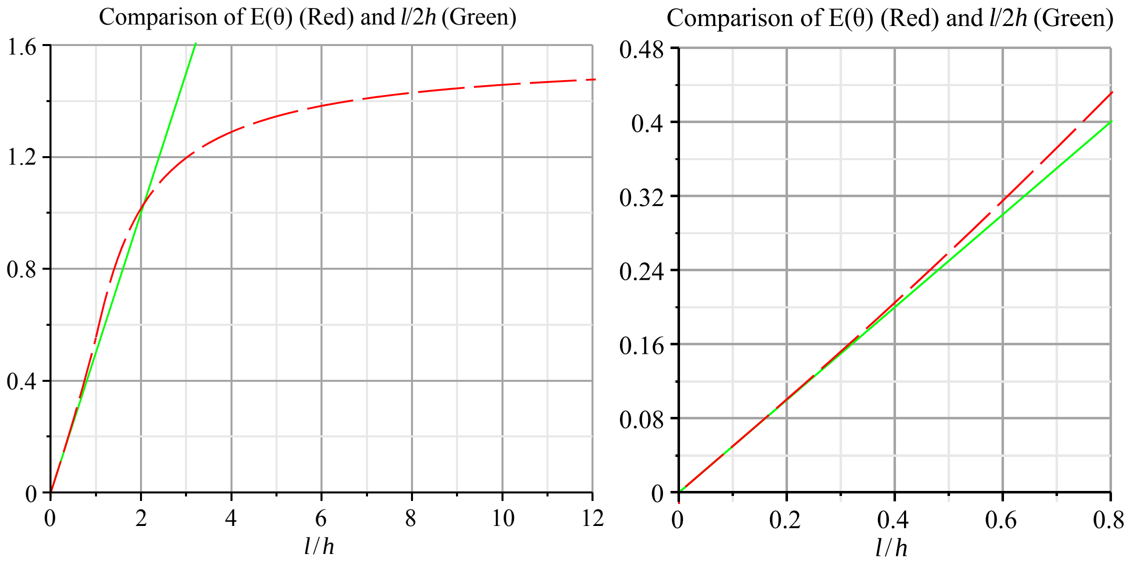


Figure 4.2: **Left:** Plot of $\mathrm{E}(\theta)$ and its tangent at 0, $h = 1$. **Right:** A close-up of the graph.

through the xy-plane, it suffices to compute $\mathrm{E}(\theta)$ for $z \geq 0$ only. Moreover, the invariance of θ when a vector parallel to the x-axis is added to u means that we do not have to integrate along x.

For $l < h$, which is the main case as it represents the tolerable amounts of noise, we get

$$\begin{aligned}\mathrm{E}(\theta) &= \frac{1}{2l^2}\int_{-l}^{l}\int_{0}^{l} \theta \; dz \; dy \\ &= \frac{1}{2l^2}\int_{h-l}^{h+l}\int_{0}^{l} \arctan(z/y) \; dz \; dy \qquad (4.1.5)\end{aligned}$$

and the definite integral can be directly computed from the indefinite integral

$$\begin{aligned}&\iint \arctan(z/y) \; dy \; dz = \\ &yz \arctan(z/y) + [(z^2 - y^2)\log(y^2+z^2) + y^2]/4 + c \qquad (4.1.6)\end{aligned}$$

giving,

$$\begin{aligned}\mathrm{E}(\theta) = \frac{1}{8r^2}\log\left(\frac{\left((1-r)^2+r^2\right)^{1-2r}\left(1+r\right)^{2(r+1)^2}}{\left((1+r)^2+r^2\right)^{1+2r}\left(1-r\right)^{2(1-r)^2}}\right) \\ + \frac{1+r}{2r}\arctan(\frac{r}{1+r}) - \frac{1-r}{2r}\arctan(\frac{r}{1-r}) \qquad (4.1.7)\end{aligned}$$

Table 4.1: The linear approximation of $\mathrm{E}(\theta)$.

l/h	.1	.2	.3	.4	.5
$\mathrm{E}(\theta) - (l/2h)$	.0001	.0006	.0021	.0049	.0092

where $r = l/h$.

The case $l > h$ in Eq. 4.1.4 differs from the case $l < h$ only by a constant and one just has to add

$$\frac{1}{2l^2}\int_{h-l}^{0}\int_{0}^{l} \pi \, dz \, dy = \frac{\pi(r-1)}{2r} \tag{4.1.8}$$

to Eq. 4.1.7 to compute $\mathrm{E}(\theta)$ for that case. In both cases it turns out that $\mathrm{E}(\theta)$ depends on the quotient l/h rather than the individual values of h and l, meaning that the expectation is scale invariant.

Simple computations, using del'Hospital's rule when necessary, show that the derivative of $\mathrm{E}(\theta)$ at $r = 0$ is equal to $1/2$. Fig. 4.2 shows the graph of $\mathrm{E}(\theta)$ and its tangent at 0. We notice that for relatively small values of l/h, the tangent approximates well the curve of $\mathrm{E}(\theta)$ and can be used as an approximation of its value by

$$\mathrm{E}(\theta) \simeq r/2 = l/2h \tag{4.1.9}$$

Table 4.1 shows the differences between the curve of $\mathrm{E}(\theta)$ and its tangent at 0 for several values of l/h. For the rest of the chapter we will mostly use this approximate value of $\mathrm{E}(\theta)$, even though the exact value can be computed from Eq. 4.1.7.

4.1.2 Adding Noise to All Three Vertices

The next step is to find heuristic bounds for $\bar{\mathrm{E}}(\theta)$, the expectation for the angle between the old and the new normal when noise is added on all three vertices of T. We also compute a point estimate of $\bar{\mathrm{E}}(\theta)$, using a simple linear function of the expectations $\mathrm{E}(\theta)$ computed on the three vertices T as described in Section 2.1.

Let $h_1 \leq h_2 \leq h_3$ be the heights of T at A, B and C, respectively and let $\mathrm{E}(\theta_1) \geq \mathrm{E}(\theta_2) \geq \mathrm{E}(\theta_3)$ be the corresponding expectations when noise is added to a single vertex of the triangle. We propose

$$\mathrm{E}_{min}(\theta) = \mathrm{E}(\theta_1) \tag{4.1.10}$$

as a heuristic lower bound for $\bar{\mathrm{E}}(\theta)$.

The heuristic argument is that if after adding noise to one of the triangle's vertices we proceed and add noise to the other two vertices too, we will increase the spatial uncertainty and thus increase the expectation for the normal perturbation $\bar{\mathrm{E}}(\theta)$. To see the same argument from a slightly different angle, a lower expectation for θ would mean that the normal of the triangle with noise on its three vertices is a better estimate of the original normal than the normal of the triangle with noise on a single vertex. As we cannot improve the estimate by adding more noise, we expect that $\bar{\mathrm{E}}(\theta) \geq \mathrm{E}(\theta_1) \geq \mathrm{E}(\theta_2) \geq \mathrm{E}(\theta_3)$.

For the heuristic construction of an upper bound for $\bar{\mathrm{E}}(\theta)$, we treat the normal pertubations from the addition of noise on each vertex as independent. This is a valid assumption when the amount of noise added to each vertex is small and does not change significantly the heights of T. Under this independence assumption, an obvious upper bound for $\bar{\mathrm{E}}(\theta)$ is the sum

$$\mathrm{E}(\theta_1) + \mathrm{E}(\theta_2) + \mathrm{E}(\theta_3) \tag{4.1.11}$$

We notice that for an equilateral triangle T we have

$$\mathrm{E}(\theta_1) + \mathrm{E}(\theta_2) + \mathrm{E}(\theta_3) = 3\,\mathrm{E}_{min}(\theta) \tag{4.1.12}$$

meaning that for the equilateral and almost equilateral triangles that are common in high quality meshes the pair of bounds in Eq. 4.1.10 and Eq. 4.1.11 is not tight enough in the sense that the quantization problems we deal with in the applications will have more than one solution. Indeed, in such problems the quantization levels are integers solutions in a logarithmic space of basis 2 and, ideally, for a unique solution we would like the lower and the upper bounds to differ by a factor of 2 at most. For the construction of a sharper heuristic upper bound, we sum the expected normal perturbations on the Gaussian sphere, using the fact that when noise is added to a single vertex, the trajectory on the Gaussian sphere of the possible normal perturbations is a circle.

Assume that the normal of T maps to the top of the Gaussian sphere P. By adding noise to the vertex A only, the orbit of the perturbed normal on the Gaussian sphere is a maximal arc with center at P. Let A_θ be a point on that arc such that $Arc(PA_\theta) = \mathrm{E}(\theta_1)$. Similarly, if we add noise to the vertex B only, the orbit of the perturbed normals is a maximal arc with center at P, and let B_θ be a point on that arc such that $Arc(PB_\theta) = \mathrm{E}(\theta_2)$. The point C_θ is constructed similarly, see Fig. 4.1 (right). For a small l, the points P, A_θ, B_θ, C_θ are almost coplanar and we can approximate A_θ, B_θ, C_θ with their projections A'_θ, B'_θ, C'_θ on the tangent of the Gaussian sphere at P. Working on that tangent plane instead of the sphere, we expect the angle corresponding to the sum

$$\max\{\pm \vec{PA'_\theta} \pm \vec{PB'_\theta} \pm \vec{PC'_\theta}\} \tag{4.1.13}$$

to be an upper bound because Eq. 4.1.13 will choose for each vector the direction that will maximise the sum.

Working on the tangent plane at P, we notice that the angles of the intersection of the the three lines PA', PB', PC' are equal to the angles of the intersection of the heights of the triangle which are equal to the angles of the triangle. We also notice that the lengths of the vectors of Eq. 4.1.13 are proportional to the inverses of the triangle's heights, meaning that they are proportional to the triangle's edges. Simple arguments show that if h_a is the smallest height of the triangle

$$\max\{\pm \vec{PA'_\theta} \pm \vec{PB'_\theta} \pm \vec{PC'_\theta}\} = 2\vec{PA'_\theta} \simeq 2\vec{PA_\theta} \tag{4.1.14}$$

and the proposed upper bound becomes

$$\mathrm{E}_{max}(\theta) = 2\vec{PA_\theta} = 2\,\mathrm{E}_{min}(\theta) \tag{4.1.15}$$

Finally, for obtaining a point estimation of $\bar{\mathrm{E}}(\theta)$ we use the simple linear approximation

$$\mathrm{E}_{appr}(\theta) = \eta \cdot (\mathrm{E}(\theta_1) + \mathrm{E}(\theta_2) + \mathrm{E}(\theta_3)) \tag{4.1.16}$$

The constant η is estimated experimentally. To estimate η we simulated the addition of a small amount of noise, $l = 2^{-10}$ in particular, to the vertices of an equilateral triangle of size 1, and after averaging over a large number of experiments we found the value $\eta = 0.608$.

Notice that, based on Eq. 4.1.9, Eq. 4.1.16 can be written as

$$\mathrm{E}_{appr}(\theta) = \eta \cdot l \cdot (\frac{1}{2h_1} + \frac{1}{2h_2} + \frac{1}{2h_3}) = \frac{\eta \cdot l}{2\rho} \tag{4.1.17}$$

where ρ is the radius of the incircle of T. In other words, a consequence of the approximation $\mathrm{E}_{appr}(\theta)$ is that, for small amounts of noise, $\mathrm{E}(\theta)$ is linear to the inverse of the radius ρ of the incircle.

4.2 Tests and Validation

In this section we test the obtained bounds and approximations on several well-known triangle meshes that are available online and measure the time performance of the proposed algorithm.

4.2.1 Validation

Let

$$\mathrm{E}_{mesh}(\theta) = \frac{\sum_{j=1}^{M} \bar{\mathrm{E}}_j(\theta)}{M} \tag{4.2.18}$$

be the average normal degradation over the triangles of a mesh. Here, M is the number of triangles of the mesh and $\bar{\mathrm{E}}_j(\theta)$ denotes the expectation $\bar{\mathrm{E}}(\theta)$ for the normal degradation of the j-th triangle.

To validate the accuracy of the bounds and the point approximation, we replaced $\bar{\mathrm{E}}_j(\theta)$ in Eq. 4.2.18 with $\mathrm{E}_{min}(\theta), \mathrm{E}_{max}(\theta)$ and $\mathrm{E}_{appr}(\theta)$, respectively, and compared it with the average normal degradation we computed experimentally using synthetic added noise.

The amounts of added noise were powers of 2, that is, $l = 2^{-i}$. The values of l were small enough for the linear correlation between $\bar{\mathrm{E}}(\theta)$ and l/h to be strong. The results are summarized in Fig. 4.3. As expected, $\mathrm{E}_{min}(\theta)$ and $\mathrm{E}_{max}(\theta)$ bound $\bar{\mathrm{E}}(\theta)$ from below and above and thus $\mathrm{E}_{mesh}(\theta)$ too. Moreover, $\mathrm{E}_{appr}(\theta)$ provides a good approximation of the experimental results. Notice that the graphs in Fig. 4.3 are drawn for values of i that are large enough for the linear approximation in Eq. 4.1.9 to be valid.

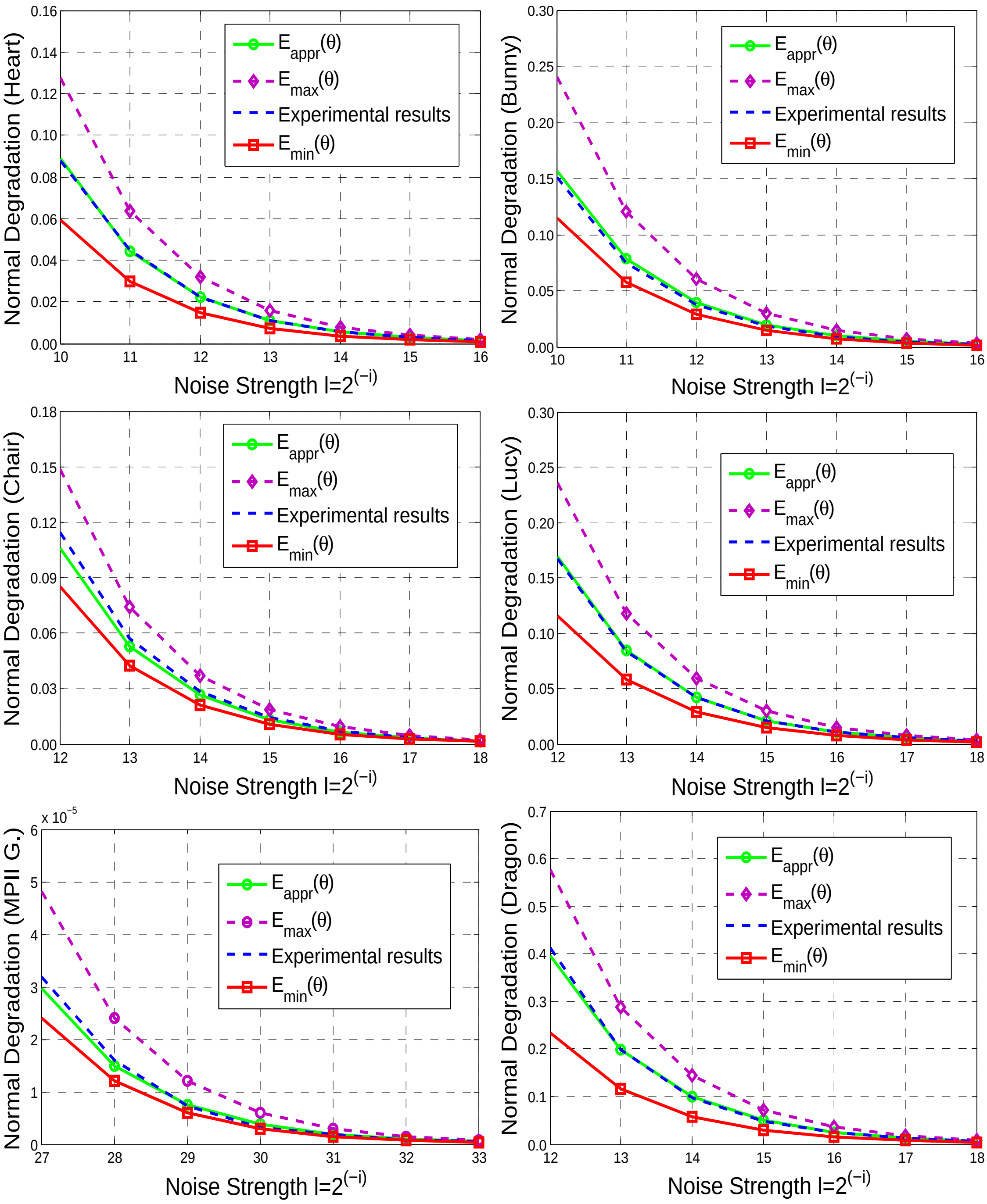


Figure 4.3: Validation results for *Heart*, *Bunny*, *Chair*, *Lucy*, *MPII Geometry* and *Welsh Dragon*. The x-axis corresponds to the values of i. The y-axis is in radians.

In some cases, if the mesh contains many small and skinny triangles the linear approximation may not be valid even for relatively large values of i. Fig. 4.4 (left) shows that in the *MPII Geometry* model $\mathrm{E}_{appr}(\theta)$ is below the lower bound $\mathrm{E}_{min}(\theta)$

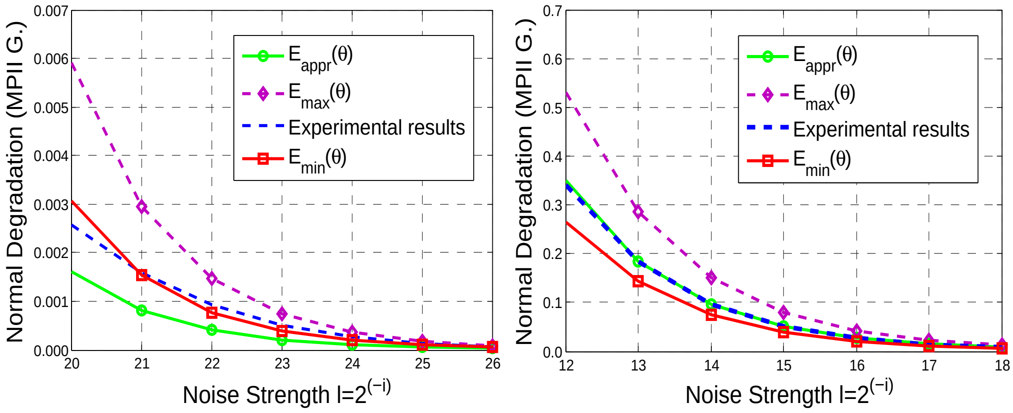


Figure 4.4: For the *MPII Geometry* model, the linear approximation of the expectation gives poor results even for relatively high values $i > 20$. However, the use of the exact expectation gives satisfactory results even for the lower range $12 \leq i \leq 18$.

even for levels of quantization $i > 20$. In these cases, one should use the exact expectation in Eq. 4.1.7 instead of the approximation in Eq. 4.1.9. Indeed, Fig. 4.4 (right) shows that the use of the exact expectation gives satisfactory results even for the range $12 \leq i \leq 18$.

4.2.2 Time Performance

Regarding its computational complexity, the algorithm is linear with the number of triangles. Given a normal degradation tolerance ϵ, our current non-optimized *Matlab* implementation on an Intel Core 2 Duo E8400 3.00 GHz processor with 2GB memory calculates the optimal quantization level for the 2M triangles of the *Welsh Dragon* in 56 seconds. Apart from programming optimizations, we can significantly speed up the computation by using the much simpler Eq. 4.1.10 instead of Eq. 4.1.17. Indeed, as Eq. 4.1.17 is between the lower bound in Eq. 4.1.10 and the upper bound in Eq. 4.1.15, and as the lower and the upper bounds differ by 2, the results of Eq. 4.1.10 will differ from the results of Eq. 4.1.17 by one bit at most.

Finally, we note that the linear correlation between spatial and normal noise can be used without any computational cost as a rule of thumb that one extra bit will halve the normal error. This simplicity is the main motivation for using the approximations for all the examples, instead of using the exact formula in Eq. 4.1.7.

4.3 Applications

In this section we discuss how the proposed method can be used for obtaining optimal dithered quantizations as well as for adaptive high-capacity mesh data hiding.

4.3.1 Optimal Dithered quantizations

A simple dithered quantizer for the mesh vertices retains the i most significant bits of each vertex coordinate and replaces any less significant bit with random bits. Such dithered quantizers avoid the blocky artifacts created by quantizers that set all less significant bits to zero.

Given a tolerance ϵ for the average normal degradation $\mathrm{E}_{mesh}(\theta)$, the bounds and the point estimate of Section 4.1 can be used to compute optimal levels for the dithered quantization of the vertices. The main assumption is that the randomization of all bits that are less significant than bit k can be approximated by the addition of uniform noise with cubic support of edgelength $2l = 2^{-k}$.

In particular, we can compute the optimal level of quantization i as

$$i = \arg\max_{k \in \mathbb{Z}} \{k \mid \mathrm{E}_{mesh}(\theta) \leq \epsilon\} \tag{4.3.19}$$

where, depending on the application, the average $\mathrm{E}_{mesh}(\theta)$ is computed using the lower bound $\mathrm{E}_{min}(\theta)$, or the upper bound $\mathrm{E}_{max}(\theta)$, or the estimate $\mathrm{E}_{appr}(\theta)$. For instance, if we use $\mathrm{E}_{min}(\theta)$, from Eqs. 4.1.9, 4.1.10, 4.2.18 and 4.3.19, we get

$$i = \left\lceil -2 - \log_2 \left(\frac{M \cdot \epsilon}{\sum_{j=1}^{M} 1/h'_j} \right) \right\rceil \tag{4.3.20}$$

Similarly, if we use $\mathrm{E}_{appr}(\theta)$, then from Eqs. 4.1.17, 4.2.18, and 4.3.19, we obtain

$$i = \left\lceil -2 - \log_2 \left(\frac{M \cdot \epsilon}{\eta \cdot \sum_{j=1}^{M} 1/\rho_j} \right) \right\rceil \tag{4.3.21}$$

In the above two equations, $\lceil \cdot \rceil$ stands for the mathematical ceiling function and h'_j and ρ_j are the minimum height and the incircle radius of the j-th triangle, respectively.

Table 4.2 summarizes the results for several values of the tolerance ϵ. The quantization levels for *MPII Geometry* were computed using the exact Eq. 4.1.7 instead of Eq. 4.1.17.

Table 4.2: For each of $\mathrm{E}_{min}(\theta)$, $\mathrm{E}_{appr}(\theta)$ and $\mathrm{E}_{max}(\theta)$, the **left**, **middle** and **right** columns show the quantization levels corresponding to $\epsilon = 0.1°$, $\epsilon = 1°$ and $\epsilon = 10°$, respectively.

	#Tri	$\mathrm{E}_{min}(\theta)$			$\mathrm{E}_{appr}(\theta)$			$\mathrm{E}_{max}(\theta)$		
Heart	37690	15	11	8	15	12	9	16	12	9
Bunny	69666	16	12	9	16	13	9	17	13	10
Chair	6664	17	14	10	17	14	11	18	15	11
Lucy	525814	18	14	11	18	15	11	19	15	12
MPII Geometry	70761	20	16	12	20	16	13	21	17	13
Welsh Dragon	2210635	19	15	12	19	16	13	20	16	13

We notice that the size of the mesh influences the quantization level. The reason is that the number of triangles in the common natural meshes is correlated to their average size, with the large meshes mainly consisting of small triangles with normals that are more sensitive noise. For an information-theoretic explanation, we can say that larger meshes contain more geometric detail, which requires more bits per vertex coordinate for its representation. The shape of the mesh triangles also influences the level of quantization. In particular, after taking the number of mesh triangles into account, the CAD meshes *Chair* and *MPII Geometry*, which have a significant portion of skinny triangles, require more bits per coordinate compared to scan meshes such as *Lucy* and *Welsh Dragon* which mainly consist of well shaped triangles. For an information-theoretic explanation, CAD meshes usually contain more information per triangle than the scan meshes and thus, require more bits per vertex coordinate.

Table 4.3 shows that the experimental values of the normal degradation computed as

$$\psi^i = \frac{\sum_{j=1}^{M} \mathsf{angle}\left(\hat{\mathbf{n}}_j, \hat{\mathbf{n}}_j^i\right)}{M} \tag{4.3.22}$$

where $\hat{\mathbf{n}}_j$ and $\hat{\mathbf{n}}_j^i$ are the normals of the j-th triangle of the original mesh and the mesh quantized at i bits, respectively. For a given tolerance ϵ, the level of quantization i is the one corresponding to $\mathrm{E}_{appr}(\theta)$ as shown in Table 4.2. We notice

Table 4.3: The experimental values of normal degradation ψ^i corresponding to the quantization levels given by the $\mathrm{E}_{appr}(\theta)$ column in Table 4.2

	$\epsilon = 0.1°$	$\epsilon = 1°$	$\epsilon = 10°$
Heart	$\psi^{15} = 0.111$	$\psi^{12} = 0.884$	$\psi^{9} = 7.064$
Bunny	$\psi^{16} = 0.099$	$\psi^{13} = 0.750$	$\psi^{9} = 12.276$
Chair	$\psi^{17} = 0.099$	$\psi^{14} = 0.851$	$\psi^{11} = 7.461$
Lucy	$\psi^{18} = 0.105$	$\psi^{15} = 0.837$	$\psi^{11} = 13.543$
MPII Geometry	$\psi^{20} = 0.092$	$\psi^{16} = 0.892$	$\psi^{13} = 6.496$
Welsh Dragon	$\psi^{19} = 0.123$	$\psi^{16} = 0.982$	$\psi^{13} = 7.684$

that, in all cases, the experimental results differ from the prescribed tolerance by a factor smaller than two, meaning that the computed quantization levels differ from the optimal for that tolerance levels by one bit at most. On the other hand, we also notice that in four cases the experimental normal degradation exceeds the tolerance, even though the quantization levels corresponding to $\mathrm{E}_{appr}(\theta)$ happen to be the same as the ones corresponding to $\mathrm{E}_{min}(\theta)$. At the beginning of this chapter, we discuss several limitations of the method that might be responsible for this discrepancy.

Optimally quantized test meshes for several values of ϵ are shown in Fig. 4.5 and close-ups of some of them are shown in Fig. 4.6. As expected, for all test meshes, the quantizations corresponding to tolerances $\epsilon = 0.1°$ and $\epsilon = 1°$ are almost indistinguishable between them and visually equivalent to the originals. By contrast, the meshes that have suffered an average $\epsilon = 10°$ normal degradation are of noticeably inferior visual quality.

In some applications, the average normal degradation over all mesh triangles may not be a satisfactory indicator of mesh quality. One such case is when the size or the shape of the mesh triangles is not uniform across the mesh. Another issue limiting the visual relevance of the average normal degradation is that the human eye sensitivity to normal perturbations depends on the smoothness of the surface areas where the degradation occurred. In particular, the eye is more sensitive to normal perturbations in the smoother regions of a surface. Hence, some applications

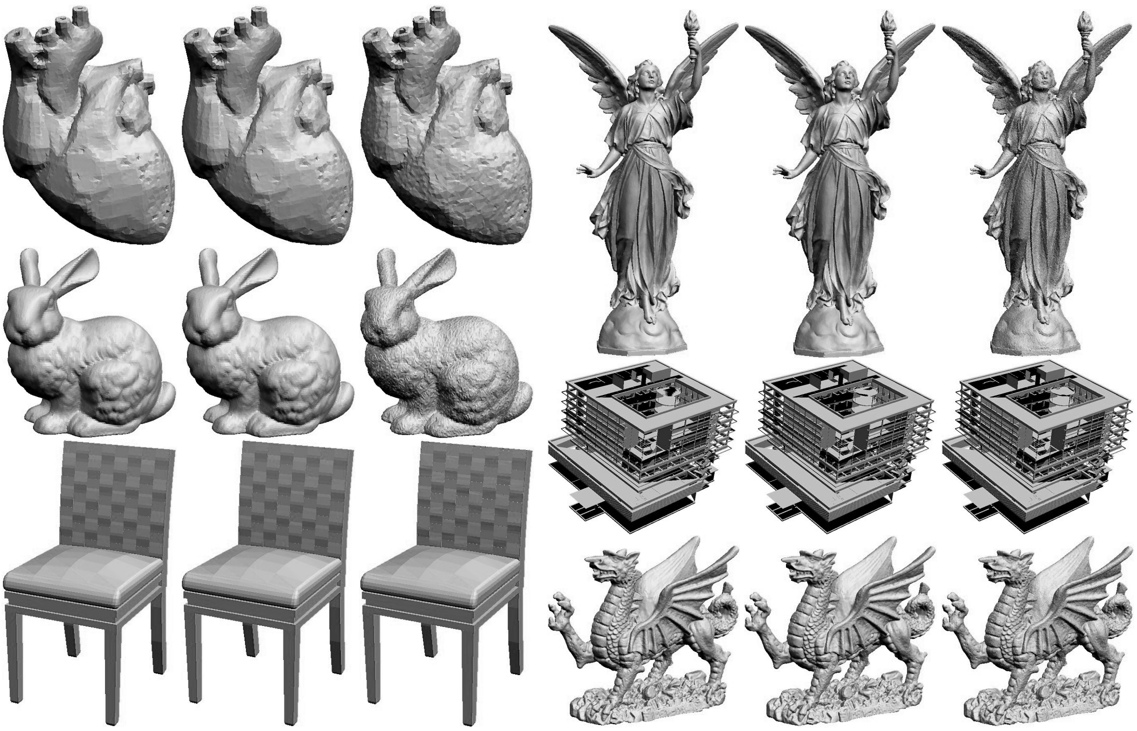


Figure 4.5: For each mesh group, the **left**, **middle** and **right** show the dithered quantized meshes for $\epsilon = 0.1^\circ$, $\epsilon = 1^\circ$ and $\epsilon = 10^\circ$, respectively.

might require a normal degradation tolerance that is smaller at the smoother areas of the surface. In such cases, computing a separate optimal level of quantization for each mesh vertex may give a more satisfactory solution.

To do this, we use the Gaussian curvature κ_n given by

$$\kappa_n = 2\pi - \sum_{i=1}^{v_n} \alpha_{n,i}, \quad 1 \leq n \leq N \tag{4.3.23}$$

as a measure of mesh smoothness at a mesh vertex, where N is the number of the mesh vertices, v_n is the degree of the n-th vertex and $\alpha_{n,i}$ is the i-th angle incident the n-th vertex. The smaller the absolute value $|\kappa_n|$ is, the flatter the surface at the vicinity of the n-th vertex.

The normal degradation tolerance ϵ_j at the j-th triangle is given by

$$\epsilon_j = \min_n \{\alpha - \phi(\kappa_n; \alpha, \mu, \sigma) + \epsilon_0\},\ 1 \leq j \leq M \tag{4.3.24}$$

where $\phi(x; \alpha, \mu, \sigma)$ is the Gaussian function with peak value α, mean μ and standard deviation σ, the minimum is over the three vertices of the j-th triangle

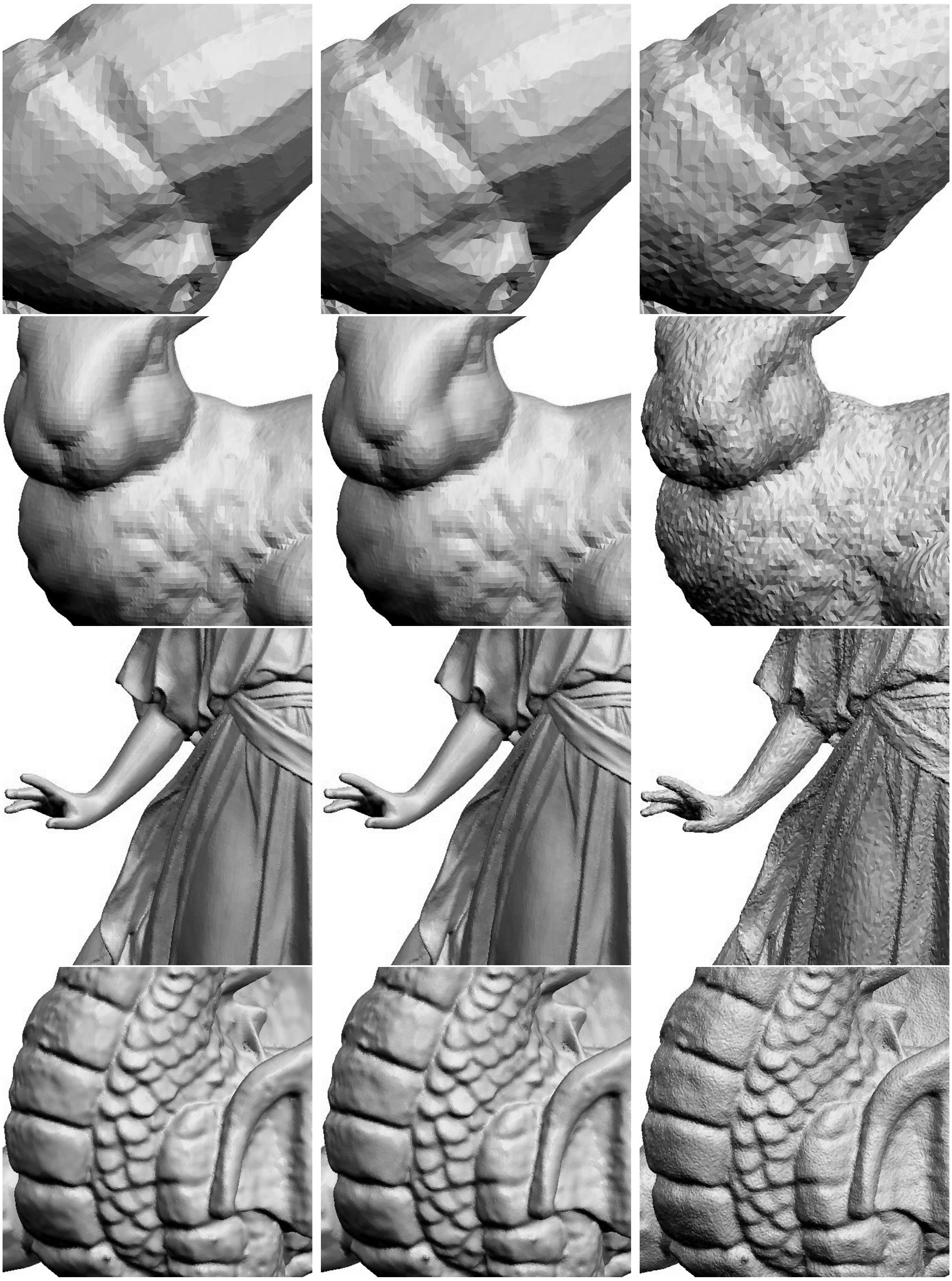


Figure 4.6: Close-ups of the quantized *Heart*, *Bunny*, *Lucy* and *Welsh Dragon* models in Fig. 4.5. The **left**, **middle** and **right** columns correspond to $\epsilon = 0.1°$, $\epsilon = 1°$ and $\epsilon = 10°$, respectively.

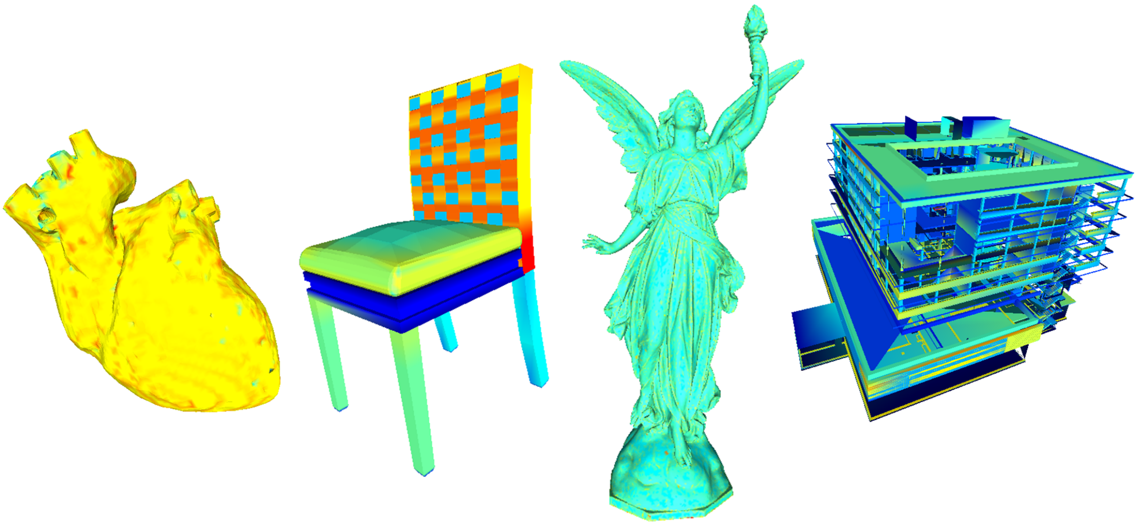


Figure 4.7: Vertices with high and low quantization levels are shown in red and blue color, respectively. From left to right: the ranges of the quantization levels are [6, 20], [4, 21], [12, 26] and [2, 32] bits per vertex coordinate.

and ϵ_0 is the threshold of normal degradation that is considered acceptable for a planar surface. The quantization level i of each triangle is then computed by

$$i = \arg\max_{k \in \mathbb{Z}} \{k \,|\, \bar{\mathrm{E}}_j(\theta) \leq \epsilon_j\} \tag{4.3.25}$$

Finally, the quantization level of a vertex is computed as the maximum level of quantization of the incident triangles.

Fig. 4.7 shows adaptive quantizations corresponding to parameter values $\alpha = 0.7979$, $\mu = 0$, $\sigma = 0.5$, $\epsilon_0 = 0.0017 \simeq 0.1^\circ$) and $\bar{\mathrm{E}}_j(\theta)$ approximated by $\mathrm{E}_{appr}(\theta)$. As expected the scanned meshes tend to have a more uniform and smooth distribution of quantization levels compared to the CAD meshes. Despite the higher cost of computing an adaptive, curvature depended quantization, the method is still local and thus, linear in complexity with respect to the size of the mesh.

4.3.2 Adaptive High-Capacity Data Hiding

The challenge in designing a good data hiding algorithm lies in balancing two conflicting requirements: high *embedding capacity* and low *embedding distortion*. The obtained relations between spatial and normal noise describe a trade-off between

Table 4.4: For each of $\epsilon = 0.1^\circ$, $\epsilon = 1^\circ$ and $\epsilon = 10^\circ$, the **left** and the **right** columns show the average embedding capacity in bits per vertex and the ratio of non-zero components in the corrective vector $\mathbf{e}$ to the number of mesh vertices, respectively.

	$\epsilon = 0.1^\circ$		$\epsilon = 1^\circ$		$\epsilon = 10^\circ$	
Heart	≈ 48.59	1.28×10^{-3}	≈ 58.40	1.22×10^{-2}	≈ 68.26	1.59×10^{-1}
Bunny	≈ 46.79	1.15×10^{-4}	≈ 56.14	1.37×10^{-2}	≈ 67.58	5.46×10^{-1}
Chair	≈ 41.26	3.25×10^{-2}	≈ 51.46	2.93×10^{-4}	≈ 60.66	1.27×10^{-1}
Lucy	≈ 39.85	1.62×10^{-3}	≈ 49.88	1.61×10^{-2}	≈ 59.74	1.54×10^{-1}
MPII	≈ 37.63	6.69×10^{-4}	≈ 47.60	2.61×10^{-2}	≈ 57.34	2.13×10^{-1}
Dragon	≈ 37.61	1.99×10^{-3}	≈ 47.00	2.77×10^{-4}	≈ 56.01	5.99×10^{-3}

embedding capacity, in the form of unused bits in the vertex coordinates, and visual distortion, in the form of normal degradation. As such, they can be directly used to inform least significant bit data hiding algorithms about the embedding capacity of the carrier for a given distortion tolerance.

To make the data hiding algorithm blind and robust against vertex permutation, we use two orderings $\pi_1(\cdot)$ and $\pi_2(\cdot)$ of the vertex set, which are obtained from the projections of the sets of unmarked $\mathbf{v}$ and marked $\mathbf{v}'$ quantized vertices onto their respective *first principal axes.* These two orderings will assist the extraction process in finding the embedding order of the message bits, avoiding thus potential synchronization problems. The details of the embedding and extraction algorithms are as follows.

Embedding: Given the carrier triangle mesh with set of vertices $\mathbf{v} = (v_1, v_2, \cdots, v_N)$, represented in 32 bits, and a normal degradation tolerance ϵ, we compute the vector $\mathbf{i} = (i_1, i_2, \cdots, i_N)$ of quantization levels as in Section 4.3.1. All unused vertex coordinate bits are set to zero. The message bits are embedded by a bit-replacement operation. More specifically, to mark the j-th vertex, we retain the i_j most significant bits and the c least significant bits of its coordinates and sequentially replace the other $32 - i_j - c$ bits with message bits. The embedding uses the vertex ordering $\pi_1(\mathbf{v})$. The vertex set obtained this way is denoted by $\mathbf{v}' = (v'_1, v'_2, \cdots, v'_N)$.

The c least significant bits of the vertex coordinates of $\mathbf{v}'$ will carry side information recording changes in the quantization levels between the vertices of the marked and unmarked meshes. In particular, we compute the vector of quantization levels $\mathbf{i}' = (i'_1, i'_2, \cdots, i'_N)$ of $\mathbf{v}'$ and from it and the original quantization levels $\mathbf{i}$ the corrective vector

$$\mathbf{e} = (i_1 - i'_1, i_2 - i'_2, \cdots, i_N - i'_N) \tag{4.3.26}$$

We sort the components of $\mathbf{e}$ according to the ordering $\pi_2(\cdot)$ and $\pi_2(\mathbf{e})$ is compressed and embedded into the c least significant bits of the vertex coordinates of $\mathbf{v}'$. The embedding follows the vertex ordering $\pi_2(\mathbf{v}')$. Notice that $\mathbf{e}$ can be compressed very efficiently, see Table 4.4. Thus, we can always use $c = 1$ and embed it in the least significant bits of the vertex coordinates.

Extraction: Given the marked mesh and the normal degradation tolerance ϵ, the key for extracting the message bits is to recover the orderings $\pi_1(\cdot)$ and $\pi_2(\cdot)$.

To compute $\pi_2(\cdot)$, we set the c least significant bits of the marked mesh to zero and quantize the resulting vertex set, which is a permutation of $\mathbf{v}'$, as in Section 4.3.1. The quantized vertices are projected onto their first principal axis. Notice that because the first principal axis is invariant under vertex permutation the ordering obtained from the projection is $\pi_2(\cdot)$. Having recovered $\pi_2(\cdot)$, the corrective vector $\pi_2(\mathbf{e})$ is obtained by extracting and decompressing the c least significant bits of the input marked mesh in the order of $\pi_2(\cdot)$.

To recover $\pi_1(\cdot)$, we first obtain the permuted quantized levels $\pi_2(\mathbf{i})$ of the original mesh by

$$\pi_2(\mathbf{i}) = \pi_2(\mathbf{e}) + \pi_2(\mathbf{i}') \tag{4.3.27}$$

Next, we quantize the vertices of $\pi_2(\mathbf{v}')$ at the original levels $\pi_2(\mathbf{i})$, obtaining the $\pi_2(\cdot)$ ordering of the unmarked quantized vertices. Using again the invariance of first principal axis under vertex permutation, the projection of the unmarked quantized vertices on their first principal axis gives $\pi_1(\cdot)$.

Having recovered $\pi_1(\cdot)$, we can now extract the message from the marked vertices $\pi_1(\mathbf{v}')$ and the quantization levels $\pi_1(\mathbf{i})$ of the unmarked vertices. To extract information from the j-th vertex, we just remove the i_j most significant bits and the c least significant bits of its coordinates and consider the remaining $32 - i_j - c$ bits

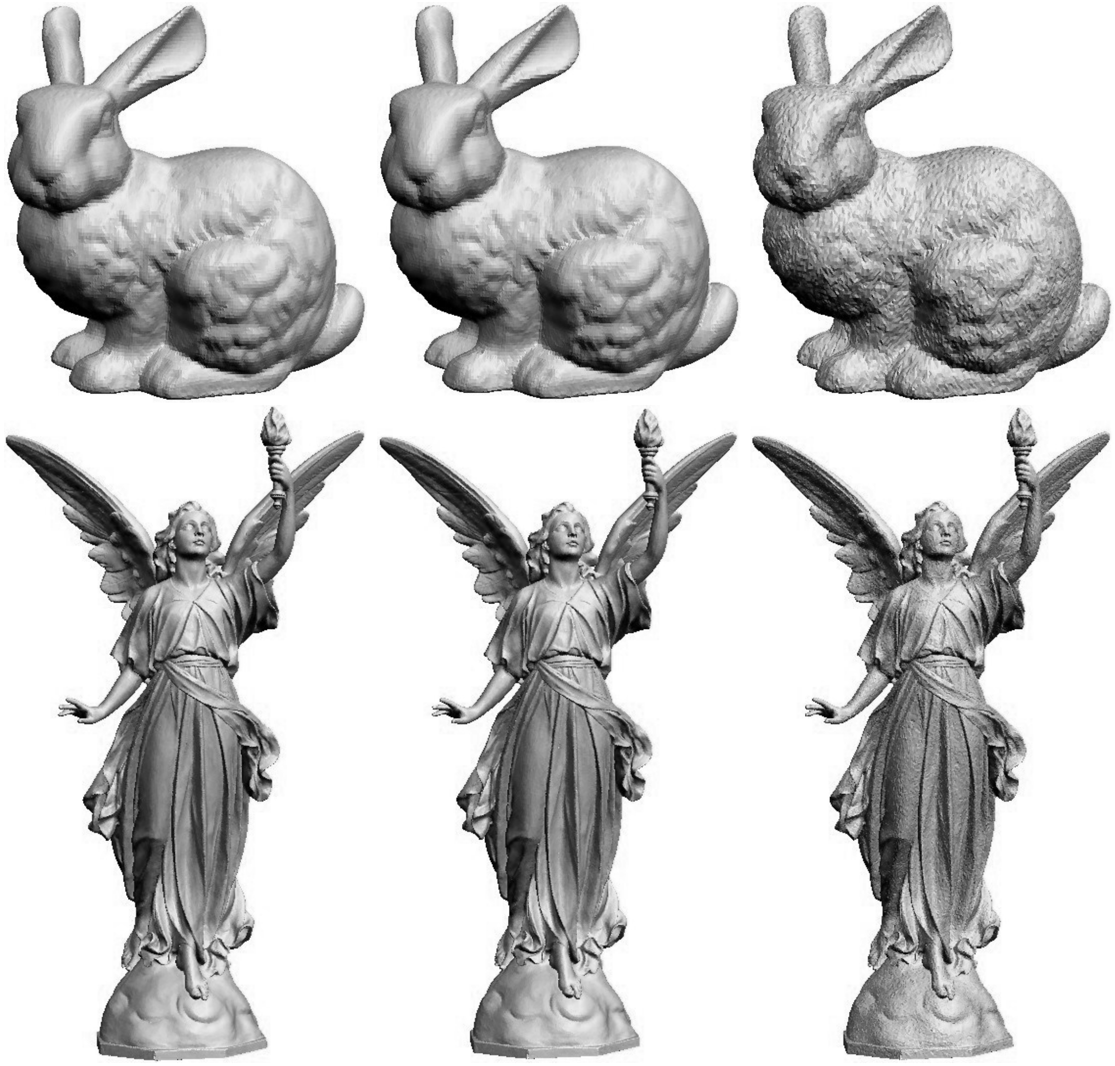


Figure 4.8: The **left**, **middle** and **right** columns show the marked meshes for $\epsilon = 0.1°$, $\epsilon = 1°$ and $\epsilon = 10°$, respectively.

as the embedded message.

Fig. 4.8 shows the models of *Bunny* and *Lucy* marked with various normal degradation tolerances ϵ. The quantization levels were computed using Eq. 4.3.25 with equal $\epsilon_j = \epsilon$ for all triangles. As expected, the meshes with lower degradation tolerance are of higher visual quality. In particular, the embedding distortion is almost invisible for $\epsilon = 0.1°$ and $\epsilon = 1°$, while it is clearly visible for $\epsilon = 10°$.

Table 4.4 reports the embedding capacity of each of the test meshes under various degradation tolerances, and the ratio of non-zero components in the corrective vector $\mathbf{e}$ to the number of mesh vertices. Capacity results for 16 or 24 bit quantizations of the same meshes, which often appear in practice, can be obtained from the results in Table 4.4 after subtracting 16 or 8 bits per coordinate, respectively, provided that these coarser quantizations do not degrade the normal beyond the prescribed tolerance.

We notice that, as expected, the capacity increases monotonically with ϵ, which implies that the extra capacity is achieved at the expense of increased embedding distortion. As expected, in terms of capacity the proposed method outperforms all of the previously reported algorithms [5, 6, 9, 12, 171] for hiding information in the geometry of a 3D mesh.

Discussion: The main advantages of the proposed data hiding algorithm are:

- It is a high-capacity method which, for $\epsilon = 1^\circ$, is able to embed at least 10 more bits per vertex than the current state-of-the-art for embedding information on mesh geometry [6].
- It gives the user control over the trade-off between capacity and embedding distortion. That is, we can embed the appropriate payload knowing that we do not violate the application's requirements on visual quality.

The main disadvantage of the algorithm is its fragility. Indeed, even slight attacks may lead to the loss of most of the embedded information. It is worth mentioning here that, to exactly recover the hidden message, the proposed algorithm requires fixed number of vertices with no requirement on mesh connectivity, because changing (increasing or decreasing) the number of vertex edges will not alter the order of message extraction, which is the same as that of message embedding.

4.4 Summary

We have studied the relationship between spatial and normal noise in a triangle mesh. We have explicitly computed the expected angle $\mathrm{E}(\theta)$ between the old and the new normal when a small amount of uniform random noise with cubic support of edgelength $2l$ is added to a single vertex. We have noticed that for small amounts of noise there is a linear correlation between $\mathrm{E}(\theta)$ and l/h which allows the development of simple heuristic methods for estimating E(θ) when noise is added to all three vertices of the triangle.

Based on the obtained results, two possible applications have been mentioned. As a first application, we have applied the normal perturbation estimators to compute optimal levels of dithered quantizations of the mesh vertices when a tolerance for the accuracy of the normals is given. As a second application, we have proposed an adaptive high-capacity data hiding algorithm for 3D triangle meshes. The algorithm computes for each vertex the appropriate dithered quantization level i, retains the i most significant bits of each vertex coordinate and hides the secret message into the less significant bits.

In the future, we plan a similar study of the relationship between spatial noise and the accuracy of the discrete curvatures of a mesh. Preliminary findings of this study are presented in Chapter 5. The aim, again, is to develop methods for estimating mesh quality when a dithered quantizer is applied to the vertex coordinates.

Chapter 5

Curvature Degradation of Triangle Meshes

The previous two chapters investigated the impact of changing the mesh vertex coordinates of 3D triangle models on face normals, providing an understanding of how spatial accuracy of mesh vertices relates to normal information. This chapter looks into the relation between spatial accuracy of mesh vertices and the Discrete Gaussian Curvature (DGC) defined on these vertices.

The strategy we follow for finding asymptotic results on the changes of DGC caused by spatial pertubations identifies three principal directions on the vertex v, and does approximate asymptotic computations independently for each of these three directions. The assumption is that any other direction of the spatial displacement vector will result to a behaviour of DGC change that will be asymptotically between these three cases. Our computations are based on several heuristic assumptions about the behaviour of DGC under vertex displacement. Moreover, the computations only cover the simplest case of adding a displacement vector on a single mesh vertex. To verify whether the computations generalize well when uniformly random displacement vectors are added to all mesh vertices, we experimented on a representative set of large irregular triangle meshes. The experimental results show the initial asymptotic computations to be valid regarding displacements in the normal direction, but invalid on tangential directions. We proposed a simple explanation for this singular behaviour of DGC under tangential noise and updated

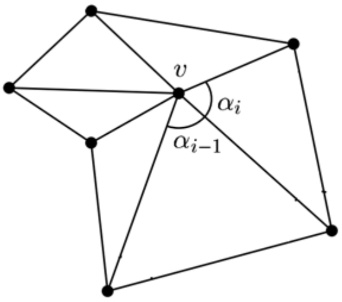


Figure 5.1: The discrete Gaussian curvature as the angle deficit around a vertex.

our initial assumptions.

The rest of this chapter is organized as follows. In Section 5.1, we present the results of the asymptotic computations related to the change of the DGC when a random displacement is added to a single vertex of the mesh. Section 5.2 experimentally evaluates the computations and our modelling assumptions. Finally, we briefly conclude in Section 5.3. Material in this chapter has been submitted to *Eighth International Conference on Mathematical Methods for Curves and Surfaces, 2012* for consideration.

5.1 Curvature Degradation

The most commonly used discretisation of the Gaussian curvature at a vertex v of a triangle mesh is defined as

$$\kappa_v = 2\pi - \sum_{i}^{N} \alpha_i \tag{5.1.1}$$

where α_i is the i-th angle adjacent to the vertex v and N is the number of triangles adjacent to v, or the valence of v (see Fig. 5.1). Our aim is to obtain an approximate computation of the expectation of κ_v when a uniformly random displacement is added on v. The main difficulty with such a computation is that if we treat every vertex in the 1-ring neighbourhood of v as an independent variable, then we have to do the computations in a $3N$-dimensional space.

To reduce the dimensionality of the problem, we first observe that the disc of the one ring-neighbourhood of v is a developable surface and the DGC is equal to $2\pi - \theta$, where θ is the angle of development at v. Our strategy is to approximate the

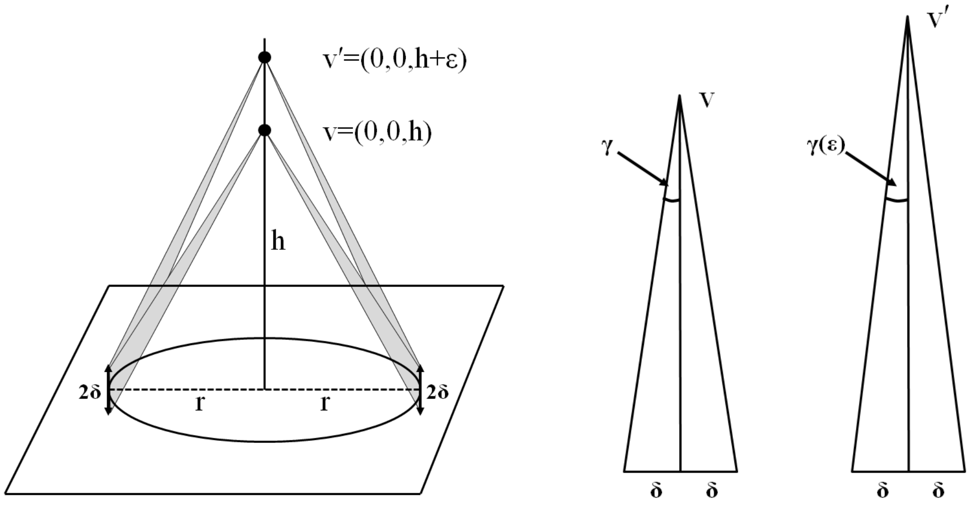


Figure 5.2: Vertex displacement in the normal direction. **Left:** The pair of diametrically opposed triangles we use to measure the change in the development angle of the elliptical cone. **Right:** The triangle before and after the displacement.

disc of the 1-ring neighbourhood with a simpler developable surface and compute the change in θ caused by the displacement of v on this simplification. For that purpose we only need to approximate the polygon of the 1-ring neighbours of v with a curve S and then compute the angle of development of the cone defined by the vertex v and the directrix S. In this chapter, the polygon is approximated by an ellipse $\mathcal{E}$, and thus θ becomes the angle of development of an elliptical cone. The two axes of $\mathcal{E}$ can be seen as the two principal directions of the boundary of the 1-ring neighborhood of v, and together with the normal of $\mathcal{E}$ form the three directions on which we do the computations. To further simplify the problem, we assume that the projection of v on the plane of $\mathcal{E}$ is on the centre of $\mathcal{E}$, that is, we have a right elliptical cone.

In the next step, we approximate $\mathcal{E}$ with a polygon with small edges of length δ. To simplify the problem even further, instead of considering triangles with one vertex v and a small base of length δ anywhere in $\mathcal{E}$, we consider a pair of triangles with the centres of their bases at the intersection of $\mathcal{E}$ with its major axis, see Fig. 5.2 (left). The heuristic justification for that simplification is that while the displacement of v will cause some of angles incident to v to get larger and other smaller, a pair of

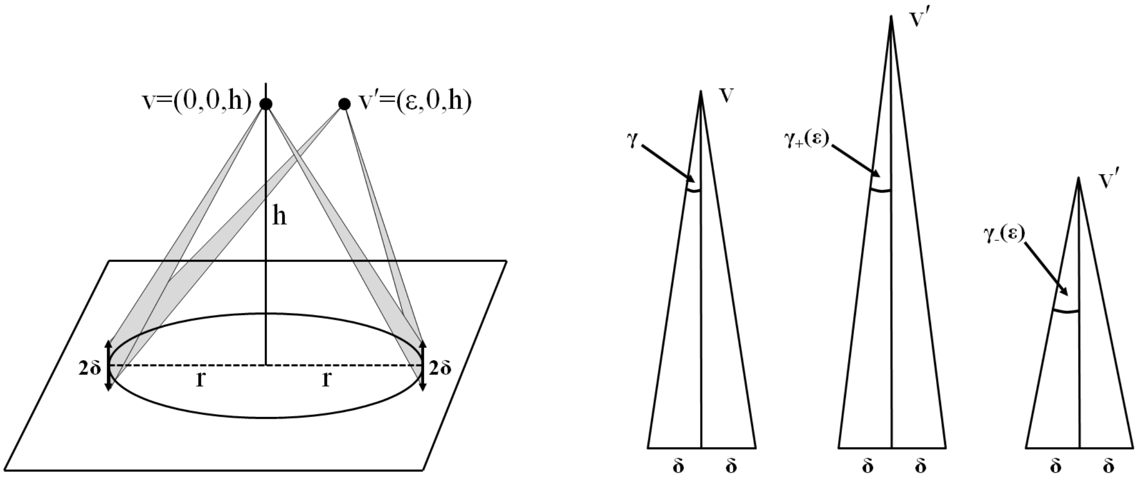


Figure 5.3: Vertex displacement in the direction of the major axis of $\mathcal{E}$. **Left:** The pair of diametrically opposed triangles we use to measure the change in the development angle of the elliptical cone. **Right:** The triangle before and the two triangles after the displacement.

diametrically opposite triangles will be enough to capture the asymptotic behavior of the change of θ.

In the case of displacement in the normal direction we notice that the configuration of the pair of triangles is symmetric and we only have to compute the difference $\gamma - \gamma(\epsilon)$, where γ is the original angle at v, while $\gamma(\epsilon)$ is the angle after the displacement of v, see Fig. reffig:chap5compNormal (right). After some straightforward but not trivial computations, we find that for small δ, asymptotically, the difference is given by

$$\gamma - \gamma(\epsilon) \ \sim \ \frac{4h}{r^2 + h^2}\gamma\epsilon \ \sim \ O(\epsilon). \tag{5.1.2}$$

The next case is when the displacement is parallel to the major axis of $\mathcal{E}$, see Fig. 5.3 (left). In this case there are two different triangles with one vertex v' and their base on $\mathcal{E}$. Thus, we have to compute $2\gamma - \gamma_+(\epsilon) - \gamma_-(\epsilon)$ for small δ, see Fig. 5.3 (right). Similarly to the first case, straightforward but not trivial computations show that

$$2\gamma - \gamma_+(\epsilon) - \gamma_-(\epsilon) \ \sim \ \frac{4r^2}{(r^2 + h^2)^2}\gamma\epsilon^2 \ \sim \ O(\epsilon^2). \tag{5.1.3}$$

The computations in the third direction, which is parallel to the minor axis of

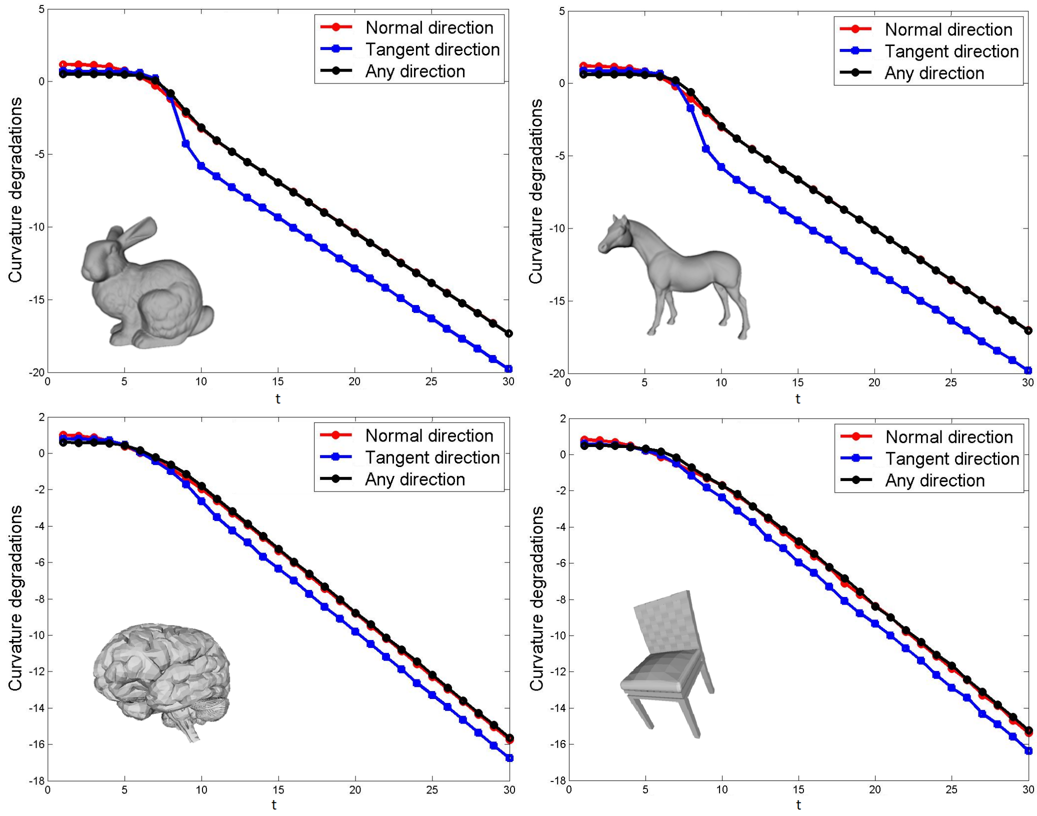


Figure 5.4: Experimental results of DGC degradation when the noise with strength level 2^{-t} is added to all the vertices of the *Bunny*, *Horse*, *Brain* and *Chair* models.

$\mathcal{E}$, are slightly more complicated but similar to the second case and result is again $O(\epsilon^2)$.

Regarding algorithmic complexity, the algorithm is linear to the number of mesh vertices.

5.2 Validation

First we want to validate the modelling assumptions and simplifications behind our computations. In a first experiment, we consider a regular right hexagonal pyramid with height and radius equal to one, we add several amounts of random noise either in the normal or in the tangential direction, measure the absolute value of the difference in the DGC, and for each amount of random noise we plot the average. We use a logarithmic scale, that is, we plot $\log_2 \bar{\kappa}_{dif}$, where $\bar{\kappa}_{dif}$ is the average of

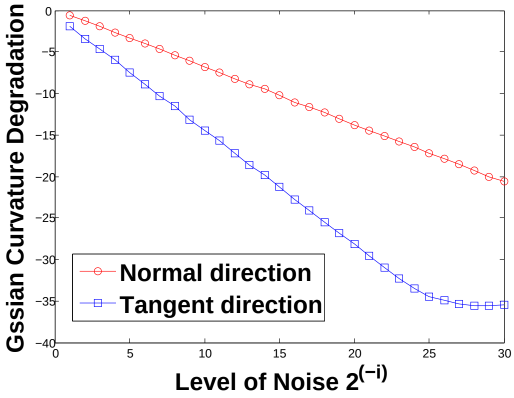


Figure 5.5: Experimental results of DGC degradation at the apex of a regular right hexagonal pyramid when the noise with strength level 2^{-t} is added to the apex.

the absolute value of the difference of the DGC.

From Fig. 5.4 it is clear that in both cases $\log_2 \bar{\kappa}_{dif}$ is linear, meaning that $\bar{\kappa}_{dif}$ is asymptotically polynomial. It should be noticed however that in the tangential case the behaviour of the model is subquadratic which means that the asymptotic expectation $O(\epsilon^2)$ has not been fully validated.

We also notice that the behaviour of the models is polynomials even for very small values of t, where we do not expect our modelling assumption to hold. We believe that this is a result of the high regularity of the pyramid. We also notice that for very large levels of quantisation, at the right end of Fig. 5.5, the curve levels off as a result of the limited accuracy of the machine arithmetic.

In the second experiment, we experimentally validate the proposed approximations, obtained in Section 5.1 by adding a random displacement to a single mesh vertex, on large irregular models with random displacements added to all vertices. The validation experiment was conducted on a set of synthetic and natural 3D triangle mesh models consisting of the *Bunny*, *Horse*, *Brain* and *Chair*.

Fig. 5.4 shows the experimental results of DGC degradation with respect to the various values of t when a uniformly random displacement with average strength level 2^{-t} is added along *normal*, *tangent* and *arbitrary* directions of the mesh vertices. The normal of a vertex was computed as the mean average of the normals of the triangles incident to that vertex. As expected from Eqs. 5.1.2 and 5.1.3, Fig. 5.4 shows that

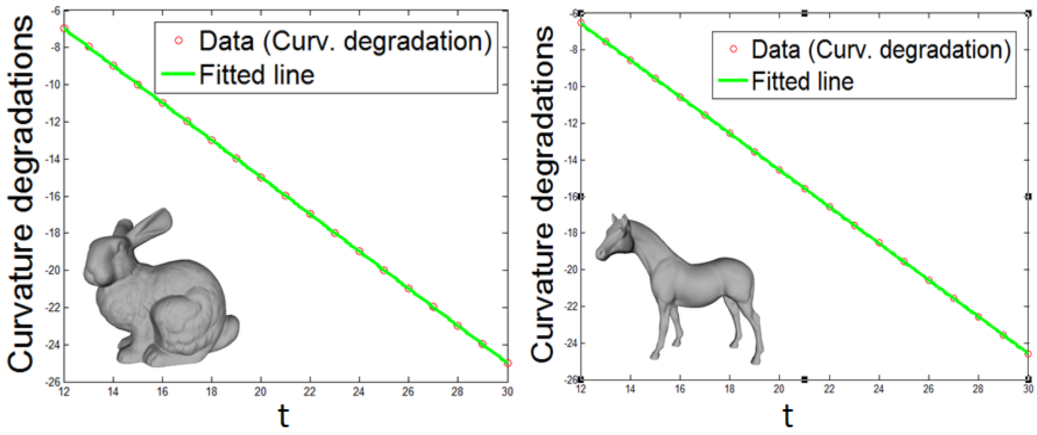


Figure 5.6: Fitting results for the *Bunny* and *Horse* models when the noise is added to the normal direction of the mesh vertices.

the change of the DGC is approximately linear to t when the average value of the random displacement 2^{-t} is small. We also notice that the DGC change after vertex displacement in any direction is almost equal to the DGC change after displacement in the normal direction. This was also expected given that the DGC change in the normal direction is asymptotically larger than these caused by displacements in the tangential directions.

However, we can also notice that the curve corresponding to displacements in the tangential direction, after a sharp drop at the coarse levels of quantisation, becomes a line parallel to that of the normal direction. That seems to suggest that Eq. 5.1.3 does not generalize when all mesh vertices are perturbed, or we would have obtained a line with a slope almost twice the slope of the line corresponding to the normal direction, that is, a behaviour similar to the one shown in Fig. 5.5. We believe that this phenomenon has a simple explanation, namely that the perturbation of the neighbours of the vertex v causes a slight perturbation of the tangent plane at v, and thus when a random displacement at the original tangential direction is added, in reality it also contains a small normal component. Eventually, that small normal component dominates the behaviour of the DGC change.

The implication of the above observation is that while vertex displacements in the normal and tangential directions have measurably different impact on the DGC, they both have the same asymptotic behaviour. In practice, that means that smoothing

Table 5.1: Mesh sizes and the parameters of the linear fitting model when noise is added to the normal direction of the mesh vertices.

	#Vertex	a	b
Bunny	34835	-1	5.05
Dragon	19851	-1	5.46
Brain	18375	-1	1.85
Chair	3413	-1	7.84

or a mesh evolution algorithms who move vertices in the tangential direction only have indeed a gentler impact on the mesh curvature as intended, however the gain in curvature accuracy is by a constant only amount of bits.

Fig. 5.6 shows least square fitted lines for the experimental results from the Bunny and the Horse, for the normal direction and for $l \geq 12$. We notice that this linear fit is very good. The figure suggests that the linear model for the expected curvature degradation

$$\bar{\kappa}_{dif} = -a \cdot t + b, \quad t \geq 0 \tag{5.2.4}$$

should be very accurate for sufficiently large levels of quantisation t. In Table 5.1 we report the numerical values of the a and b in Eq. 5.2.4 for the four test models, together with mesh sizes. Most characteristically, in the two decimal digits accuracy we used to report the results, the slope of all lines is exactly -1, as predicted by Eq. 5.1.2.

5.3 Summary

We studied the changes in the DCG of the vertices of a triangle mesh resulting from spatial random vertex displacements modelling a dithered quantiser. In the simplest case of adding a random displacement to a single vertex, even though the asymptotic computations are based on some strong simplifying assumptions, experimental validation shows them to be very accurate. Moreover, the results on the computed and the experimentally observed asymptotic behaviours seem to extend to the most interesting case of adding random displacements to all mesh vertices, even though in the case of displacements in the tangential direction this extension is not trivial.

The preliminary work presented in this chapter, naturally introduces two characteristic numbers associated with a triangle mesh. The first is the intercept of the linear model of DGC degradation caused by displacements at the normal or at any direction, that is, the value of b in Eq. 5.2.4. The second is the distance between the two parallel lines modelling DGC degradation in the normal and tangential directions, respectively.

Chapter 6

Histogram-based 3D Robust Watermarking

In this chapter, we propose a robust watermarking algorithm for triangle mesh models based on the modification of the histogram of vertex coordinates in a spherical coordinate system computed by applying PCA to the cover model. The histogram is invariant under transformations such as rotation, translation and vertex reordering, and hence our watermarking method is robust against these attacks. The proposed method carries out blind watermark extraction. Experimental results demonstrate the excellent performance of our method in terms of resistance to a wide range of attacks, as well as its superiority over similar existing methods.

The proposed method can been seen as a modification of the watermarking algorithm proposed by Cho et al. [7]. The main motivation behind the modified algorithm was the observation that the original Cho et al.'s algorithm is vulnerable to a specific steganalytic attack we developed while our new algorithm is designed to be resistant to that attack.

The main contributions of this chapter are:

- A watermarking approach robust against a wide range of attacks, based on modifying the histogram of the radial coordinate of the mesh vertices in a spherical coordinate system.
- A specific steganalytic algorithm against Cho et al.'s watermarking [7].

The main limitation of the first contribution is that the proposed watermarking is not resistant to mesh editing/deformation operations that change the global shape of mesh. Indeed, such operations could change the radial coordinates significantly and thus their histogram significantly. The main limitation of the second contribution is the limited scope of the proposed steganalytic algorithm. Indeed, it is a specific method and only works when applied to detect the messages hidden by Cho et al.'s watermarking.

The rest of this chapter is organized as follows. Section 6.1 describes in details the embedding process, while Section 6.2 briefly presents the extraction algorithm. Section 6.3 presents the experimental results, demonstrating the performance of our method. Finally, Section 6.5 concludes this chapter.

6.1 Embedding Process

Following Cho et al. [7], the system embeds the watermark in the spherical rather than the Cartesian coordinates. However, there are two main differences between the proposed embedding process and the original algorithm in Cho's et al. [7]. The first difference is that in our algorithm the origin of the spherical coordinate system is not the barycentre of the vertices. Instead, the origin is a point computed on the principal axis of the set of vertices, a choice that increases the variance of the radial coordinates of the mesh vertices, and thus, the robustness of the algorithm. The second and most important difference is the way the histogram of the radial coordinates is modified to embed message bits. Our choice of embedding technique makes the algorithm robust against the steganalytic attack we devised for breaking the original Cho et al. [7].

Next we describe the two stages of the proposed algorithm, first the coordinate system transformation and then the watermark embedding.

6.1.1 Coordinate System Transformation

Let (x_i, y_i, z_i) denote the Cartesian coordinates of the i-th vertex of a triangle mesh with N vertices. After computing the point $(\bar{x}, \bar{y}, \bar{z})$ and translating the origin of the coordinate system to that point, we compute the spherical coordinates $(\rho_i, \phi_i, \theta_i)$ of (x_i, y_i, z_i) by

$$\begin{aligned} \rho_i &= \sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2} \\ \phi_i &= \cos^{-1} \frac{z_i - \bar{z}}{\sqrt{(x_i - \bar{x})^2 + (y_i - \bar{y})^2 + (z_i - \bar{z})^2}} \\ \theta_i &= \tan^{-1} \frac{y_i - \bar{y}}{x_i - \bar{x}} \end{aligned} \tag{6.1.1}$$

where $1 \leq i \leq N$, $\rho_i \in [0, +\infty)$, $\phi_i \in [0, \pi]$, $\theta_i \in [0, 2\pi)$, see [172].

One obvious choice for $(\bar{x}, \bar{y}, \bar{z})$ is the barycentre of the vertex set of the original model as in [7]. However, the set of radial coordinates $\mathcal{P}$ resulting from this choice of origin might not be a satisfying message carrier when watermarking 3D models resembling a uniformly sampled sphere. Indeed, the barycentre of a uniform sample from a sphere is the centre of that sphere and the variance of $\mathcal{P}$ is zero. As a result, for models resembling a uniformly sampled sphere, which are very common in practice, the variance of $\mathcal{P}$ will be very low, affecting the robustness of the watermarking method.

Instead, we are looking for a point $(\bar{x}, \bar{y}, \bar{z})$ as the origin of the spherical coordinate system that will result to a set of radial coordinates $\mathcal{P}$ with high enough variance to allow for robust watermarking, but not excessively high variance which might increase the distortion of the marked model when the watermarking algorithm changes the radial coordinate of mesh vertices to alter the histogram of $\mathcal{P}$. Moreover, if we want the watermaking algorithm to the able to withstand a certain attack, the computation of the origin should also be robust against that attack.

The proposed choice for computing the origin is based on averaging vertices projected on the principal, an operation that is invariant under common affine transformations and reasonably stable under small vertex perturbations. More specifically, we project the mesh vertices onto their principal axis and then compute $(\bar{x}, \bar{y}, \bar{z})$ as the average of the half of the vertex set lying at the most right-hand side of the principal axis. Notice that by averaging the right most half of the verices on the principal axis we shift the origin away from the barycentre and, generally, we increase the variance of $\mathcal{P}$. We have empirically found that the proposed averaging, which in a regular setting places the origin in a position splitting the projection of the mesh on its principal axis at about three quarters, gives a high but not excessively high variance of $\mathcal{P}$, resulting thus in a good trade-off between robustness and distortion for the watermarking algorithm.

6.1.2 Watermark Embedding

The proposed algorithm embeds the watermark by changing the histogram of $\mathcal{P}$. To embed a watermark bit $w_i \in \{-1, +1\}$, we take two neighboring bins of the histogram and possibly transfer some elements from one bin to another. The details of the embedding process are described as follows.

Step 1: The first step is to build a histogram with K bins $\mathcal{B} = \{\mathcal{B}_k : 1 \leq k \leq K\}$ for $\mathcal{P} = \{\rho_i : 1 \leq i \leq N\}$. To do so, we construct $\mathcal{B}_k$ by classifying all the elements $\rho_i \in \mathcal{P}$ into the K bins based on

$$\mathcal{B}_k = \{\rho_i : \rho_{\min} + (k-1) \cdot \Delta_\rho \leq \rho_i < \rho_{\min} + k \cdot \Delta_\rho\} \tag{6.1.2}$$

where $\rho_{\min}$ and $\rho_{\max}$ are the minimum and the maximum of $\mathcal{P}$, and

$$\Delta_\rho = (\rho_{\max} - \rho_{\min})/K \tag{6.1.3}$$

is the range size of each bin. We also assume that $\rho_{\max} \in \mathcal{B}_K$. The histogram of $\mathcal{P}$ is produced by counting the number of elements in each bin $\mathcal{B}_k$.

Step 2: In this step, we insert watermark message bits by modifying the histogram computed in Step 1. Starting from the second bin $\mathcal{B}_2$, we arrange any two adjacent bins into pairs as $(\mathcal{B}_2, \mathcal{B}_3), (\mathcal{B}_4, \mathcal{B}_5), \cdots$ and then proceed to hide a watermark bit w_i into each *embeddable* pair. A pair $(\mathcal{B}_k, \mathcal{B}_{k+1})$ is considered embeddable if

$$|\mathcal{B}_k| + |\mathcal{B}_{k+1}| \geq 1 \tag{6.1.4}$$

where $|x|$ denotes the number of elements of the set x.

Notice that the proposed method does not utilize the bins $\mathcal{B}_1$ and $\mathcal{B}_K$ to carry watermark. As it will become apparent later, this ensures that the watermark can be extracted with no reference to the original model. Since $\mathcal{B}_1$ and $\mathcal{B}_K$ are excluded from watermarking and since one pair of bins is required to carry one message bit, if the bin number K is odd, we need to exclude one more bin from the embedding process, here, we choose $\mathcal{B}_{K-1}$.

A watermark bit w_i is embedded in an embeddable pair $(\mathcal{B}_k, \mathcal{B}_{k+1})$ by increasing the values of some radiuses $\rho_i \in \mathcal{B}_k$, or decreasing the values of some radiuses $\rho_i \in \mathcal{B}_{k+1}$, depending on the value of w_i. Specifically, to insert $w_i = -1$, we increase the values of the n_{mov} largest elements ρ_i of $\mathcal{B}_k$, pushing them into $\mathcal{B}_{k+1}$ through

$$\rho_i' = \rho_{\min}^{k+1} + \frac{\Delta_\rho}{\arg\min_n \{n : \rho_{\min}^{k+1} + \Delta_\rho / n < \rho_{\max}^{k+1}, n \in \mathbb{N}, n \geq 3\}} \tag{6.1.5}$$

where ρ_i' is the vertex radius in the marked mesh corresponding to ρ_i and $\rho_{\min}^{k+1}$ and $\rho_{\max}^{k+1}$ denote the minimum and the maximum radiuses in $\mathcal{B}_{k+1}$. The basic idea is to try to create a histogram difference between $\mathcal{B}_k$ and $\mathcal{B}_{k+1}$ by updating n_{mv} radiuses ρ_i in $\mathcal{B}_k$. That is, we attempt to achieve

$$|\mathcal{B}'_{k+1}| - |\mathcal{B}'_k| \geq n_{\text{thr}} \tag{6.1.6}$$

for the marked bins $\mathcal{B}'_k$ and $\mathcal{B}'_{k+1}$, where $n_{\text{thr}} \geq 1$ is an integer threshold controlling the tradeoff between robustness and distortion. We separate the following three cases:

Case 1: If $|\mathcal{B}_{k+1}| - |\mathcal{B}_k| \geq n_{\text{thr}}$, then $n_{\text{mov}} = 0$, meaning no alteration is required.

Case 2: Else if $|\mathcal{B}_k| + |\mathcal{B}_{k+1}| < n_{\text{thr}}$, then $n_{\text{mov}} = |\mathcal{B}_k|$, meaning all the radiuses in $\mathcal{B}_k$ will be transferred into $\mathcal{B}_{k+1}$ by updating them according to Eq. 6.1.5.

Case 3: Else if $|\mathcal{B}_k| + |\mathcal{B}_{k+1}| >= n_{\text{thr}}$, then n_{mov} is given by

$$n_{\text{mov}} = \left\lceil (|\mathcal{B}_k| - |\mathcal{B}_{k+1}| + n_{\text{thr}})/2 \right\rceil \tag{6.1.7}$$

Notice that by choosing to move the largest elements of $\mathcal{B}_k$ into $\mathcal{B}_{k+1}$, we keep the embedding distortion to a minimum.

The process of embedding $w_i = +1$ over an embeddable pair $(\mathcal{B}_k, \mathcal{B}_{k+1})$ is completely analogous. We create a histogram difference between $\mathcal{B}_k$ and $\mathcal{B}_{k+1}$ by decreasing if needed the values of the n_{mv} smallest radiuses in $\mathcal{B}_{k+1}$.

For all non-embeddable pairs $(\mathcal{B}_k, \mathcal{B}_{k+1})$, we do not embed any watermark bits into them and hence keep their radiuses ρ_i intact.

6.2 Extraction Process

Given a watermarked mesh model, the watermark extraction process is straightforward and can be carried with no reference to the original mesh.

From the given marked mesh, we compute the set of watermarked radiuses $\mathcal{P}' = \{\rho_i' : 1 \leq i \leq N\}$ using Eq. 6.1.1. After classifying the radiuses in $\mathcal{P}'$ into K bins using Eq.6.1.2 and 6.1.3, we obtain the set of watermarked bins $\mathcal{B}' = \{\mathcal{B}_k' : 1 \leq k \leq K\}$. Notice that, given K, the range size of each watermarked $\mathcal{B}_k'$ is the same as the range size Δ_ρ of the original bins in (see Eq. 6.1.3). This is because the fist and the last bins $\mathcal{B}_1$ and $\mathcal{B}_K$ have not been altered by the watermarking process, and thus the minimum $\rho_{\min}'$ and maximum $\rho_{\max}'$ radiuses after watermarking are equal to the minimum $\rho_{\min}$ and maximum $\rho_{\max}$ radiuses before watermarking, giving an efficient solution to the synchronization problem. However, notice that the number of bins K should be passed to the extraction algorithm as side information.

In the next step, shadowing the embedding process, the extraction algorithm forms the pairs of bins $(\mathcal{B}_2', \mathcal{B}_3'), (\mathcal{B}_4', \mathcal{B}_5'), \cdots$ and identifies the *embeddable* pairs $(\mathcal{B}_k', \mathcal{B}_{k+1}')$ that carry watermark messages. Recall that embeddable are the pairs satisfying $|\mathcal{B}_k| + |\mathcal{B}_{k+1}| \geq 1$. We find the corresponding pairs watermarked pairs $(\mathcal{B}_k', \mathcal{B}_{k+1}')$ by checking if

$$|\mathcal{B}_k'| + |\mathcal{B}_{k+1}'| \geq 1 \tag{6.2.8}$$

based on the invariance of the number of elements in a bin pair during watermarking

$$|\mathcal{B}_k'| + |\mathcal{B}_{k+1}'| = |\mathcal{B}_k| + |\mathcal{B}_{k+1}| \tag{6.2.9}$$

which is a consequence of the embedding strategy that swaps radiueses inside pairs of bins only.

Finally, each embedded watermark bit w_i' is sequentially extracted from each embeddable pair $(\mathcal{B}_k', \mathcal{B}_{k+1}')$ by

$$w_i' = \begin{cases} -1 & \text{if } |\mathcal{B}_{k+1}'| \geq |\mathcal{B}_k'| \\ +1 & \text{otherwise} \end{cases} \tag{6.2.10}$$

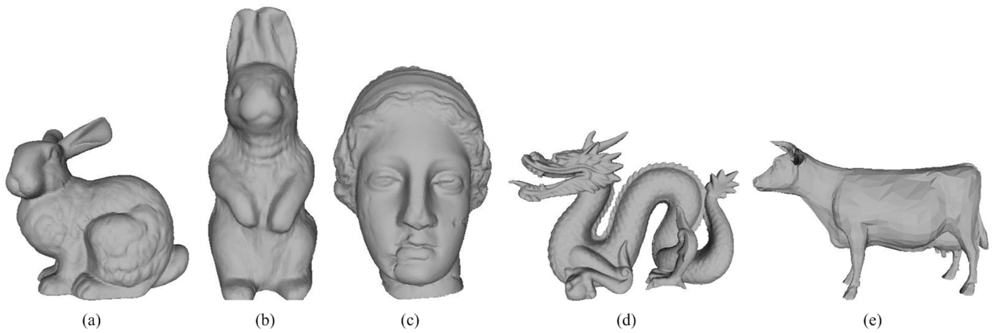


Figure 6.1: Original cover mesh models: (a) Bunny, (b) Rabbit, (c) Venus, (d) Dragon and (e) Cow.

Table 6.1: Model details and parameter setting.

Model	#Vertices	#Faces	n_{thr}
Bunny	34835	69666	43
Rabbit	70658	141312	80
Venus	100759	201514	100
Dragon	50000	100000	75
Cow	2904	5804	8

6.3 Experimental Results

In this section, we study the performance of the proposed watermarking algorithm and compare it against the variant of Cho et al.'s algorithm based on shifting the mean values of the radiuses in a bin. The evaluation includes a *distortion analysis*, a *robustness analysis* against various attacks and a study of its *steganalytic properties*. In the tests, we use a small representative set of well-known 3D models, consisting of the *Bunny*, *Rabbit*, *Venus*, *Dragon* and the *Cow*, see Fig. 6.1. The parameters used are listed in Table 6.1. The algorithm has linear time complexity $\mathrm{O}(n)$ as the runtime grows linearly as the vertex number n increases.

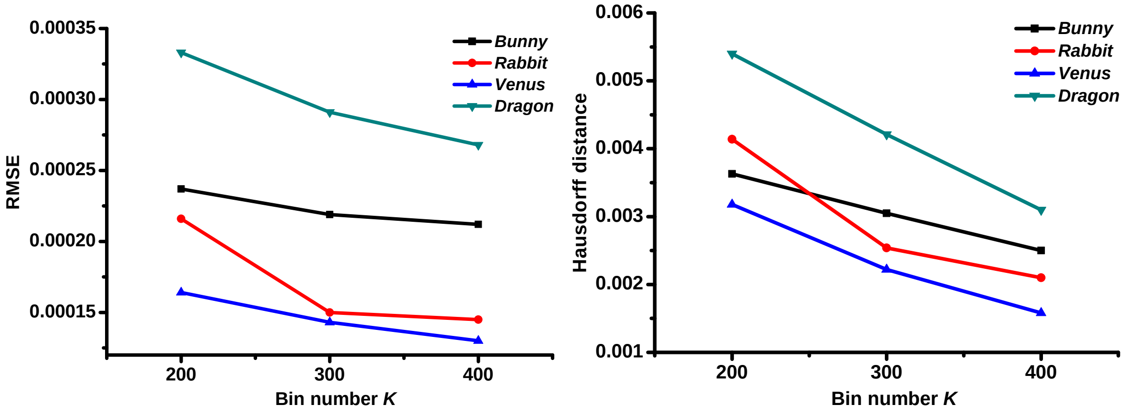


Figure 6.2: Embedding distortion measured as the RMSE (left) and the Hausdorff distance (right) for 200, 300 and 400 bins.

6.3.1 Distortion Analysis

For measuring the amount of distortion caused by the embedding of the watermark we utilize two widely used quantitative measures: the root mean square error (RMSE) and the Hausdorff distance between the original and the marked mesh. We compute these two measures with the Metro tool [173]. Notice that this subsection is only concerned with distortion analysis, that is, we do not apply any attacks to the watermarked models.

Fig. 6.2 plots the RMSE and the Hausdorff distance for various values of the bin number K. We notice that both measures of distortion are small, indicating that the proposed watermarking introduces a small only amount of degradation to the carrier model. We also notice that an increase of K results to an increase of the RMSE and the Hausdorff distance, that is, to larger amounts of distortion. The reason for this is that by increasing K we decrease the bin step Δ_ρ (see Eq. 6.1.3) and hence decrease the modulation of the radiuses ρ_i when they are transferred from one bin to an adjacent.

To gauge the visual significance of these distortion measurements, in Fig. 6.3 we show several marked mesh models corresponding to $K = 400$. Any distortion caused by the insertion of the watermark is hardly noticeable; however, after zooming in we can observe some artifacts in the smooth areas of the mesh. Fig. 6.4 show close-ups

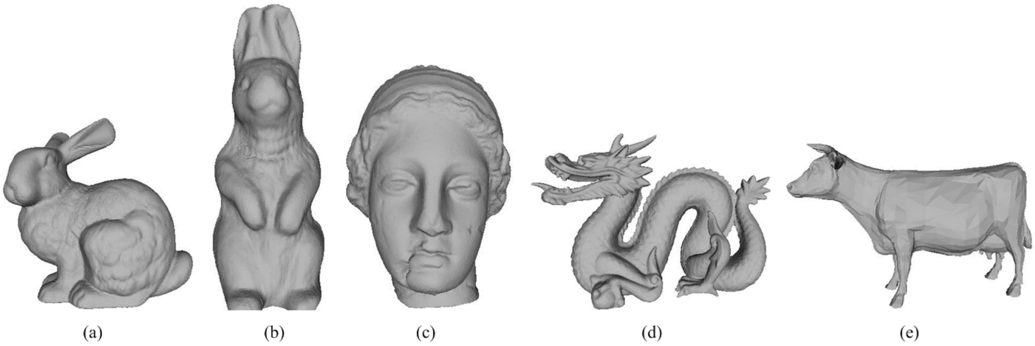


Figure 6.3: Watermarked mesh models under $K = 400$: (a) Bunny, (b) Rabbit, (c) Venus, (d) Dragon and (e) Cow.

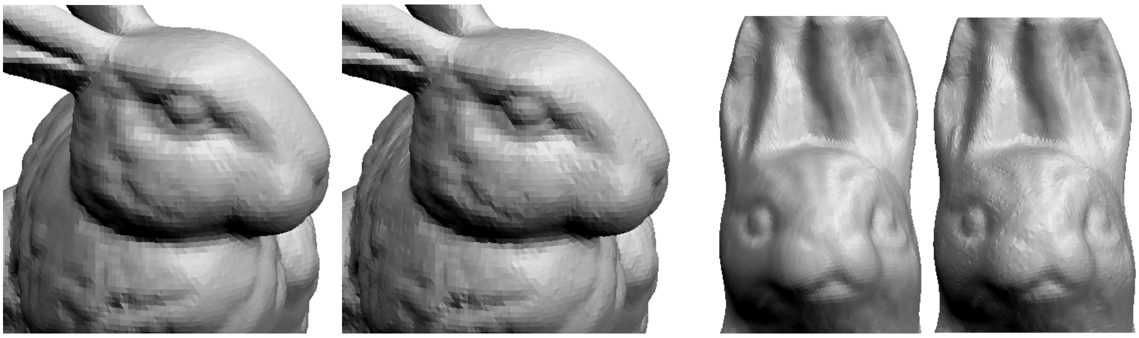


Figure 6.4: Close-ups of the original and watermarked *Bunny* and *Rabbit* models. For each mesh group, the left and right figures show the original and marked models, respectively. The watermarked models are from Fig. 6.3.

of the watermarked *Bunny* and *Rabbit* in Fig. 6.3.

Regarding the embedding capacity, which can be traded-off for distortion, we notice that the maximum length of the watermark bit sequence is $\lfloor (K-2)/2 \rfloor$, however for large values of K this maximum capacity is usually unachievable. The reason is that some bin pairs $(\mathcal{B}_k, \mathcal{B}_{k+1})$ do not contain any radiuses, i.e., $|\mathcal{B}_k| + |\mathcal{B}_{k+1}| = 0$, making the pair non-embeddable.

6.3.2 Robustness Analysis

The proposed method is obviously robust against distortionless attacks such as vertex reordering, translation, rotation, uniform scaling and their combinations because

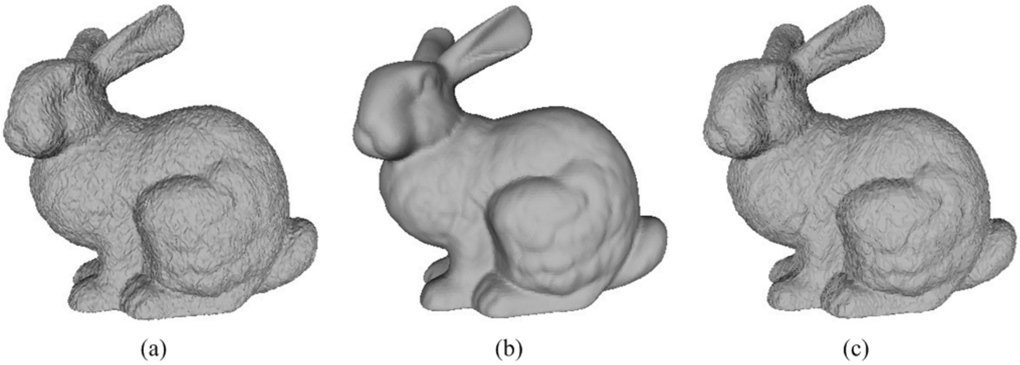


Figure 6.5: Results of attacking the marked *Bunny* model by: (a) adding $A = 0.50\%$ noise, (b) Laplacian smoothing of 50 iterations ($\lambda = 0.02$) and (c) 8-bit quantization.

the histogram of the radiuses ρ_i is invariant under these attacks. For analyzing the robustness of the watermark against malicious attacks such as noise addition, smoothing and quantization, we use the standard measure of the correlation coefficient

$$\mathcal{C}(\mathbf{w}, \mathbf{w}') = \frac{\sum_i (w_i - \bar{w}) \cdot (w'_i - \bar{w}')}{\sqrt{\sum_i (w_i - \bar{w})^2 \cdot \sum_i (w'_i - \bar{w}')^2}} \tag{6.3.11}$$

where $\bar{w}$ and $\bar{w}'$ denote the means of the original watermark sequence $\mathbf{w}$ and the detected watermark sequence $\mathbf{w}'$, respectively.

We performed these using the 3D mesh watermarking benchmark [174]. In the tests, we fixed $K = 400$ and carried out attacks with varying strength. Fig. 6.5 shows a marked Bunny model under various attacks.

Noise Addition: To measure the robustness under noise addition attacks, pseudo-random additive noise was added to the vertex coordinates (x_i, y_i, z_i) of each test model according to (resp. y_i, z_i)

$$x'_i = x_i + a_i \cdot \bar{d} \tag{6.3.12}$$

where $\bar{d}$ denotes the average distance from vertices to object center, and a_i is the noise strength and is a pseudo-random number uniformly distributed in interval $[A, A]$ with A the maximum noise strength. The four levels of noise strength used in the test are: $A = 0.05\%$, 0.10%, 0.25%, 0.50%. For each level, we conducted five experiments with different seeds, generating five different noise patterns and we report the average of the five results in Fig. 6.6.

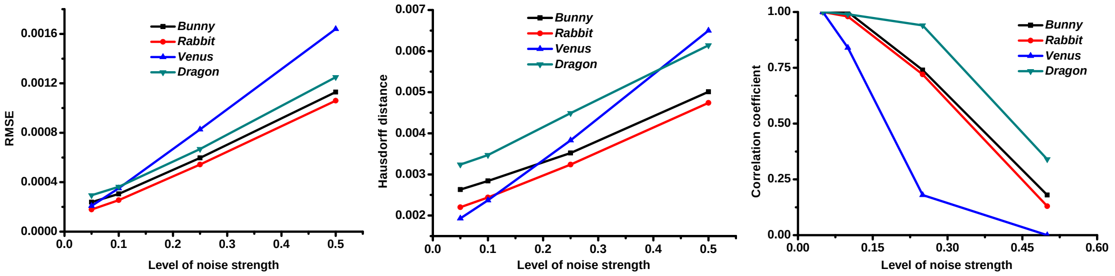


Figure 6.6: Robustness against different levels of noise attack (%). The figures on the left and middle plot the distortion of the marked models as a result of adding noise to them. The figure on the right plots the correlation coefficients $\mathcal{C}(\mathbf{w}, \mathbf{w}')$, where $\mathbf{w}'$ is the message extracted from the noisy marked model.

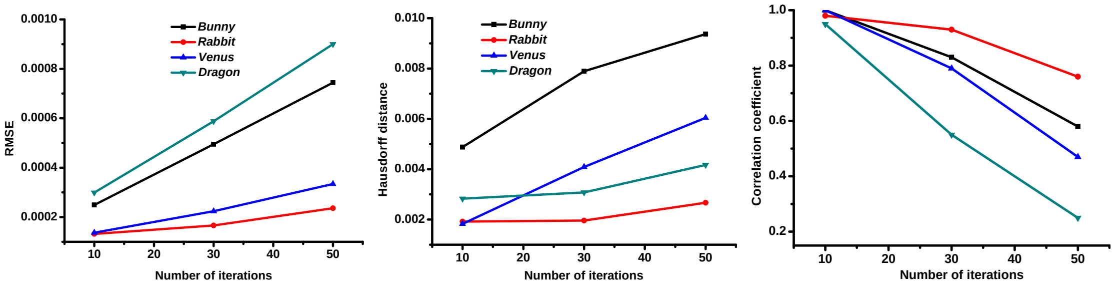


Figure 6.7: Robustness against different levels of smoothing attack. The figures on the left and middle plot distortion of the marked models as a result of smoothing them. The figure on the right plots the correlation coefficients $\mathcal{C}(\mathbf{w}, \mathbf{w}')$, where $\mathbf{w}'$ is the message extracted from the smoothed marked model.

As expected, in each model the inserted watermark degrades gradually as the noise strength increases. Even for strength levels as high as 0.25%, the correlation $\mathcal{C}(\mathbf{w}, \mathbf{w}')$ has a value that is larger than 0.70 for all the test models expect *Venus*. Notice that a noise attack at level 0.25% will degrade significantly the visual quality of the carrier model, meaning that even though the watermark may be removed, the model will be of no use for the attacker.

We also notice that the *Dragon* model appears to be exceptionally robust against the noise attack as $\mathcal{C}(\mathbf{w}, \mathbf{w}') \approx 0.95$ even when the noise level reaches $A = 0.25\%$. We believe that this is due to the fact that *Dragon* is the most 'complex' amongst these test models and the histogram of the radiouses has the highest variance.

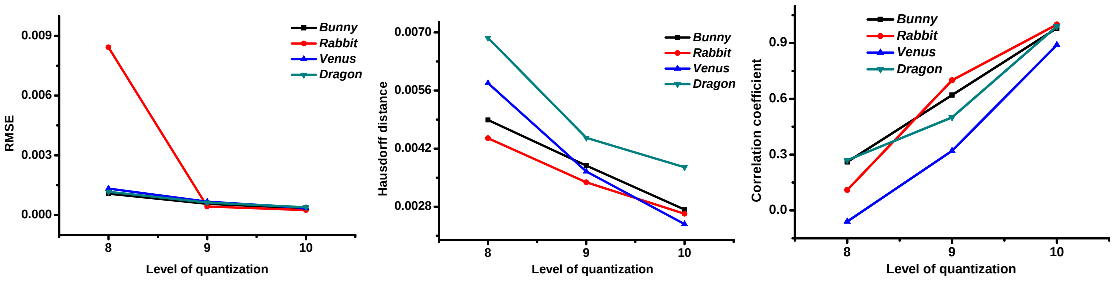


Figure 6.8: Robustness against quantization attack. The figures on the left and middle plot the distortion of the marked models as a result of quantizing them. The figure on the right plots the correlation coefficients $\mathcal{C}(\mathbf{w}, \mathbf{w}')$ for various levels of quantization, where $\mathbf{w}'$ is the message extracted from the quantized marked model.

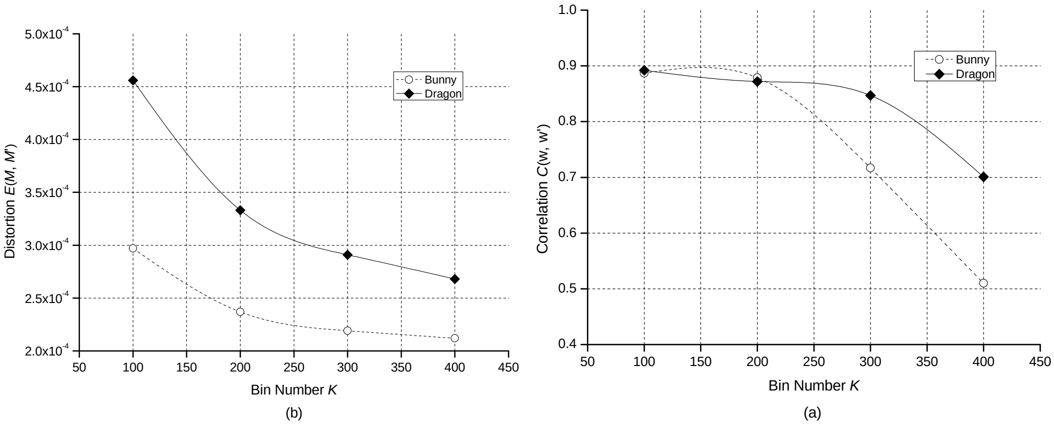


Figure 6.9: **Left:** The relationship between the embedding distortion measured by RMSE and the bin number K. **Right:** The relationship between the robustness measured by the correlation coefficient and the bin number K. The robustness was measured against a noise attack of noise level $A = 0.30\%$.

Smoothing Attack: To evaluate robustness against smoothing attacks, we applied to the marked models 10, 30 and 50 iterations of Laplacian smoothing [175], fixing the deformation factor at $\lambda = 0.02$. The results are shown in Fig. 6.7. Again, the high values of the correlation coefficients $\mathcal{C}(\mathbf{w}, \mathbf{w}')$ imply that the proposed watermarking method is able to survive mesh smoothing as well.

Quantization Attack: To evaluate robustness against quantization attacks, we quantized the Cartesian coordinates of the marked models at 8, 9 and 10 bits and

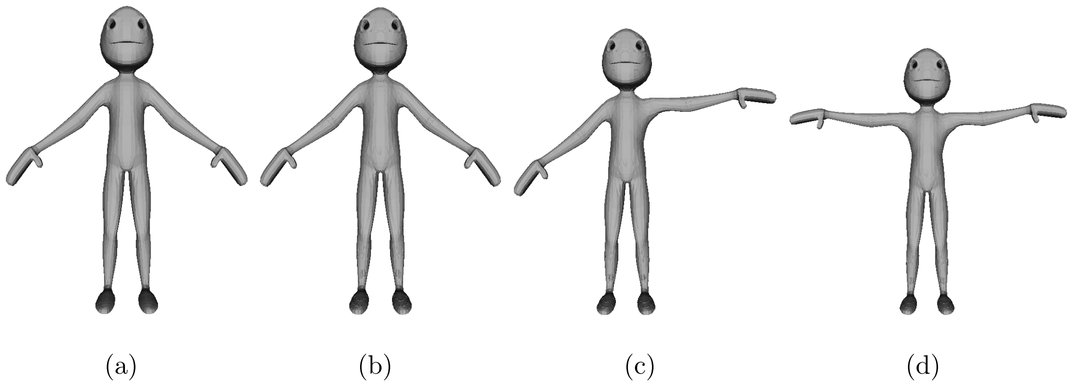


Figure 6.10: Illustration of the original Hank, the watermarked Hank and its deformed versions. (a) original Hank, (b) watermarked Hank, (c) deformation of (b), and (d) deformation of (b).

tried to retrieve the embedded watermark $\mathbf{w}'$ from the quantized models. Fig. 6.8 shows the correlation coefficients. As expected, the robustness decreases with the level of quantization because a lower level quantization degrades more the model.

One limitation of the proposed watermarking is its weak robustness against mesh deformation attacks that change the global shape of a 3D model. Fig. 6.10 shows the original model *Hank* from *Blender 2.49* [176], the watermarked *Hank* and the deformed versions of the marked *Hank*. While we are able to recover the embedded watermark bits perfectly from the marked *Hank* in Fig. 6.10 (b), we cannot extract the message bits from the deformed versions Figs. 6.10 (c) and (d) of Fig. 6.10 (b), i.e., we obtain the correlation coefficients $\mathcal{C}(\mathbf{w}, \mathbf{w}') = -0.029$ for Fig. 6.10 (c) and $\mathcal{C}(\mathbf{w}, \mathbf{w}') = 0.225$ for Fig. 6.10 (d). This is expected, because such a mesh deformation alters the model significantly and thus the distances between the mesh vertices and the model center change greatly. Given this limitation, we proposed a Laplacian coordinates-based 3D watermarking in Chapter 7 able to survive mesh deformation attacks.

6.3.3 Robustness w.r.t Bin Number K

In this subsection, we briefly study the relationship between the robustness and the bin number K. Fig. 6.9 shows that both the embedding distortion measured by

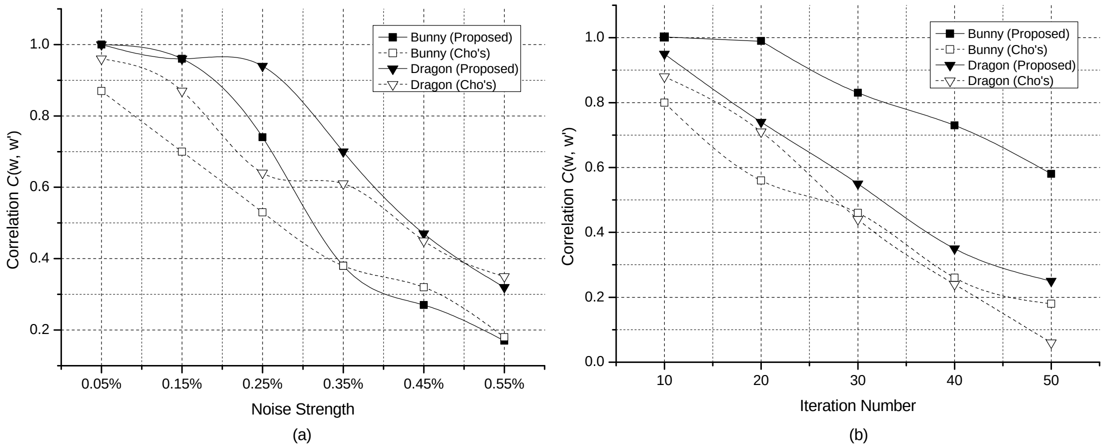


Figure 6.11: Experimental comparison between Cho et al.'s method and the proposed method. (a) noise addition attack and (b) Laplacian smoothing attack ($\lambda = 0.02$).

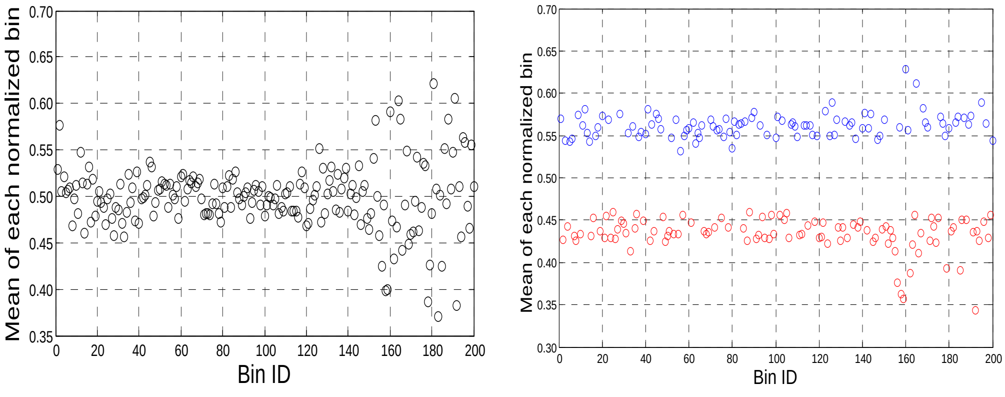
RMSE (left) and the correlation coefficients (right) decrease as K increases. That is, while the use of a smaller K reduces the distortion, it also decreases the robustness of watermark. This was expected given that distortion and robustness are conflicting requirements.

6.3.4 Comparison with Cho et al.'s method

Finally, we compare the proposed watermarking scheme with the mean based variant Cho et al. [7] in terms of robustness. Fig. 6.11 (left) compares the robustness of the two methods against noise addition attacks. Generally, the proposed methods exhibits better robustness properties than Cho et al.'s against this type of attack. Fig. 6.11 (right) compares the robustness of the two methods against smoothing attacks. Again, the proposed method seems to perform better. In both experiments, for Cho et al.'s method we used the parameter settings recommended in [7].

6.4 Anti-steganalysis Properties

In this section, we present a steganalytic algorithm we developed for Cho et al.'s mean based scheme and show that the proposed modification exhibits a superior



(a) (b)

Figure 6.12: Scatter plot of the mean values m_k for (a) the clean *Bunny* and (b) the watermarked *Bunny* with $K = 200$ bins.

anti-staganalytic behaviour against that method.

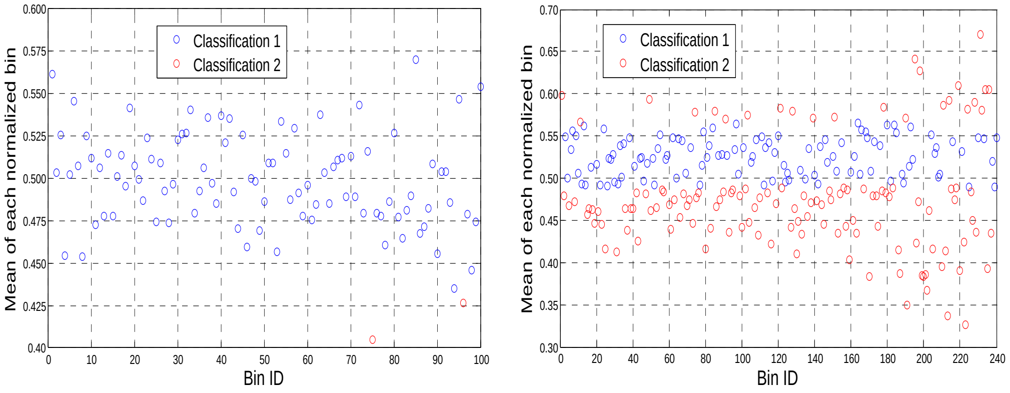
Let $\tilde{\mathcal{B}}'_k = \{\tilde{\rho}_{k,j} : j = 1, 2, 3, ...\}$ denote the k-th $(1 \leq k \leq K)$ bin of the normalized marked vertex norms $\tilde{\rho}_{k,j}$ in Cho et al.'s watermarking. The main idea of the steganalytic algorithm is based on the observation that the watermark embedding will result in a 2-clustering of the mean values $\bar{m}_k$

$$\bar{m}_k = \frac{1}{|\mathcal{B}'_k|} \sum_j \tilde{\rho}_{k,j} \tag{6.4.13}$$

of the bins $\tilde{\mathcal{B}}'_k$. As an example, Fig. 6.12 illustrates the scatter plot of $\{\bar{m}_k : 1 \leq k \leq K\}$ for the original and the watermarked *Bunny*, for $K = 200$.

However, if K is unknown, a case that appears often in practice, the means values $\bar{m}_k$ obtained by a poor estimate of K may not exhibit the same clustering behaviour. Fig. 6.13 illustrates this problem, demonstrating the need for an algorithm that will accurately estimate K.

The proposed algorithm is based on an exhaustive search through possible values of K and for each K we classify the mean values $\{\bar{m}_k : 1 \leq k \leq K\}$ into two clusters. We use a standard clustering algorithm fitting the data $\{\bar{m}_k : 1 \leq k \leq K\}$ with a mixture of two Gaussians $\mathcal{N}(\mu_{K,i}, \sigma^2_{K,i}), i = 1, 2$ with means $\mu_{K,i}$ and variances $\sigma^2_{k,i}$. If C and $\tilde{C}$ denote the two clusters, we measure the degree of separation between C



(a)

(b)

Figure 6.13: Scatter plot of the mean values m_k of the marked *Bunny* for (a) a 100 bin histogram and (b) a 240 bin histogram. In both cases the watermarked algorithm used $K = 200$ bins.

and $\tilde{C}$ as the Bhattacharyya distance D_K of the two Gaussians of the mixture

$$\frac{1}{8}(\mu_{K,2} - \mu_{K,1})^2 \left[\frac{\sigma_{K,1}^2 + \sigma_{K,2}^2}{2}\right]^{-1} + \frac{1}{2} \ln \frac{(\sigma_{K,1}^2 + \sigma_{K,2}^2)/2}{\sigma_{K,1}\sigma_{K,2}}. \tag{6.4.14}$$

Notice that the use of the Bhattacharyya to measure the distance between discrete or continuous probability distributions is quite common in practical applications [177]. Assuming that the more accurate estimates of K will most likely yield higher degrees of separation D_K between C and $\tilde{C}$ than the poor estimates of K (see Fig. 6.12 and 6.13), we estimate K as

$$K' = \arg\max_{K} \{D_K \, : \, K \in [K_{\min}, K_{\max}], K \in \mathbb{N}\} \tag{6.4.15}$$

where $K_{\min}$ and $K_{\max}$ define the range of K we would like to consider. Notice that the whole process is not very computationally demanding since for each value of K we only construct the bins $\{\tilde{\mathcal{B}}'_k \, : \, 1 \leq k \leq K\}$ and compute the mean values $\{\bar{m}_k \, : \, 1 \leq k \leq K\}$. Additionally, we do not need heavy computational cost, as the search space $[K_{\min}, K_{\max}]$ is small. In our experiments, we fix $K_{\min} = 1$ and $K_{\max} = 500$ since larger values of K create many empty bins that are unable to carry watermark messages and the watermarking algorithm itself fails.

After obtaining our estimate K' of K the next task is to decide whether the mesh has been watermarked. Recall that Cho et al.'s mean based watermarking

Table 6.2: Comparison of resistance against steganalysis. Column "Accuracy of K" shows the accuracy rate of the estimations of K. Column "Accuracy" shows the accuracy rate of watermark detection.

Methods	#Bits	#Marked models	Accuracy of K	Accuracy
Cho's	64	443	96.84%	94.58%
	100	386	96.63%	93.01%
Ours	64	377	0.65%	6.63%
	100	371	0	6.74%

algorithm embeds a watermark bit in such a way that $\bar{m}_k > 0.5$ or $\bar{m}_k < 0.5$ (see Fig. 6.12 (right)), depending on the value of the watermark bit. Consequently, if the model being considered is indeed watermarked, it should satisfy

$$\begin{cases} \min\limits_{c_i \in C}\{c_i\} > 0.5 > \max\limits_{\tilde{c}_i \in \tilde{C}}\{\tilde{c}_i\} \\ \text{abs}\left(\min\limits_{c_i \in C}\{c_i\} + \max\limits_{\tilde{c}_i \in \tilde{C}}\{\tilde{c}_i\} - 1\right) \leq \epsilon \end{cases} \tag{6.4.16}$$

assuming, without loss of generality, that the elements in C are larger than the element in $\tilde{C}$. We also require

$$\text{abs}\left(\frac{|C|}{K'} - \frac{|\tilde{C}|}{K'}\right) = \text{abs}\left(\frac{2\,|C|}{K'} - 1\right) \leq \epsilon' \tag{6.4.17}$$

The rationale behind the second requirement is the assumption that the watermark bits follow the uniform random distribution, hence, C and $\tilde{C}$ should contain almost equal numbers of elements, i.e., $|C| \approx |\tilde{C}|$. ϵ in Eq. 6.4.16 and ϵ' in Eq. 6.4.17 are two user-specified thresholds. In the experiments we used $\epsilon = 0.15$ and $\epsilon' = 0.03$.

In summary, the two main steps of the proposed steganalytic algorithm are: 1) find an estimate K' of K by maximizing the separation of the two clusters, using Eqs. 6.4.14 and 6.4.15; and 2) for the estimated K', check if Eqs. 6.4.16 and 6.4.17 are simultaneously satisfied or not.

To test the steganalytic algorithm, we constructed two large 3D databases: one is composed of clean models, most of which are from Princeton's University repository [178], while the other consists of marked models resulting from applying Cho et al.'s and our watermarking algorithms to the clean models in the first database. Notice that the use of a large bin number K might make some 3D meshes unable to carry the watermark, so we may end up with a different number of marked models for different K's (see Table 6.2).

Regarding Cho et al.'s watermarking method, Table 6.2 shows that the proposed steganalysis in most cases estimates correctly the bin number K and then detects the existence of watermark with high accuracy. Regarding our proposed watermarking method, the steganalysis fails, mainly because the estimates of the bin number K are very poor. As a result the detection accuracy is very low; more specifically, around 6%.

Even though the proposed steganalytic algorithm was specifically designed to target Cho et al.'s watermarking, we note that our attempt to also test it on our algorithm was a decision justified by the similarities between the two approaches, rather than an arbitrary one. In particular, we notice that when, for example, our algorithm moves elements from bin $\tilde{\mathcal{B}}'_k$ to bin $\tilde{\mathcal{B}}'_{k+1}$ to the right, the expected mean inside this pair of bins also shifts to the right. Thus, when we watermark a histogram with K bins, we would expect a 2-clustering in the means of bin pairs, that is, a clustering in the means of the histogram with $K/2$ bins.

The reason that such a 2-clustering goes largely undetected is that our algorithm would move elements from $\tilde{\mathcal{B}}'_k$ to $\tilde{\mathcal{B}}'_{k+1}$ and thus, shift the mean of the pair to the right, only if bin $\tilde{\mathcal{B}}'_k$ has more elements than bin $\tilde{\mathcal{B}}'_{k+1}$. That is, it will only shift to the right the means of bin pairs that are expected to be biased towards the left. In other words, the modulation of the normalized radiuses, instead of being a source of bias, tends to remove existing bias inside bin pairs and thus, it is more difficult to detect statistically.

6.5 Summary

We have presented a robust, blind watermarking algorithm which embeds a watermark on a 3D model by altering the histogram of the radial coordinates of the model's vertices. The algorithm has been demonstrated to be robust against common attacks to 3D models. Also, it is able to survive the operation that changes the mesh connectivity. The method is a modification of the watermarking algorithm proposed by Cho et al. [7], and compared to the original algorithm exhibits a better robustness distortion trade-off. Most characteristically, compared to Cho et al.'s watermarking, the proposed method exhibits stronger resistance against a specific steganalytic algorithm which we devised and also presented in this chapter.

Chapter 7

Laplacian Coordinates-based 3D Watermarking

As demonstrated by theoretical arguments or experiments, many existing 3D watermarking algorithms are able to withstand common attacks, such as geometric transformations, addition of noise and mesh smoothing. However, they have not taken into account the robustness of the watermark against mesh editing attacks that could be applied to 3D models to change the global shape of the mesh.

We believe that, in many cases, the assumption of a mesh editing attack is more realistic than the assumption of a noise addition, or a smoothing attack. Indeed, even if the attacker can use smoothing or noise addition to remove the watermark without degrading the visual quality of the model, still they would probably want to alter the global shape of the model through mesh editing in order to, either disguise the appropriation of the model, or to create a model fit for their purpose.

In this chapter, we are primarily concerned with the robustness against the above-mentioned type of mesh editing attacks. The watermarking method is based on modifying the Laplacian coordinate vectors (x, y, z). The basic idea is to embed the watermark into the histogram of the lengths of (x, y, z). The main reason of using the Laplacian coordinate lengths is due to their invariance property under mesh editing that deforms the global shape, as well as other common attacks, including translation, rotation, uniform scaling and vertex reordering. To encode a watermark bit, we change the heights of two adjacent histogram bins via transferring some elements from one bin to another. The watermark extraction is very simple and can be carried out blindly, with no reference to the original mesh.

The main contributions of this chapter can be summarized as follows:

- A new polygonal mesh watermarking method based on Laplacian coordinates.
- A demonstration that the algorithm is robust against editing operations altering the global shape of the mesh.

The main limitation is that the proposed watermarking scheme targets the resistance against mesh editing attacks only, so it offers weak robustness to other attacks, including mesh simplification and remeshing.

The rest of this chapter is organized as follows. The preliminaries and the basic ideas of the proposed watermarking are presented in Section 7.1. The details of watermark embedding and extraction are described in Section 7.2 and Section 7.3, respectively. The experimental results are presented and discussed in Section 7.4. Finally, we conclude in Section 7.5. Material in this chapter has been published in [9].

7.1 The Watermarking Algorithm

In this section we introduce the basic notation and terminology and then give an overview of the proposed algorithm.

7.1.1 Preliminaries

Let $\mathcal{M}$ be a polygonal mesh model with N vertices and let V and E denote the sets of vertices and edges of $\mathcal{M}$, respectively. Let v_i be the vertex indexed by i, its position described in Cartesian coordinates by $[x_i \; y_i \; z_i]$ (written as a row vector). The 1-ring neighbor of the vertex v_i is denoted by

$$\mathcal{N}(v_i) = \{v_j | (v_i, v_j) \in E, 1 \leq i, j \leq N\} . \tag{7.1.1}$$

For each vertex v_i, the vectors of the Laplacian coordinates $\mathbf{L}_i = [x'_i \; y'_i \; z'_i]$ are the rows of the matrix

$$\mathbf{L} = \mathbf{M} \times \begin{bmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_N & y_N & z_N \end{bmatrix} \tag{7.1.2}$$

where $\mathbf{M}$ is the *Kirchhoff* matrix [110] given by

$$\mathbf{M}_{i,j} = \begin{cases} |\mathcal{N}(v_i)| & \text{if } i = j \\ -1 & \text{if } v_j \in \mathcal{N}(i) \qquad 1 \leq i, j \leq N \\ 0 & \text{otherwise} \end{cases} \tag{7.1.3}$$

The use of a different discrete Laplacian, as for example the one used for mesh editing in [179], would lead to a watermarking algorithm of comparable performance.

The Cartesian coordinates of the vertices are obtained from the Laplacian coordinates by $\mathbf{M}^{-1} \times \mathbf{L}$. For numerical stability, as $\mathbf{M}$ may be not invertible, we update the $\mathbf{M}$ in Eq. 7.1.3 by

$$\mathbf{M}_{i,j} = \begin{cases} \mathbf{M}_{i,j} - \epsilon & \text{if } i = j \\ \mathbf{M}_{i,j} & \text{otherwise} \qquad 1 \leq i, j \leq N \end{cases} \tag{7.1.4}$$

where $\epsilon > 0$ is a small real number.

The proposed watermarking method embeds the watermark into the histogram of the lengths of the Laplacian coordinate vectors

$$d_i = \|\mathbf{L}_i\| = \sqrt{x_i'^2 + y_i'^2 + z_i'^2} \qquad 1 \leq i \leq N \tag{7.1.5}$$

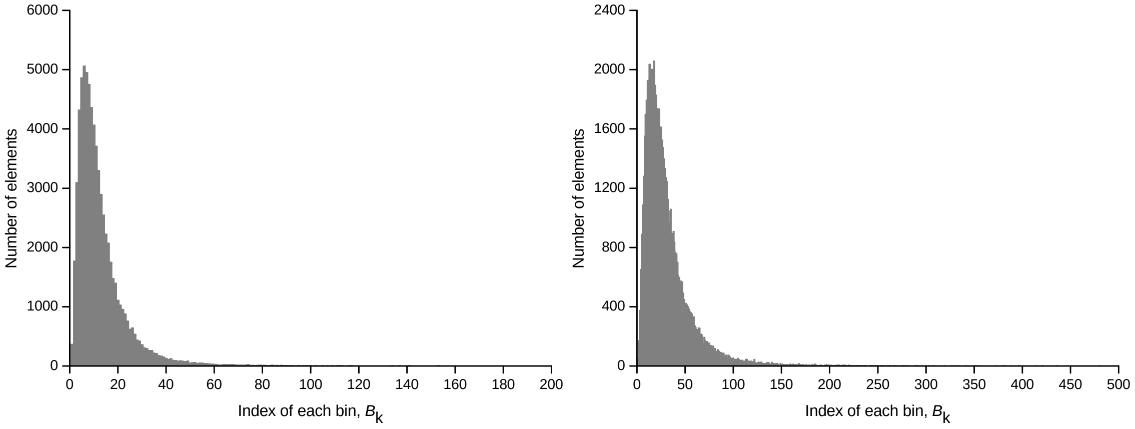
These lengths are then classified into K bins $\mathcal{B}_k$, according to

$$\mathcal{B}_k = \{d_i | d_{\min} + \delta(k-1) \leq d_i < d_{\min} + \delta k\} \tag{7.1.6}$$

with $1 \leq i \leq N$ and $1 \leq k \leq K$. The $d_{\min}$ and $d_{\max}$ are the minimum and maximum elements of $\mathbf{d} = \{d_1, d_2, \cdots, d_N\}$ and

$$\delta = \frac{d_{\max} - d_{\min}}{K} \tag{7.1.7}$$

is the size of the bin. We also assume that $d_{\max} \in \mathcal{B}_K$.



(a) (b)

Figure 7.1: Histograms of the lengths of the vectors of the Laplacian coordinates of the Rabbit model with (a) $K = 200$ and (b) $K = 500$ bins.

The histogram of $\mathbf{d}$ is produced by counting the number of elements in each bin $\mathcal{B}_k$. Two examples for the *Rabbit* model for the values of $K = 200$ and $K = 500$ are shown in Fig. 7.1. The figure indicates that many bins are empty.

7.1.2 Overview of The Algorithm

We assume that the watermark is a bitstring. Each bit ($w_j = -1$ or $+1$) is inserted into a pair of bins $(\mathcal{B}_{k_1}, \mathcal{B}_{k_2})$ with $k_1 \neq k_2$. However, in order to ensure exact watermark extraction, some pairs of bins will not be used. A pair of bins $(\mathcal{B}_{k_1}, \mathcal{B}_{k_2})$ is *valid* if it satisfies

$$f(\mathcal{B}_{k_1}, \mathcal{B}_{k_2}) = |\mathcal{B}_{k_1} \cup \mathcal{B}_{k_2}| = |\mathcal{B}_{k_1}| + |\mathcal{B}_{k_2}| \geq n_{\text{thr}} \tag{7.1.8}$$

with $k_1 \neq k_2$ and $|X|$ denoting the number of elements of X. Here, n_{thr} is an embedding threshold.

A pair of bins that is not valid is called *invalid*. The proposed method does not utilize the bins $\mathcal{B}_1$ and $\mathcal{B}_K$, and thus, $(\mathcal{B}_k, \mathcal{B}_1)$ and $(\mathcal{B}_k, \mathcal{B}_K)$ are always invalid. As it will become apparent later, this requirement ensures that the watermark can be extracted with no reference to the original model.

Assuming that $(\mathcal{B}_{k_1}, \mathcal{B}_{k_2})$ is valid, we insert one watermark bit w_j into this pair by, if necessary, moving some elements from $\mathcal{B}_{k_1}$ into $\mathcal{B}_{k_2}$ or vice-versa, such that

$$\begin{cases} |\hat{\mathcal{B}}_{k_1}| < |\hat{\mathcal{B}}_{k_2}| & \text{if } w_j = -1 \\ |\hat{\mathcal{B}}_{k_1}| \geq |\hat{\mathcal{B}}_{k_2}| & \text{if } w_j = +1 \end{cases} \tag{7.1.9}$$

where $\hat{\mathcal{B}}_{k_1}$ and $\hat{\mathcal{B}}_{k_2}$ are the k_1-th and k_2-th bins after the watermarking operation. To move an element d_i from one bin to another, we either enlarge or reduce its value so as to push it to the other bin. Notice that no element movement operation is applied when $|\mathcal{B}_{k_1}| < |\mathcal{B}_{k_2}|$ and $w_j = -1$, or $|\mathcal{B}_{k_1}| \geq |\mathcal{B}_{k_2}|$ and $w_j = +1$.

Finally, the watermark bit w_j is extracted from the pair of bins $(\hat{\mathcal{B}}_{k_1}, \hat{\mathcal{B}}_{k_2})$

$$w_j = \begin{cases} -1 & \text{if } |\hat{\mathcal{B}}_{k_1}| < |\hat{\mathcal{B}}_{k_2}| \\ +1 & \text{if } |\hat{\mathcal{B}}_{k_1}| \geq |\hat{\mathcal{B}}_{k_2}| \end{cases} \tag{7.1.10}$$

7.2 Watermark Embedding

In order to embed the watermark, we first alter the set of Laplacian lengths $\mathbf{d}$, computing a new set $\hat{\mathbf{d}}$

$$\hat{\mathbf{d}} = \{\hat{d}_1, \hat{d}_2, \cdots, \hat{d}_N\} \tag{7.2.11}$$

carrying the watermark. Then, through a minimization process, a set of Laplacian vectors $\hat{\mathbf{L}}$ with lengths $\hat{\mathbf{d}}$ is realized, and eventually the corresponding Cartesian coordinates are computed.

7.2.1 The Computation of $\hat{\mathbf{d}}$

As the bins $\mathcal{B}_1$ and $\mathcal{B}_K$ are excluded from the embedding process, and as we need one pair of bins to embed one message bit, if K is odd we need to exclude one more bin from the embedding process (here we chose $\mathcal{B}_{K-1}$). Thus, the maximum index of a bin used for watermarking is

$$\hat{K} = 2\left\lfloor \frac{K-2}{2} \right\rfloor + 1 \tag{7.2.12}$$

For any d_i inside the bins $\mathcal{B}_1$, $\mathcal{B}_K$ or $\mathcal{B}_{K-1}$ (when K is odd), we have $\hat{d}_i = d_i$, that is, no changes are made.

For the rest of the bins, in order to reduce the embedding distortion and hence improve the visual quality of the marked mesh, the watermark bits are inserted into pairs of adjacent bins $(\mathcal{B}_{k_1}, \mathcal{B}_{k_2})$ with $k_2 = k_1 + 1$. That gives a set of $(\hat{K} - 1)/2$ candidate bin pairs $(\mathcal{B}_2, \mathcal{B}_3), (\mathcal{B}_4, \mathcal{B}_5), \ldots, (\mathcal{B}_{\hat{K}-1}, \mathcal{B}_{\hat{K}})$.

For any such pair, if $f(\mathcal{B}_k, \mathcal{B}_{k+1}) < n_{\text{thr}}$, then the pair is invalid and we do not embed a watermark bit, that is, $\hat{d}_i = d_i$.

If $f(\mathcal{B}_k, \mathcal{B}_{k+1}) \geq n_{\text{thr}}$, then the pair is valid and the watermarking process depends on the relation between $\mathcal{B}_k$, $\mathcal{B}_{k+1}$ and the bit w_j. Without loss of generality we assume that $w_j = +1$, as the case $w_j = -1$ is completely analogous. We separate two cases:

Case 1: If $|\mathcal{B}_k| \geq |\mathcal{B}_{k+1}|$, then no alteration of the Laplacian lengths is required, and thus, $\hat{d}_i = d_i$.

Case 2: If $|\mathcal{B}_k| < |\mathcal{B}_{k+1}|$, then we have to transfer some elements from $\mathcal{B}_{k+1}$ to $\mathcal{B}_k$. We order the elements of $\mathcal{B}_{k+1}$ in ascending order

$$d_{i_1} \leq d_{i_2} \leq \cdots \leq d_{i_{|\mathcal{B}_{k+1}|}} \tag{7.2.13}$$

and update the first n elements in Eq. 7.2.13 by

$$\hat{d}_i = \begin{cases} \overline{d} & \text{if } d_i \in \{d_{i_1}, d_{i_2}, ..., d_{i_n}\} \\ d_i & \text{if } d_i \in (\mathcal{B}_{k+1} - \{d_{i_1}, d_{i_2}, ..., d_{i_n}\}) \cup \mathcal{B}_k \end{cases} \tag{7.2.14}$$

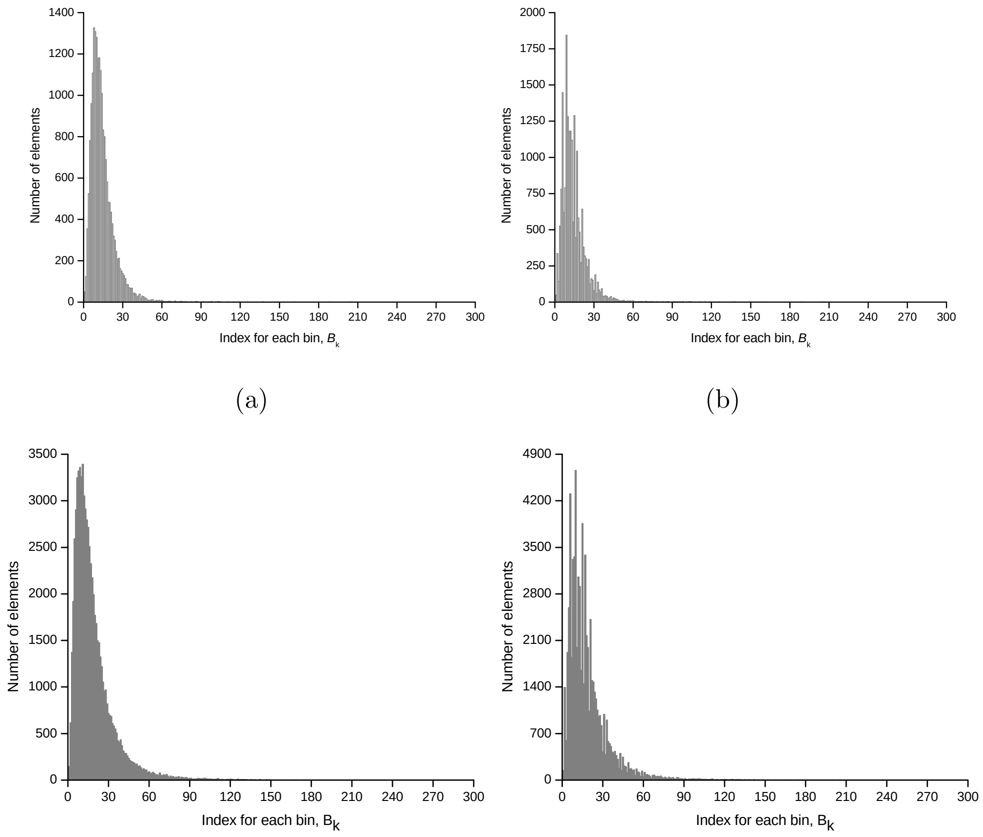
where

$$\overline{d} = \begin{cases} \dfrac{\sum_{d_i \in \mathcal{B}_k} d_i}{|\mathcal{B}_k|} & \text{if } |\mathcal{B}_k| > 0 \\ \\ \dfrac{2d_{\text{min}} + \delta(2k-1)}{2} & \text{if } |\mathcal{B}_k| = 0 \end{cases} \tag{7.2.15}$$

is the average value of elements in $\mathcal{B}_k$ if $\mathcal{B}_k$ is non-empty, or the average of the lower limits of $\mathcal{B}_k$ and $\mathcal{B}_{k+1}$, if $\mathcal{B}_k$ is empty. Notice that by transferring the smallest elements of $\mathcal{B}_{k+1}$ into $\mathcal{B}_k$, we keep the distortion of the mesh to a minimum.

The number n of points transferred from $\mathcal{B}_{k+1}$ to $\mathcal{B}_k$ is given by

$$|\mathcal{B}_{k+1}| - \left\lfloor \frac{f(\mathcal{B}_k, \mathcal{B}_{k+1}) - 1}{2} \right\rfloor + \left\lfloor \frac{f(\mathcal{B}_k, \mathcal{B}_{k+1}) - 1}{n_{\text{robust}}} \right\rfloor \tag{7.2.16}$$



(a)

(b)

(c)

(d)

Figure 7.2: Histogram changes as a result of watermarking the *Horse* and *Rabbit* models with $K = 300$ bins. (a) non-watermarked histogram of *Horse*, (b) watermarked histogram of *Horse*, (c) non-watermarked histogram of *Rabbit* and (d) watermarked histogram of *Rabbit*.

where the parameter n_{robust} controls the tradeoff between robustness of the watermark and distortion of the mesh. If the number of points to be transferred, as computed by Eq. 7.2.16, exceeds the number of points in $\mathcal{B}_{k+1}$, then we just transfer all the contents of $\mathcal{B}_{k+1}$ into $\mathcal{B}_k$.

7.2.2 Distortion Minimization and Watermarked Mesh Generation

Next we compute a set of Laplacian coordinates $\hat{\mathbf{L}}$ realizing the computed set of lengths $\hat{\mathbf{d}}$. This is an undetermined problem and we solve it by minimizing the distance between the Laplacian coordinates before and after watermarking. For computational efficiency we minimize that distance at each vertex separately, that is, we minimize

$$\|\hat{\mathbf{L}}_i - \mathbf{L}_i\|^2 = (\hat{x}'_i - x'_i)^2 + (\hat{y}'_i - y'_i)^2 + (\hat{z}'_i - z'_i)^2 \tag{7.2.17}$$

subject to

$$\hat{x}'^2_i + \hat{y}'^2_i + \hat{z}'^2_i = \hat{d}^2_i \tag{7.2.18}$$

From Eq. 7.2.17 and Eq. 7.2.18, we observe that this minimization problem is equivalent to finding a point $(\hat{x}'_i, \hat{y}'_i, \hat{z}'_i)$ on a sphere $\mathcal{S}$ of radius $\hat{d}_i$ centered at the origin that is closest to the given point (x'_i, y'_i, z'_i). This means that $(\hat{x}'_i, \hat{y}'_i, \hat{z}'_i)$ is the projection of (x'_i, y'_i, z'_i) on $\mathcal{S}$. As $\mathcal{S}$ is centered at the origin, the projection of (x'_i, y'_i, z'_i) on it is given by

$$\begin{aligned} \hat{x}'_i &= \frac{x'_i \hat{d}_i}{\sqrt{x'^2_i + y'^2_i + z'^2_i}} \\ \hat{y}'_i &= \frac{y'_i \hat{d}_i}{\sqrt{x'^2_i + y'^2_i + z'^2_i}} \\ \hat{z}'_i &= \frac{z'_i \hat{d}_i}{\sqrt{x'^2_i + y'^2_i + z'^2_i}} \end{aligned} \tag{7.2.19}$$

Finally, the Cartesian coordinates of the watermarked model $\hat{\mathcal{M}}$ are computed from its Laplacian coordinates by $\mathbf{M}^{-1} \times \hat{\mathbf{L}}$.

Fig. 7.2 shows the alteration of the histograms of the *Horse* and *Rabbit* models with $K = 300$ bins. A comparison between the non-watermarked and the watermarked histograms indicates that their global shapes are similar.

7.3 Watermark Extraction

Given a watermarked polygonal mesh model $\hat{\mathcal{M}}$ in Cartesian coordinates, the watermark extraction is very simple and can be carried out blindly, with no reference to the original mesh $\mathcal{M}$.

First, we obtain the 1-ring neighbors of each marked vertex $\hat{v}_i$ and construct the weighted Laplacian matrix $\mathbf{M}$ using Eq. 7.1.3 and Eq. 7.1.4. After calculating the watermarked Laplacian coordinates, we compute the lengths $\hat{\mathbf{d}}$ of the coordinate vectors.

Then, we classify the elements in $\hat{\mathbf{d}}$ into K bins $\hat{\mathcal{B}}_k, 1 \leq k \leq K$, using Eq. 7.1.6 and Eq. 7.1.7. Since the first and the last bins $\mathcal{B}_1$ and $\mathcal{B}_K$ have not been altered by the watermarking process, the minimum $\hat{d}_{\min}$ and maximum $\hat{d}_{\max}$ of $\hat{\mathbf{d}}$ (after watermarking) are equal to the minimum $d_{\min}$ and maximum $d_{\max}$ of $\mathbf{d}$ (before watermarking). That is, given the total number of bins K, the step size δ used in the watermark extraction process is the same as the step size used in the watermark embedding process. This is an efficient solution to the non-synchronization problem. However, $\hat{d}_{\min}$ and maximum $\hat{d}_{\max}$ may change significantly when the watermarked mesh $\hat{\mathcal{M}}$ has suffered from malicious attacks. If this is the case, we form the bins using only those $\hat{d}_i$ that satisfy $d_{\min} \leq \hat{d}_i \leq d_{\max}$ and Eq. 7.1.7.

In the next step, we disregard the bins $\hat{\mathcal{B}}_1$ and $\hat{\mathcal{B}}_k$ ($k > \hat{K}$), which do not carry watermark, and form the pairs $(\hat{\mathcal{B}}_2, \hat{\mathcal{B}}_3), (\hat{\mathcal{B}}_4, \hat{\mathcal{B}}_5), \cdots, (\hat{\mathcal{B}}_{\hat{K}-1}, \hat{\mathcal{B}}_{\hat{K}})$. From the embedding process, we know that only pairs $(\mathcal{B}_k, \mathcal{B}_{k+1})$ satisfying $f(\mathcal{B}_k, \mathcal{B}_{k+1}) \geq n_{\text{thr}}$ have been used as watermark carriers. Based on the equation

$$f(\mathcal{B}_k, \mathcal{B}_{k+1}) = f(\hat{\mathcal{B}}_k, \hat{\mathcal{B}}_{k+1}) \tag{7.3.20}$$

which is a consequence of the embedding strategy that swaps vertices inside pairs of bins only, we can find the pairs that carry watermark by checking if

$$f(\hat{\mathcal{B}}_k, \hat{\mathcal{B}}_{k+1}) \geq n_{\text{thr}} \tag{7.3.21}$$

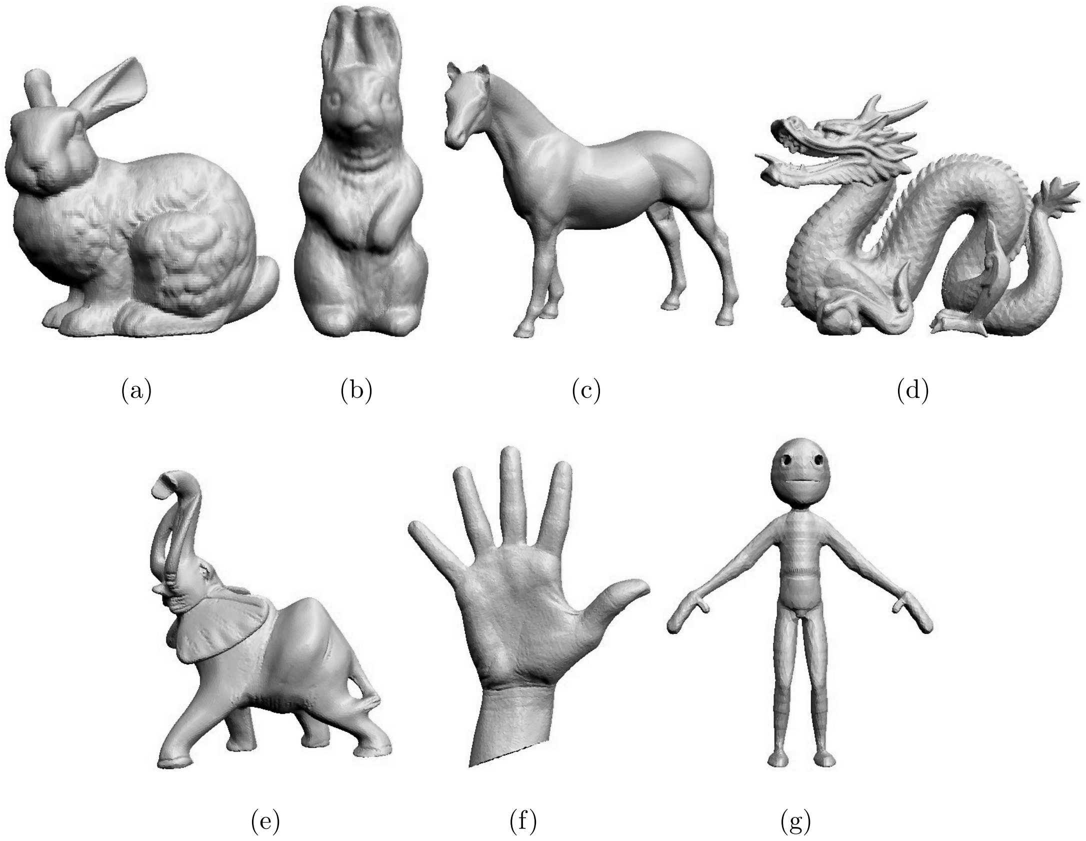
That means that we do not need the knowledge of the original mesh $\mathcal{M}$ to compute the pairs of bins that carry watermark.

Finally, each embedded watermark bit w_j is sequentially extracted from each pair $(\hat{\mathcal{B}}_k, \hat{\mathcal{B}}_{k+1})$ by

$$w_j = \begin{cases} +1 & \text{if } |\hat{\mathcal{B}}_k| \geq |\hat{\mathcal{B}}_{k+1}| \\ -1 & \text{if } |\hat{\mathcal{B}}_k| < |\hat{\mathcal{B}}_{k+1}| \end{cases}$$

7.4 Experimental Results

The proposed method has been implemented using Matlab based on Gabriel Peyré's code [180]. The 3D mesh models used in the experiments, namely, the *Bunny*, *Rabbit*,



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

(e) (f) (g)

Figure 7.3: Original mesh models used in the experiments. (a) *Bunny*, (b) *Rabbit*, (c) *Horse*, (d) *Dragon*, (e) *Elephant*, (f) *Hand*, and (g) *Hank*.

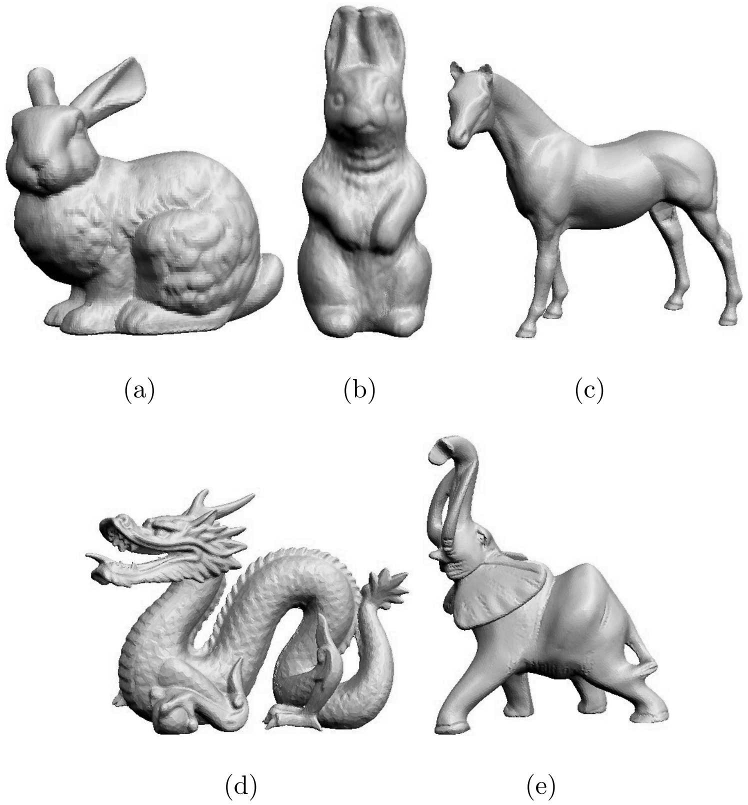
Horse, *Dragon*, *Elephant*, *Hand* and *Hank*, are shown in Fig. 7.3. The watermark bits are produced randomly with uniform distribution, using the Matlab function **randint**. The parameter ϵ in Eq. 7.1.4 that ensures the invertibility of the Laplacian matrix $\mathbf{M}$ is fixed at $\epsilon = 0.00001$.

The algorithmic complexity of the proposed watermarking algorithm is linear to the number of mesh vertices. In our Matlab implementation, which is not optimized for time efficiency, it takes few minutes (for instance, approximately one minute for the *Bunny* model) to watermark the test meshes on a PC running on an Intel Core 2 Duo T6570 2.1 GHz processor with 2 GB memory.

The parameters used in the experiment are reported in Table 7.1. Notice that the parameters n_{thr} and K need to be passed to the watermark extraction algorithm. In a real installation framework, we can avoid the necessity of passing n_{thr} and K to the extraction algorithm by always fixing n_{thr} at a constant and using a K proportional

Table 7.1: Parameter setting and experimental results. $\mathcal{C}$ is the embedding capacity of the model in bits. In each test, the number of actually changed bins is $2\mathcal{C}$.

Model	N	K	n_{thr}	n_{robust}	$\mathcal{C}$
Bunny	34835	498	46	5	21
Rabbit	70658	1010	46	5	102
Horse	19851	284	46	5	22
Dragon	50000	715	46	5	103
Elephant	24955	498	46	5	25
Hand	17117	245	46	5	52
Hank	15488	150	70	5	30



(a) (b) (c)

(d) (e)

Figure 7.4: Watermarked mesh models. (a) Bunny, (b) Rabbit, (c) Horse, (d) Dragon, and (e) Elephant.

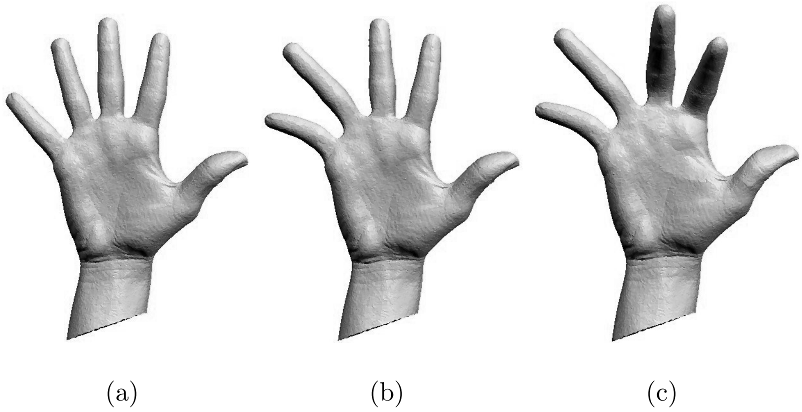


Figure 7.5: Illustration of the watermarked Hand and its deformed versions. (a) watermarked Hand, (b) deformation of (a), and (c) deformation of (a).

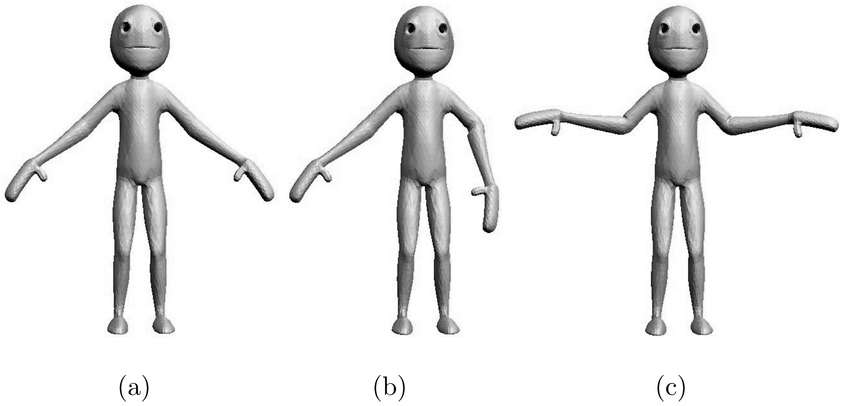


Figure 7.6: Illustration of the watermarked Hank and its deformed versions. (a) watermarked Hank, (b) deformation of (a), and (c) deformation of (a).

to N, that is, $K = \lceil N/N_c \rceil$, where N_c is a constant. In most experiments, $n_{\text{thr}} = 46$ and the parameter K was computed this way with $N_c = 70$.

7.4.1 Evaluation of the Visual Degradation

A first desirable characteristic of a watermarking method is the transparency of the watermark. That is, we expect the embedded watermarks to be imperceptible. Our first experiment evaluates the visual impact of the mesh alterations caused by watermarking.

Fig. 7.4, Fig. 7.5 and Fig. 7.6 show the test models after watermarking. We found

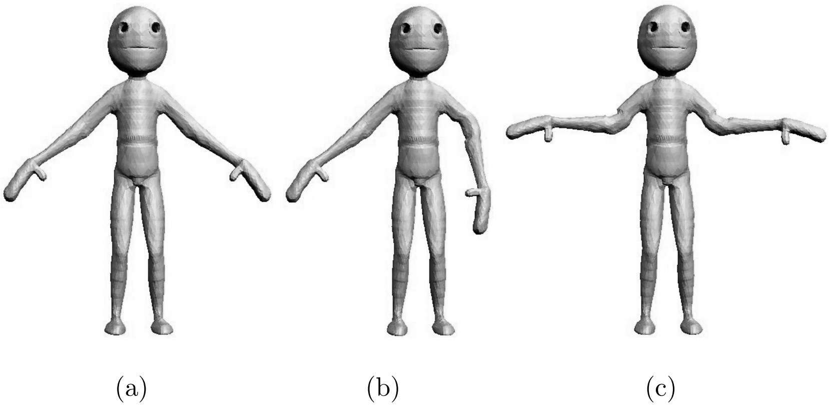


Figure 7.7: Illustration of Cho et al.'s [7] watermarked Hank and its deformed versions. (a) watermarked Hank, (b) deformation of (a), and (c) deformation of (a).

that it is fairly difficult to observe any undesirable artifacts on the watermarked models, even upon close inspection. In addition, the watermarked models shown in Fig. 7.4, Fig. 7.5 and Fig. 7.6 and the corresponding originals shown in Fig. 7.3 are visually indistinguishable. We conclude that the proposed watermarking method is capable of preserving the visual quality and the global shape of the cover models.

7.4.2 Evaluation of Robustness

A watermarking method should be capable of surviving unintentional or malicious attacks, preventing the removal of the embedded watermark by an adversary. To evaluate the robustness of the proposed method, we apply the attack to the watermarked model and attempt to extract the watermark from the attacked model. The standard measures of robustness are the normalized correlation (NC) and the correct detection rate (CDR), which are obtained by a comparison between the original watermark string and the extracted one. In this chapter, two types of attacks were conducted: distortion-free geometric transformations and mesh editing operations. Note that we do not measure the robustness against some other common attacks, such as noise addition and cropping, as the visual alterations resulted from such operations are usually at an unacceptable level.

As expected, the watermark is perfectly robust against vertex reordering and geometric transformations, including translation, rotation and uniform scaling. This is

Table 7.2: Evaluation of the robustness of the proposed algorithm against mesh editing operations.

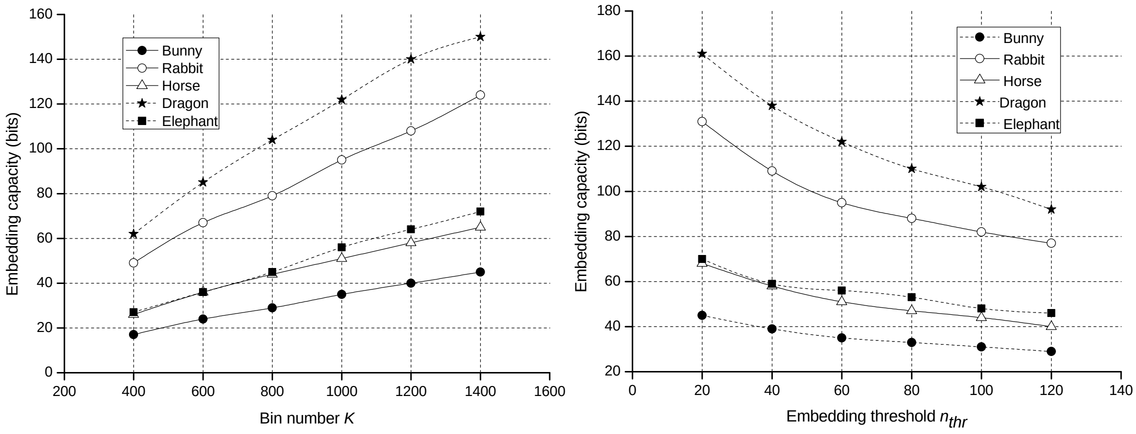
Model	NC	CDR
Fig. 7.5 (b)	0.93	0.96
Fig. 7.5 (c)	0.73	0.86

Table 7.3: Evaluation of the robustness of the proposed algorithm against mesh noise insertion and Laplacian smoothing for Bunny.

Attack	NC	CDR
Noise Insertion	0.85	0.93
Laplacian Smoothing	0.26	0.64

because the watermark carrier, that is, the histogram of the lengths of the Laplacian coordinate vectors, is invariant under these operations, except uniform scaling. Although uniform scaling may deform the histogram, the watermark survives because both embedding and extraction are based on relationships between histogram bins that are invariant under uniform scaling.

Regarding vicious attacks, we only consider mesh editing operations that are likely to be applied to the 3D models in practice. We carried out such operations using the popular *Character Rigging* tool of the well-known 3D editing software *Blender 2.49* [176]. Some instances of the mesh models undergoing deformation are shown in Fig. 7.5 and Fig. 7.6. In both examples, the shapes of the deformed models are meaningful and can still be used in practical applications. Table 7.2 lists the robustness results under mesh editing for the two models. We notice that we obtain high values of NC and CDR, even for severely edited models. The reason is that even though the mesh editing operations modify the global shape of the models, the watermark carrier, i.e., the histogram of the lengths of the Laplacian coordinates vectors, is almost unchanged. We conclude that we are able to correctly extract most of the embedded watermark bits, and hence, the proposed watermarking method



(a)

(b)

Figure 7.8: Embedding capacity versus (a) number of bins K, with fixed embedding threshold $n_{\text{thr}} = 60$ and (b) embedding threshold n_{thr}, with fixed $K = 1000$. In both cases, $n_{\text{robust}} = 5$.

offers satisfactory resistance against editing attacks.

Notice that the proposed watermarking is specially designed to survive mesh editing attacks that deform the global shape of a 3D model and thus that it may offer relatively weaker robustness to other types of operations, such as noise insertion and Laplacian smoothing. This is because the lengths of Laplacian vectors could change significantly when those operations are applied. As Table 7.3 shows, we obtain lower values of NC and CDR from the attacked *Bunny*, which was generated by adding noise and applying Laplacian smoothing to the marked *Bunny* as shown in Fig. 7.4 (a).

7.4.3 Embedding Capacity

The data in the second and fourth columns of Table 7.1 imply that, for each of the test models, the embedding capacity does not attain its theoretical maximum $\lfloor (K-2)/2 \rfloor$, indicating that the method cannot embed a watermark bit into every bin pair. This happens because many bins are empty, and because some bin pairs do not meet the embedding condition in Eq. 7.1.8, see also Fig. 7.1 and Fig. 7.2.

Regarding the real embedding capacity of the method, the two crucial parameters

Table 7.4: Comparison with previously proposed methods. Here, $|v|$ and $|w|$ represent the numbers of the mesh vertices and watermark bits, respectively; the symbols $\times$ and $\surd$ indicate that the method can weakly survive and can survive (similarity transformation, reordering or mesh editing) attacks, respectively; n_{layers} is an embedding parameter. The previous approaches used for comparison include those by Cayre et al. [5], Wang et al. [11], Chao et al. [6], Zafeiriou et al. [12], Cho et al. [7] and Ohbuchi et al. [13].

Method	Domain	Capacity	Extraction	Similarity	Reordering	Editing
Cayre	spatial	$\sim\|v\|$	B	$\surd$	$\times$	$\times$
Wang	spatial	$3\|v\|$	B	$\times$	$\times$	$\times$
Chao	spatial	$3n_{layers}\|v\|$	B	$\times$	$\times$	$\times$
Zafeirio	spatial	$\|w\|$	B	$\surd$	$\surd$	$\times$
Cho	spatial	$\|w\|$	B	$\surd$	$\surd$	$\times$
Ohbuchi	frequency	$\|w\|$	NB	$\surd$	$\surd$	$\times$
Ours	frequency	$\|w\|$	B	$\surd$	$\surd$	$\surd$

are the number of bins K and the embedding threshold n_{thr}. Fig. 7.8 (a) shows the embedding capacity for variable K and fixed $n_{\text{thr}} = 60$. As expected, the embedding capacity increases monotonically with K, as a larger K means that more bin pairs are available for watermarking. Fig. 7.8 (b) shows the embedding capacity for variable n_{thr} and fixed $K = 1000$. As expected, the embedding capacity decreases monotonically with n_{thr}, as a greater n_{thr} means that more bin pairs become invalid.

7.4.4 Comparison and Discussion

We compared the proposed method with existing watermarking and steganographic algorithms. The results of the evaluation are summarized in Table 7.4. We notice

Table 7.5: Performance comparison between ours and Cho et al.'s [7] methods.

Model	Method	NC	CDR
Fig. 7.6 (b)	Proposed	0.87	0.93
Fig. 7.7 (b)	Cho's	0.49	0.73
Fig. 7.6 (c)	Proposed	0.93	0.96
Fig. 7.7 (c)	Cho's	0.19	0.60

that steganographic methods have higher embedding capacity but weaker robustness compared to the watermarking ones. In fact, steganographic methods are quite sensitive to mesh alterations and even a slight processing could cause the complete removal of the embedded message. As such, they are not appropriate for applications such as digital content protection and authentication. By contrast, watermarking methods show higher levels of robustness, at the expense of their embedding capacity.

We claim that the proposed method outperforms other schemes in terms of robustness under common shape-changing editing operations. The reason is that the watermark primitives used by previous schemes, for example, the vertex norms used by Cho et al. [7], are sensitive to changes of the global shape. In contrast, the histogram of the lengths of the Laplacian vectors is less sensitive to such operations.

Table 7.5 summarizes the experimental comparison between the proposed method and Cho et al.'s method [7], regarding robustness to mesh editing. We implemented Cho et al.'s watermark algorithm that alters the distribution of vertex norms, defined as the distances between the vertices and the center of the mesh model. In the experiment, the same bit string was embedded to the *Hank* model using Cho et al.'s and our method, respectively. For Cho et al.'s algorithm, we set the watermark strength factor $\alpha = 0.04$ and the bin number to 30, while for our algorithm we used the parameter values listed in Table 7.1. Then, using *Blender*, we deformed Cho's and ours watermarked models, trying to obtain edited models with the same appearance, see Fig. 7.6 and Fig. 7.7. Table 7.5 shows that our approach achieves higher NC and CDR values, and hence, it is more robust.

The Ohbuchi et al. [13] method is weakly resistant against mesh editing attacks because the watermark is embedded in the low frequencies of the spectrum of the Laplacian. However, variants of their method using medium or high frequencies as watermark carriers could be more resistant to mesh editing attacks. Moreover, their method outperforms ours in that it is robust against a wider range of attacks, including mesh simplification and remeshing. On the other hand, their method is a non-blind scheme and requires the original mesh for watermark extraction. That makes it unsuitable for a range of practical applications where such information can not be provided to the extraction algorithm.

7.5 Summary

We have proposed a watermarking algorithm for polygonal meshes based on the histogram of the lengths of Laplacian coordinate vectors. The watermark extraction is carried out blindly without reference to the original model. The proposed method is robust against translation, rotation, uniform scaling and vertex reordering as a result of using primitives that are invariant under these operations to carry the watermark. Most importantly, the proposed method is robust under common mesh editing operations that change the global shape of the mesh. However, it may not be able to survive mesh editing attack that changes the mesh connectivity since such an attack could modify the Laplacian coordinates significantly and thus the histogram of the lengths of Laplacian vectors greatly. We believe that mesh editing is the logical choice of a malicious attacker aiming at the unauthorized use of copyrighted material, and thus, the research on watermarking methods that are robust against such attacks can have immediate practical implications.

The main limitation of the proposed watermarking technique is its weak resistance against other attacks, such as mesh simplification, remeshing and topological changes. Such operations may change the Laplacian coordinates significantly, and consequently alter the watermark carrier, i.e., the histogram of the lengths of the Laplacian coordinate vectors, to the extent that the embedded watermark is destroyed.

Chapter 8

Discriminative Features Extraction for 3D Steganalysis

In this chapter, we propose a steganalytic method for triangle meshes, adopting a framework which has been successfully developed for image steganalysis. The main idea is that even though the presence of the watermark is often imperceptible to the human eye, it may nevertheless disturb the natural statistics of the 3D signal and thus become detectable.

We implement this idea by computing for each mesh a characteristic feature vector capturing geometric information extracted from its Cartesian and Laplacian coordinates, its dihedral angles and face normals. For the extracted feature vector to have sufficient for our purposes discriminative power, we do not compute it directly on each mesh, but on the difference between the mesh and a filtered copy of it called the *reference* mesh. This technique is known in the literature of image steganalysis as *calibration*. Here, the reference mesh is produced by applying one iteration of Laplacian smoothing.

In [144], where calibration was introduced, the reference image was deemed to serve as an approximation of the original image before the embedding of the watermark. However, as [181] points out, this approximation does not need to hold for the calibration to work. Instead, what we expect from the calibration is to erase a part of the changes introduced by watermarking, without altering the cover significantly. Then, as a result, the difference between a mesh and its reference will be distinctively larger for watermarked than for clean models. From this point of view, what our experiments demonstrate is that when one of the five test watermarking/steganographic algorithms is used, the difference between a mesh and its smoothed version is distinctively larger for marked meshes, when a certain well-chosen statistical measure of that difference is used.

After the extraction of the feature vector, a supervised learning algorithm based on Quadratic Discriminate Analysis is applied to a training set of feature vectors from unmarked meshes and meshes marked by a given watermarking/steganographic algorithm, yielding a steganalytic classifier for that particular method. We also obtained a universal steganalyzer by training the classifier on a set of meshes marked by all five watermarking/steganographic test algorithms.

The main contribution of this chapter is a steganalytic technique for triangle meshes, which, to the best of our knowledge, is the first 3D steganalytic method in the literature. Our experiments show that the method can be successfully used as a benchmark for the anti-steganalysis performance of existing and future 3D steganographic/watermarking algorithms.

Regarding the limitations of the proposed technique, we notice given that there are no other steganalytic methods to compare our results against, it is difficult to judge the significance of the obtained accuracy rates. However, we may assume that as it is the case with image steganalysis too, one may achieve higher accuracy rates by devising algorithms targeting specific steganographic/watermarking methods.

A second limitation is that the proposed steganalytic method only targets algorithms that hide information into the mesh geometry. Even though such algorithms are the mainstream of 3D steganography/watermarking, it should be noted that there are also algorithms hiding information into the mesh connectivity, or into the data redundancy of polygonal list files encoding triangle meshes. In its current form, our method cannot detect watermarks inserted by such algorithms.

The rest of this chapter is organized as follows. We describe in detail the method concerning how to compute feature vectors from 3D models in Section 8.1. The steganalysis results are presented and discussed in Section 8.2. Section 8.3 concludes this chapter. Material in this chapter has been submitted to *ACM Transactions on*

Multimedia Computing, Communications and Applications for consideration.

8.1 Triangle Mesh Steganalysis

Given a target watermarking/steganographic algorithm, the steganalytic algorithm extracts N-dimensional feature vectors $\mathbf{F}_i$ from a training set of clean and watermarked models

$$\mathbf{F}_i = (f_{i,1}, f_{i,2}, \cdots, f_{i,N}) \tag{8.1.1}$$

where i is the index of the model in the training set and $N = 208$ in the chapter. These feature vectors are the input of a supervised learning algorithm, which produces a classifier associated with the target watermarking/steganographic method.

8.1.1 Normalization

In a pre-processing step, the model is normalized by a coordinate system change before feature extraction. We apply Principal Component Analysis (PCA) to the vertex coordinates, and align the xyz axes of the coordinate system with the three principal directions $\mathbf{q}_1$, $\mathbf{q}_2$ and $\mathbf{q}_3$, respectively, assuming, without loss of generality that $\lambda_1 \geq \lambda_2 \geq \lambda_3$ for the corresponding eigenvalues. After this coordinate system transformation, we uniformly scale the model into the unit cube centered at $(0.5, 0.5, 0.5)$.

The normalization ensures that the feature vector $\mathbf{F}_i$ of each model is invariant to affine transformations. Also, normalization restricts the values of each component $f_{i,j}$ of $\mathbf{F}_i$ to a limited range and thus, prevents the large feature values dominating the smaller values. Notice that because of using PCA, we are not able to specify the orientation of the axes of the new coordinate system. That means that all extracted features should be invariant under an orientation change of any of the axes.

8.1.2 Calibration

The reference mesh $\mathbf{M}'$ we use for calibration is produced by applying one iteration of Laplacian smoothing on the original mesh $\mathbf{M}$. We use a standard Laplacian operator corresponding to the *Kirchhoff* matrix [110] with entries

$$\mathbf{R}_{i,j} = \begin{cases} \mathrm{val}(v_i) & \text{if } i = j \\ -1 & \text{if } v_j \in \mathcal{N}(i) \quad 1 \leq i, j \leq V \\ 0 & \text{otherwise} \end{cases} \tag{8.1.2}$$

where v_i is the vertex indexed by i, $\mathrm{val}(v_i)$ and $\mathcal{N}(i)$ denote the valence and the 1-ring neighborhood of v_i, and V is the number of mesh vertices.

We have found that this simple calibration process works well against all watermarking algorithms we have tested the proposed method on. Moreover, its simplicity gives a reasonable expectation that the experimental results will generalize well against other watermarking algorithms that have been proposed, or will be proposed in the future.

8.1.3 Feature Extraction

The computation of the feature vector $\mathbf{F}_i$ is a two-step process: *extracting features* and *computing components of* $\mathbf{F}_i$. More specifically, we first compare the original and the reference models and compute vectors with components corresponding to mesh vertices, edges or faces. Next, these vectors are processed to produce the components of the feature vector $\mathbf{F}_i$.

Regarding the vertex vectors, we compute the absolute values of the differences of the x, y and z coordinates of $\mathbf{M}$ and $\mathbf{M}'$, as well as the length of the vector of Cartesian coordinate differences. Essentially, these components are the absolute values of the Laplacian coordinates and the length of the vector of Laplacian coordinates of $\mathbf{M}$. Next, we obtain four more vectors by computing the same absolute differences but on the Laplacian rather than the Cartesian coordinates of $\mathbf{M}$ and $\mathbf{M}'$. The computations are done separately on vertices with valence less than, equal, or greater than six, excluding all boundary vertices. That is, we treat separately vertices that are topologically convex, planar, or concave in the combinatorial Gaussian curvature sense [182]. The total number of vectors obtained from the mesh vertices is 24.

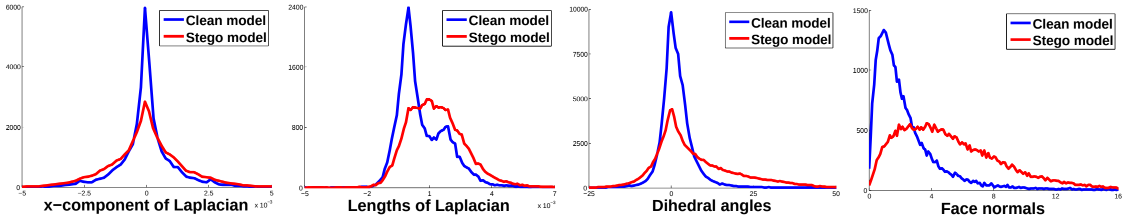


Figure 8.1: Comparison of the histograms before and after embedding with the variance-based watermarking in [7] on the Stanford Bunny. The y axis shows the frequency. The histograms were constructed from the x component of Laplacian coordinates **(Left)**, the differences of the lengths of Laplacian coordinate vectors **(Second from Left)**, the differences of the dihedral angles **(Second from Right)** and the differences of the angles between face normals **(Right)**.

Regarding the edges of the mesh, we compute the vector of the absolute values of the differences of the dihedral angles between $\mathbf{M}$ and $\mathbf{M}'$. Finally, regarding mesh faces, we compute the vector of the angles between the normals of $\mathbf{M}$ and $\mathbf{M}'$, obtaining a total of 26 vectors from vertices, edges, and faces.

Fig. 8.1 shows the changes in the histograms of the extracted features induced by the variance-based watermarking in [7] when applied on the Stanford *Bunny*. The observed large differences in the shape of the histogram between the clean and the marked *Bunny* model illustrate in a nutshell why the proposed method works.

8.1.4 The Feature Vector $\mathbf{F}_i$

From each of the 26 vectors computed in the previous section, we compute eight components of the feature vector $\mathbf{F}_i$, creating a vector of dimension $N = 208$. Let $\mathbf{f}$ be one of these 26 vectors. The first four components of $\mathbf{F}_i$ corresponding to $\mathbf{f}$ are the mean, variance, skewness and kurtosis of $\log(|\mathbf{f}|)$. Notice that these four statistical characteristics are commonly used in image staganalysis [30] as vector shape descriptors. The purpose of the log transform is the reduction of the range of the values and also an increase in the weight of small positive values.

For the other four components, we first build the histogram of $\mathbf{f}$ with

$$H = \left\lceil \frac{\max(\mathbf{f}) - \min(\mathbf{f})}{h} \right\rceil \tag{8.1.3}$$

bins, where the size of the bin is given by the Freedman–Diaconis rule [183]

$$h = 2\,\frac{\mathrm{IQR}(\mathbf{f})}{n^{1/3}}, \tag{8.1.4}$$

where $\mathrm{IQR}(\cdot)$ denotes interquartile range and n is the number of components in $\mathbf{f}$. By counting the number of elements in each bin, we obtain the frequency vector $\mathbf{n} = (n_1, n_2, \cdots, n_H)$ and its difference vector $\mathbf{n}' = (n_2 - n_1, n_3 - n_2, \cdots, n_H - n_{H-1})$. Then, the four remaining components of $\mathbf{F}_i$ obtained from $\mathbf{f}$ are the mean, variance, skewness and kurtosis of $\log(|\mathbf{n}'|)$.

We notice that the dimension of the vector $\mathbf{F}_i$ is relatively low, compared for example with the PEV-274 [184], which has dimension 274 and is used for image steganalysis. Indeed, given that images are structurally simpler than triangle meshes, one would expect that a good discriminative feature vector for meshes would have a higher dimension. Nevertheless, the proposed feature vector works well in practice.

8.1.5 The Classifiers

We train the classifiers using the feature vectors extracted from a training set of 3D models with and without watermarks. Before training the classifiers, the features are scaled into comparable dynamic ranges. More specifically, for any component f of the feature vector, we find its minimum $f_{\min}$ and maximum $f_{\max}$ values on the set of all training models. For any training or test model, we scale the feature component f by

$$\bar{f} = \frac{f - f_{\min}}{f_{\max} - f_{\min}} \tag{8.1.5}$$

Notice that $\bar{f} \in [0, 1]$ for all training models, while it is expected that $\bar{f}$ will also fall in [0, 1] for most test models. This scaling process prevents the features with large numerical ranges from dominating those with small numerical ranges and thus greatly improve classification accuracy [23].

Finally, the classifier is obtained via quadratic discrimination that fits multivariate normal densities with covariance estimates stratified by group [185]. Notice that the simpler Fisher Linear Discriminant has been successfully employed in prior steganalysis work [30]; however, we have empirically found that in our case it is slightly outperformed by the quadratic discrimination. We have also tested Adaboost classifiers based on Linear and Quadratic Discriminants, but there was no improvement on the results.

When the classifiers for each target embedding method have been computed, the steganalysis process is straightforward. Given a test 3D model, we just compute its feature data, apply the classifier for a particular steganographic method to the feature data and assign it to one of the two categories: *unmarked* by that method or *marked* by that method.

8.2 Experimental Results

In this section, we experimentally validate the proposed steganalytic algorithm against five well-known embedding techniques. They include Least Significant Bit modification, the high capacity steganographic scheme based on Principal Axis projection [6], two watermarking methods proposed in [7] which modify either the mean or the variance of the vertex norms inside the bins of a histogram, and the watermarking method in [9].

Regarding LSB modification, for each vertex we compute the number of modifiable bits using [10], making the expectation of the normal distortion to be lower than a threshold ϵ. Regarding the parameter setting of Chao's steganography [6], we use $n_{\text{layers}} = 10$ for the number of layers and $n_{\text{intervals}} = 10000$ for the number of intervals. Regarding the mean and variance-based watermarking methods in [7], the incremental step size is fixed at $\Delta k = 0.001$ and the number of bins at 64, while we vary the value of the strength factor α for different models. In particular, α is a random number within the ranges $[0.02, 0.06]$ and $[0.12, 0.20]$ for mean-based and variance-based watermarking, respectively. Notice that the purpose of varying α is to simulate the situation of incurring different amounts of embedding distortion to different 3D models. Finally, for [9], the embedding threshold is set at $n_{\text{thr}} = 46$, the robustness threshold at $n_{\text{robust}} = 5$, while the number of bins is equal to $\lceil V/70 \rceil$,

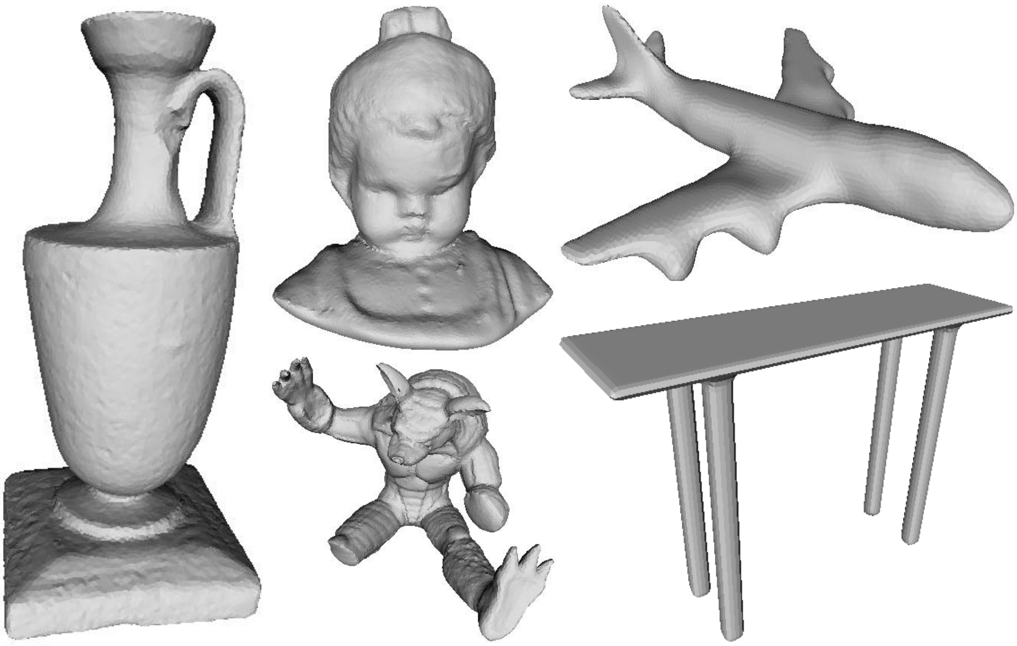


Figure 8.2: Some of the models of the experimental dataset.

where V is the number of mesh vertices.

All algorithms have been implemented in MATLAB and C++ and tested on a PC running on an Intel Core 2 Duo T6570 2.1 GHz processor with 2 GB memory. The training stage requires longer computational time than the test stage does. That is, the algorithmic complexity for training m 3D models is $\mathrm{O}\big(m \cdot n\big)$, while it is $\mathrm{O}\big(n\big)$ for testing a single model. Here, n is the number of edges for a 3D model. In our current non-optimized implementation, it takes about 40 minutes to compute the feature vectors in the training dataset and less than one second to construct a classifier. Notice that this one-off training process is the computational bottleneck of the algorithm. After the classifiers have been computed, it takes about six seconds to analyze a test model with 10,000 vertices.

The experimental dataset consists of the 360 models in Princeton Shape Benchmark [186] and four models from the Stanford 3D Scanning Repository. We assumed that all these 364 downloadable models were unmarked. Notice that the 3D models in the dataset vary greatly in overall shape and size. Fig. 8.2 illustrates a part of the experimental dataset.

Table 8.1: Clean and marked models in the training and test datasets.

Method	Training Dataset		Test Dataset	
	#Clean	#Marked	#Clean	#Marked
Cho's Mean	260	267	104	92
Cho's Variance	260	255	104	98
Laplacian-based	260	263	104	101
Chao's	260	262	104	100
LSB	260	262	104	100
All Methods	260	350	104	118

8.2.1 Specific Steganalyzer

To construct a steganalyzer specific for each one of the five embedding methods, training and test datasets were produced separately. Table 8.1 shows the number of clean and marked models in both datasets for each watermarking method.

Regarding the output of the experiments, apart from measuring the accuracy of the constructed steganalyzers, we would also want to be able to detect any possible large information redundancy in the feature vector $\mathbf{F}_i$, which could indicate that our selection of features was not optimal. For that reason we use PCA to reduce the dimension of $\mathbf{F}_i$. Figs. 8.3 and 8.4 plot the detection accuracy of the steganalyzers, computed on the test sets of the five implemented embedding techniques, against the number of principal components retained from the feature vector. We notice that the right choice of number of principal components boosts in all cases the accuracy to above 80%. However, the figures also show near optimal rates when all components of $\mathbf{F}_i$ are retained, meaning that each component of $\mathbf{F}_i$ is likely to contribute positively to the process.

From Fig. 8.4 we deduce that, as expected, the detection accuracy increases monotonously with the amount of embedding-induced distortion. That is, we obtain better results on marked models with large distortion. The comparison of Figs. 8.3 and 8.4 indicates that, in descending order, the anti-steganalysis performance of

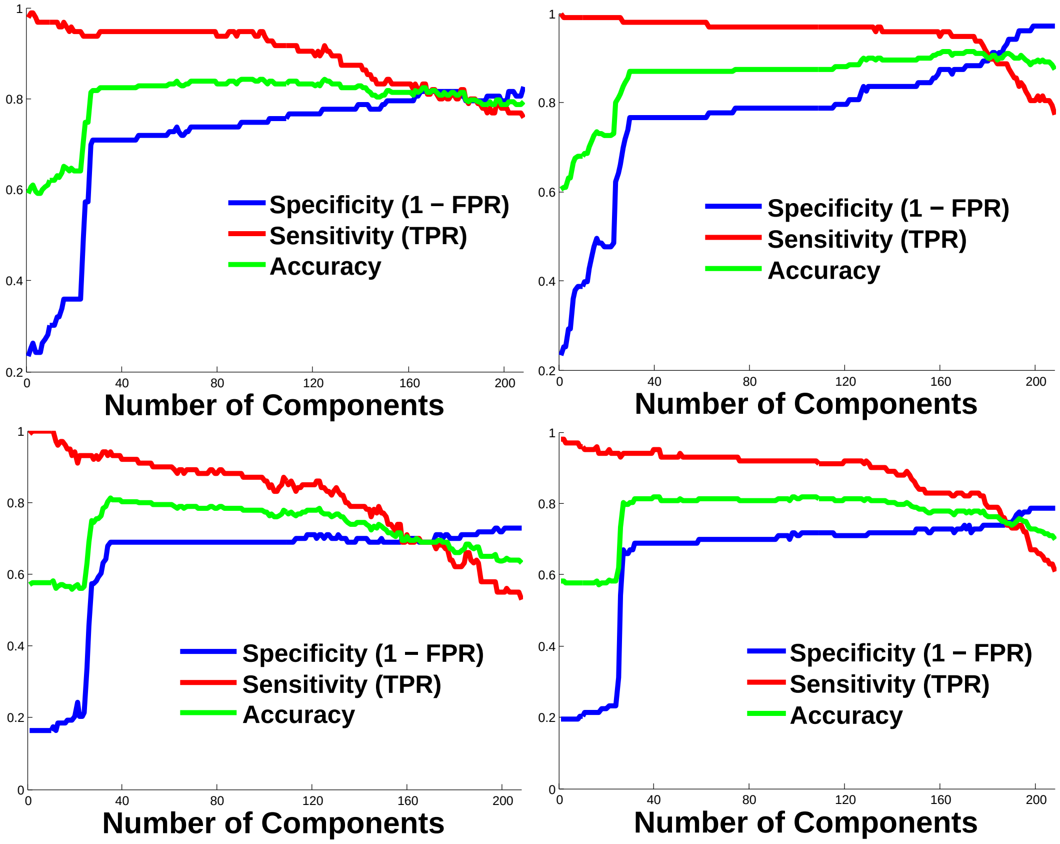


Figure 8.3: Detection accuracy plotted against the number of principal components for Cho's mean and variance-based watermarking methods [7], Yang's Laplacian coordinates based watermarking [9] and Chao's high-capacity steganography [6] (from left to right). Here, TPR and FPR indicate true positive rate and false positive rate, respectively.

the five tested methods ranks as follows: Yang's watermarking, Chao's steganography, Cho's mean-based watermarking, LSB modification and Cho's variance-based watermarking, where the accuracy rate is greater than 90%.

8.2.2 Universal Steganalyzer

In addition to the five specific steganalyzers, we also constructed a *universal steganalyzer*. The training set was created by randomly selecting 260 clean models from the dataset, creating 350 marked models by randomly selecting clean models from the training set and marking them with a randomly chosen embedding algorithm,

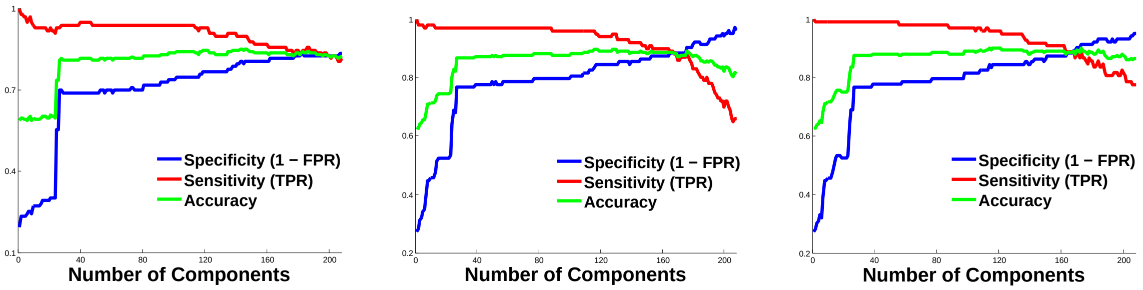


Figure 8.4: Detection accuracy plotted against the number of principal components for LSB steganography [10] with expected normal distortion of $\epsilon = 1°$, $\epsilon = 2°$ and $\epsilon = 5°$ (from left to right). Here, TPR and FPR indicate true positive rate and false positive rate, respectively.

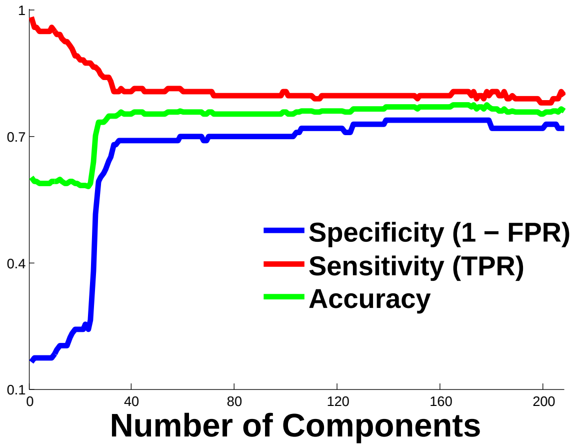


Figure 8.5: Average detection rate for all five embedding methods, plotted against the number of retained principal components. Here, TPR and FPR indicate true positive rate and false positive rate, respectively.

and finally mixing them all together into a training set of 610 models.

Fig. 8.5 shows the average detection rate for the five embedding methods, against the number of principal components retained from the feature vector. The accuracy is about 75%. Moreover, the figure indicates that a near optimal detection rate is achieved when all components of $\mathbf{F}_i$ are used, which again means that each component of $\mathbf{F}_i$ is likely to have played a useful role in the steganalysis process.

A comparison of Figs. 8.3, 8.4 and 8.5 shows that, as expected, the universal steganalyzer achieves lower detection rates compared to the specific steganalyzers.

Nevertheless, the main motivation for creating a universal steganalytic algorithm is that it can be used as a benchmark for measuring the anti-steganalysis performance of other existing and most importantly future watermarking/steganographic algorithms.

8.3 Summary

We have presented a steganalytic algorithm for the detection of hidden messages in triangle meshes. The algorithm has been evaluated on five state-of-the-art 3D watermarking/steganographic methods with satisfactory accuracy rates. We think that the high success rate may be partly due to a certain lack of sophistication in the current state-of-the-art of 3D watermarking/steganography and that algorithms with better anti-steganalytic performance should be developed.

Chapter 9

Conclusions and Future Work

In this chapter, we summarize the work covered in this thesis covers and discuss the contributions and limitations. Finally, we outline the possible directions for future work.

9.1 Summaries

In Chapters 1 and 2, we have discussed the context of the thesis background within the broader areas of information hiding and steganalysis, have stated the research problem and presented the findings of the survey of prior research related to the thesis.

In Chapters 3 to 5, we have studied how the face normals and the Discrete Gaussian Curvature (DGC) will degrade when modifying the vertex coordinates of 3D triangle models. Using these results, we presented a simple application on LSB-based 3D steganography. The proposed steganographic method has two apparent advantages over the existing techniques, namely, high-capacity and the ability to control the embedding distortion.

In Chapters 6 and 7, we proposed two novel algorithms for watermarking 3D models. The algorithm in Chapter 6, which is based on the histogram of the radial distances of the mesh vertices, has been experimentally shown to withstand a wide range of attacks to 3D models, such as noise addition, quantization and smoothing. The algorithm in Chapter 7, which is based on the histogram of the norms of the Laplacian coordinates, has been specially designed to be resistant to editing operations that alter the global shape of the 3D mesh.

In Chapter 8, based on the existing work on image steganalysis, we proposed a universal 3D steganalysis, which, to the best of our knowledge, is the first steganalytic algorithm in the 3D domain. The algorithm computes mesh features that are sensitive to watermark embedding and uses them to train a binary classifier.

9.2 Contribution

The main contributions of this thesis can be summarized as follows:

- A systematic, mathematical and empirical study, leading to an in-depth understanding of the relationship between the normal/DGC degradation and the spatial perturbation of the mesh vertices. The relevant material has been developed in Chapters 3 to 5.

- A logistic model describing the normal degradation of a triangle mesh as the quantization level decreases and a method to calculate an appropriate mesh quantization level given a tolerance for the normal accuracy, proposed in Chapter 3.

- An exact closed-form formula, and a linear approximation of it, for the expectation of the normal degradation when noise is added to a single vertex of a triangle, obtained in Chapter 4. An approximation, and heuristic upper and lower bounds, for the expectation of the normal perturbation when noise is added to all three vertices of a triangle, derived in Chapter 4.

- Fast computation of the dithered quantization level of a vertex when a tolerance for the normal degradation is given and also a LSB-based 3D steganographic algorithm that can claim maximum capacity for the given tolerance of normal degradation, presented in Chapter 4. In addition to the high embedding capacity, a second advantage is user control by making possible to keep the embedding distortion below a pre-defined threshold.

- A robust 3D watermarking algorithm and a specific steganalytic algorithm against the watermarking by Cho et al. [7] proposed in Chapter 6. A Laplacian coordinate-based watermarking algorithm that is robust to 3D editing operations, presented in Chapter 7.

- The first universal steganalytic technique in the literature for 3D triangle meshes, based on a framework that has previously been used in image steganalysis, proposed in Chapter 8. This steganalytic algorithm has been tested on five well-known 3D watermarking/steganographic methods and has achieved satisfactory detection accuracy on each one of them. More importantly, as a universal method, it could possibly be used for breaking future watermarking/steganographic algorithms.

9.3 Limitations

The main limitations of our work can be summarized as follows:

- The assumption that the meshes are "clean" with no significant amount of noise when computing appropriate levels of quantization of mesh vertices in Chapters 3 to 5.

- The linear approximation in Chapter 4 of the expectation of the normal degradation when noise was added to a single vertex held only under the addition of small amounts of noise.

- The fragility of the proposed high-capacity data hiding method in Chapter 4, that is, the fact that it is not robust under the most attacks and that even mild mesh processing operations may lead to the complete loss of the embedded message bits.

- The fragility of the histogram-based watermarking method in Chapter 6 against mesh editing operations and the fragility of the Laplacian coordinate-based watermarking in Chapter 7 against common mesh processing attacks rather than mesh editing operations.

- The scope of the proposed steganalytic method in Chapter 8, that is, it only works for those algorithms that hide information into the mesh geometry, not into mesh connectivity, or into the data redundancy of polygonal list files.

9.4 Future Work

This thesis has provided a new understanding of the relationships between normal/DGC degradation and spatial degradation and has proposed some novel steganographic, watermarking and steganalytic algorithms for 3D models. However, there is still a large scope for further improvements and several questions remained unanswered as summarized below.

Parameter Computation: In Chapters 3 and 5, there are two parameters a and b in their respective developed mathematical models. As yet, we compute them by means of fitting a model to the experimental results and cannot obtain them directly without going through the fitting process. We believe that these two numbers may be capturing interesting intrinsic properties of the triangle mesh related to the number, size and shape of the triangles in the mesh. As a direction for future work, we plan to develop methods for their direct computation and study their relation with other intrinsic properties of the mesh.

Robust 3D Watermarking: The robustness against mesh editing operations has been recently recognized as a basic property that a good 3D watermarking should possess and thus has attracted the attention of researchers. However, as yet, there exist only a handful of methods that offer resistance to mesh editing. That means that more research effort should be directed towards developing mesh editing-resistant watermarking algorithms. Mean value coordinates and harmonic coordinates have demonstrated good behavior under mesh editing operations and thus, they are good candidates as potential watermark carriers.

3D Steganalysis: Although the features extracted by the proposed method in Chapter 8 have been shown to be adequate for the purpose of 3D steganalysis, one would hope to come up with other features that are more sensitive to message embedding and thus the improved detection performance. Another possible future direction for the 3D steganalysis work is to devise steganalytic algorithms based on the modification of mesh connectivity rather than geometry.

Message Length Estimation: The proposed 3D steganalysis framework can only respond to the question of whether a given 3D model contains secret information and it is not capable of estimating the size of the information embedded within in the model. Message length estimation has been found to be valuable information for the steganalysts, whose ultimate goal is to extract and decipher the secret message, especially in the absence of information on the steganographic scheme and the cipher keys. One could probably attempt to tackle the message length estimation problem for the 3D domain by using the ideas developed for hidden message length estimation in digital images.

Bibliography

[1] G. Ye. Image scrambling encryption algorithm of pixel bit based on chaos map. *Pattern Recognition Letters*, 31(5):347–354, 2010. xiii, 2

[2] T. Gao and Z. Chen. A new image encryption algorithm based on hyper-chaos. *Physics Letters A*, 372(4):394–400, 2008. xiii, 2

[3] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding—a survey. *Proceedings of the IEEE*, 87(7):1062–1078, 1999. xiii, 3, 5, 6

[4] Y. Yang, X. Sun, H. Yang, and C.-T. Li. Removable visible image watermarking algorithm in the discrete cosine transform domain. *Journal of Electronic Imaging*, 17(3):033008–1–033008–11, 2008. xiii, 7, 21

[5] F. Cayre and B. Macq. Data hiding on 3-D triangle meshes. *IEEE Trans. on Signal Processing*, 51(4):939–949, 2003. xiii, xxi, 23, 24, 81, 126

[6] M.-W. Chao, C.-H. Lin, C.-W. Yu, and T.-Y. Lee. A high capacity 3D steganography algorithm. *IEEE Trans. on Visualization and Computer Graphics*, 15(2):274–284, 2009. xiii, xix, xxi, 25, 26, 81, 126, 135, 138

[7] J.-W. Cho, R. Prost, and H.-Y. Jung. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. *IEEE Trans. on Signal Processing*, 55(1):142–155, 2007. xiv, xviii, xix, xxi, 15, 31, 32, 33, 92, 93, 94, 105, 110, 123, 126, 127, 133, 135, 138, 143

[8] A. Bronstein, M. Bronstein, and R. Kimmel. *Numerical geometry of non-rigid shapes.* Springer, 2008. xiv, 34

[9] Y. Yang and I. Ivrissimtzis. Polygonal mesh watermarking using laplacian coordinates. *Computer Graphics Forum (SGP 2010)*, 29(5):1585–1593, 2010. xix, 37, 81, 112, 135, 138

[10] Y. Yang, N. Peyerimhoff, and I. Ivrissimtzis. Linear correlations between spatial and normal noise in triangle meshes. *IEEE Trans. on Visualization and Computer Graphics*, 19(1):45 – 55, 2013. xix, 26, 63, 135, 139

[11] C.-M. Wang and Y.-M. Cheng. An efficient information hiding algorithm for polygon models. In *EUROGRAPHICS*, volume 24, pages 591–600. Amsterdam: North Holland, 2005. xxi, 24, 126

[12] S. Zafeiriou, A. Tefas, and I. Pitas. Blind robust watermarking schemes for copyright protection of 3D mesh objects. *IEEE Trans. on Visualization and Computer Graphics*, 11(5):596–607, 2005. xxi, 32, 81, 126

[13] R. Ohbuchi, A. Mukaiyama, and S. Takahashi. A frequency-domain approach to watermarking 3D shapes. *Computer Graphics Forum*, 21(3):373–382, 2002. xxi, 36, 126, 127

[14] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687, 1997. 1, 2, 8, 21, 34

[15] R. L. Rivest, A. Shamir, and L. Adleman. A method for obtaining digital signatures and public-key cryptosystems. *ACM Commun.*, 21(2):120–126, 1978. 1

[16] National Institute of Standards and Technology (NIST). *Digital Signature Standard (DSS)*, May 1994 (accessed 15 November 2012). http://www.itl.nist.gov/fipspubs/fip186.htm. 1

[17] N. Koblitz. Elliptic curve cryptosystems. *Mathematics of computation*, 48(177):203–209, 1987. 1

[18] I. J. Cox, M. L. Miller, J. A. Bloom, J. Fridrich, and T. Kalker. *Digital watermarking and steganography*. Morgan Kaufmann, 2008. 3, 5

[19] R. J. Anderson and F. A. P. Petitcolas. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, 16(4):474–481, 1998. 3, 6

[20] D.-C. Wu and W.-H. Tsai. A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, 24(9-10):1613–1626, 2003. 4

[21] C.-C. Chang, J.-Y. Hsiao, and C.-S. Chan. Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy. *Pattern Recognition*, 36(7):1583–1595, 2003. 4

[22] S. Katzenbeisser and F. Petitolas. Information hiding techniques for steganography and digital watermaking. *EDPACS*, 28(6):1–2, 2000. 4

[23] Y. Wang and P. Moulin. Optimized feature extraction for learning-based image steganalysis. *IEEE Trans. on Information Forensics and Security*, 2(1):31–45, 2007. 4, 6, 10, 43, 134

[24] J. Qin, X. Sun, X. Xiang, and C. Niu. Principal feature selection and fusion method for image steganalysis. *Journal of Electronic Imaging*, 18(3):033009–1–033009–14, 2009. 4, 10

[25] N. F. Johnson and S. Jajodia. Exploring steganography: Seeing the unseen. *Computer*, 31(2):26–34, 1998. 5

[26] L. M. Marvel, Jr. Boncelet, C. G., and C. T. Retter. Spread spectrum image steganography. *IEEE Trans. on Image Processing*, 8(8):1075–1083, 1999. 5, 6

[27] A. Kerckhoffs. La cryptographie militaire. *Journal des sciences militaires*, 9(5-38):161–191, 1883. 5

[28] D. Kahn. *The Codebreakers: the story of secret writing*. Scribner Book Company, 1967. 6

[29] A. Cheddad, J. Condell, K. Curran, and P. McKevitt. A comparative analysis of steganographic tools. In *The 7th Information Technology and Telecommunication Conference*, pages 29–37, Dublin, Ireland, 2007. Citeseer. 6

[30] H. Farid. Detecting hidden messages using higher-order statistical models. In *Proc. IEEE Int. Conf. Image Process.*, volume 2, pages II–905–II–908, 2002. 6, 10, 11, 43, 133, 135

[31] İ. Avcibaş, N. Memon, and B. Sankur. Steganalysis using image quality metrics. *IEEE Trans. on Image Processing*, 12(2):221–229, 2003. 6

[32] J. Meng and S.-F. Chang. Embedding visible video watermarks in the compressed domain. In *Proc. IEEE Int. Conf. Image Process.*, volume 1, pages 474–477, Chicago, IL, 1998. 7

[33] Y. Hu, S. Kwong, and J. Huang. An algorithm for removable visible watermarking. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(1):129–133, 2006. 7

[34] Y. Hu and B. Jeon. Reversible visible watermarking and lossless recovery of original images. *IEEE Trans. on Circuits and Systems for Video Technology*, 16(11):1423–1429, 2006. 7

[35] H.-M. Tsai and L.-W. Chang. A high secure reversible visible watermarking scheme. In *IEEE International Conference on Multimedia and Expo.*, pages 2106–2109, Beijing, China, 2007. 7

[36] Y. Yang, X. Sun, H. Yang, C.-T. Li, and R. Xiao. A contrast-sensitive reversible visible image watermarking technique. *IEEE Trans. on Circuits and Systems for Video Technology*, 19(5):656–667, 2009. 7

[37] H.-M. Tsai and L.-W. Chang. Secure reversible visible image watermarking with authentication. *Signal Processing: Image Communication*, 25(1):10–17, 2010. 7

[38] N. Nikolaidis and I. Pitas. Robust image watermarking in the spatial domain. *Signal processing*, 66(3):385–403, 1998. 7, 20

[39] A. Tefas and I. Pitas. Robust spatial image watermarking using progressive detection. In *IEEE International Conference on Acoustics, Speech, and Signal*

Processing (ICASSP '01), volume 3, pages 1973–1976, Salt Lake City, UT, 2001. 7

[40] C.-W. Tang and H.-M. Hang. A feature-based robust digital image watermarking scheme. *IEEE Trans. on Signal Processing*, 51(4):950–959, 2003. 8, 13, 20

[41] S. Lee, C. D. Yoo, and T. Kalker. Reversible image watermarking based on integer-to-integer wavelet transform. *IEEE Trans. on Information Forensics and Security*, 2(3):321–330, 2007. 8

[42] A. A. Reddy and B. N. Chatterji. A new wavelet based logo-watermarking scheme. *Pattern Recognition Letters*, 26(7):1019–1027, 2005. 8

[43] H.-H. Tsai and J.-S. Cheng. Adaptive signal-dependent audio watermarking based on human auditory system and neural networks. *Applied Intelligence*, 23(3):191–206, 2005. 8

[44] Y.Q. Shi, G. Xuan, D. Zou, J. Gao, C. Yang, Z. Zhang, P. Chai, W. Chen, and C. Chen. Image steganalysis based on moments of characteristic functions using wavelet decomposition, prediction-error image, and neural network. In *IEEE International Conference on Multimedia and Expo,*, July 2005. 10

[45] T. Pevny and J. Fridrich. Multiclass detector of current steganographic methods for JPEG format. *IEEE Trans. on Information Forensics and Security*, 3(4):635–650, 2008. 11

[46] J.A. Vince and R.A. Earnshaw. *Advances in Modelling, Animation, and Rendering*. Springer-Verlag New York Incorporated, 2002. 12

[47] T.-Y. Liu and W.-H. Tsai. A new steganographic method for data hiding in microsoft word documents by a change tracking technique. *IEEE Trans. on Information Forensics and Security*, 2(1):24–30, 2007. 13

[48] P. Bassia, I. Pitas, and N. Nikolaidis. Robust audio watermarking in the time domain. *IEEE Trans. on Multimedia*, 3(2):232–241, 2001. 13, 40

[49] M. Noorkami and R. M. Mersereau. Digital video watermarking in P-frames with controlled video bit-rate increase. *IEEE Trans. on Information Forensics and Security*, 3(3):441–455, 2008. 13

[50] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. In *Proc. ACM SIGGRAPH Conf. Computer Graphics*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., 1999. 13, 35

[51] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM systems journal*, 35(3.4):313–336, 1996. 18

[52] C.K. Chan and LM Cheng. Hiding data in images by simple lsb substitution. *Pattern Recognition*, 37(3):469–474, 2004. 18

[53] J. Mielikainen. Lsb matching revisited. *Signal Processing Letters, IEEE*, 13(5):285 – 287, may 2006. 18

[54] M.U. Celik, G. Sharma, A.M. Tekalp, and E. Saber. Lossless generalized-lsb data embedding. *IEEE Trans. on Image Processing*, 14(2):253–266, 2005. 18

[55] H. Yang, X. Sun, and G. Sun. A semi-fragile watermarking algorithm using adaptive least significant bit substitution. *Information Technology Journal*, 9(1):20–26, 2010. 18

[56] J. Tian. Reversible data embedding using a difference expansion. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(8):890–896, 2003. 19

[57] A.M. Alattar. Reversible watermark using the difference expansion of a generalized integer transform. *IEEE Trans. on Image Processing*, 13(8):1147 –1156, aug. 2004. 19

[58] Hyoung Joong Kim, V. Sachnev, Yun Qing Shi, Jeho Nam, and Hyon-Gon Choo. A novel difference expansion transform for reversible data embedding. *IEEE Trans. on Information Forensics and Security*, 3(3):456 –465, sept. 2008. 19

[59] H. Noda, M. Niimi, and E. Kawaguchi. High-performance JPEG steganography using quantization index modulation in DCT domain. *Pattern Recognition Letters*, 27(5):455–461, 2006. 19

[60] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009. 19

[61] I.G. Karybali and K. Berberidis. Efficient spatial image watermarking via new perceptual masking and blind detection schemes. *IEEE Trans. on Information Forensics and Security*, 1(2):256 – 274, june 2006. 20

[62] V. Solachidis and L. Pitas. Circularly symmetric watermark embedding in 2-D DFT domain. *IEEE Trans. on Image Processing*, 10(11):1741–1753, 2001. 20

[63] S. Pereira and T. Pun. Robust template matching for affine resistant image watermarks. *IEEE Trans. on Image Processing*, 9(6):1123 –1129, jun 2000. 20

[64] Jiwu Huang, Y.Q. Shi, and Yi Shi. Embedding image watermarks in dc components. *IEEE Trans. on Circuits and Systems for Video Technology*, 10(6):974 –979, sep 2000. 21

[65] W.C. Chu. DCT-based image watermarking using subsampling. *IEEE Trans. on Multimedia*, 5(1):34–38, 2003. 21

[66] Wenwu Zhu, Zixiang Xiong, and Ya-Qin Zhang. Multiresolution watermarking for images and video. *IEEE Trans. on Circuits and Systems for Video Technology*, 9(4):545 –550, jun 1999. 21

[67] N. Kaewamnerd and K.R. Rao. Wavelet based image adaptive watermarking scheme. *Electronics Letters*, 36(4):312 –313, feb 2000. 21

[68] E. Ganic and A.M. Eskicioglu. Robust dwt-svd domain image watermarking: embedding data in all frequencies. In *Proceedings of the 2004 workshop on Multimedia and security*, pages 166–174. Citeseer, 2004. 21

[69] Wei-Hung Lin, Shi-Jinn Horng, Tzong-Wann Kao, Pingzhi Fan, Cheng-Ling Lee, and Yi Pan. An efficient watermarking method based on significant

difference of wavelet coefficient quantization. *IEEE Trans. on Multimedia*, 10(5):746 –757, aug. 2008. 22

[70] Yang Zhao, P. Campisi, and D. Kundur. Dual domain watermarking for authentication and compression of cultural heritage images. *IEEE Trans. on Image Processing*, 13(3):430 –448, march 2004. 22

[71] X. Kang, J. Huang, Y.Q. Shi, and Y. Lin. A DWT-DFT composite watermarking scheme robust to both affine transform and JPEG compression. *IEEE Trans. on Circuits and Systems for Video Technology*, 13(8):776–786, 2003. 22

[72] M.-W. Chao, C.-H. Lin, C.-W. Yu, and T.-Y. Lee. A high capacity 3D steganography algorithm. *IEEE Trans. on Visualization and Computer Graphics*, 15(2):274–284, 2009. 22

[73] R. Ohbuchi, H. Masuda, and M. Aono. Embedding data in 3D models. In *Interactive Distributed Multimedia Systems and Telecommunication Services*, pages 1–10. Springer, 1997. 23

[74] O. Benedens. Two high capacity methods for embedding public watermarks into 3-d polygonal models. In *ACM Multimedia Security Workshop*, pages 95–99, 1999. 23

[75] P. Bas. *Méthodes de tatouage d'images fonées sur le contenu*. PhD thesis, Institut Nat. Polytechn. Grenoble, 2000. 23

[76] Y.-M. Cheng and C.-M. Wang. An adaptive steganographic algorithm for 3D polygonal meshes. *The Visual Computer*, 23(9):721–732, 2007. 25

[77] N. Aspert, E. Drelie, Y. Maret, and T. Ebrahimi. Steganography for three-dimensional polygonal meshes. In *SPIE 47th Annual Meeting*, pages 705–708. Citeseer, 2002. 26

[78] H.-T. Wu and J. L. Dugelay. Reversible watermarking of 3D mesh models by prediction-error expansion. In *IEEE 10th Workshop on Multimedia Signal Processing*, pages 797–802, 2008. 26

[79] H. Wu and Y. Cheung. A high-capacity data hiding method for polygonal meshes. In *Information Hiding*, volume 4437 of *Lecture Notes in Computer Science*, pages 188–200. Springer, 2007. 27

[80] X Mao, M Shiba, and A Imamiya. Watermarking 3D geometric models through triangle subdivision. In *Proceedings of SPIE, Security and Watermarking of Multimedia Contents III*, volume 4314, pages 253 – 260, 2001. 27

[81] P. Amat, W. Puech, S. Druon, and J. P. Pedeboy. Lossless 3D steganography based on MST and connectivity modification. *Signal Processing: Image Communication*, 25(6):400–412, 2010. 27

[82] Y.-M. Cheng and C.-M. Wang. A high-capacity steganographic approach for 3D polygonal meshes. *The Visual Computer*, 22(9):845–855, 2006. 28

[83] A. Bogomjakov, C. Gotsman, and M. Isenburg. Distortion-free steganography for polygonal meshes. *Computer Graphics Forum (Proceedings of EUROGRAPHICS 2008)*, 27(2):637–642, 2008. 28

[84] S.-C. Tu, W.-K. Tai, M. Isenburg, and C.-C. Chang. An improved data hiding approach for polygon meshes. *Vis. Comput.*, 26:1177–1181, September 2010. 28

[85] C.M. Chou and D.C. Tseng. A public fragile watermarking scheme for 3D model authentication. *Computer-Aided Design*, 38(11):1154–1165, 2006. 29, 30

[86] B. L. Yeo and M. M. Yeung. Watermarking 3D objects for verification. *IEEE Computer Graphics and Applications*, 19(1):36–45, 1999. 29

[87] H.-Y.S. Lin, H.-Y.M. Liao, C.-S. Lu, and J.-C. Lin. Fragile watermarking for authenticating 3-D polygonal meshes. *IEEE Trans. on Multimedia*, 7(6):997–1006, 2005. 29

[88] H.T. Wu and Y.M. Cheung. A fragile watermarking scheme for 3D meshes. In *Proceedings of the 7th workshop on Multimedia and security*, pages 117–124. ACM, 2005. 29

[89] Yu-Ping Wang and Shi-Min Hu. A new watermarking method for 3D models based on integral invariants. *IEEE Trans. on Visualization and Computer Graphics*, 15(2):285–294, 2009. 30

[90] M. Yeung and Boon-Lock Yeo. Fragile watermarking of three-dimensional objects. In *IEEE International Conference on Image Processing (ICIP)*, volume 2, pages 442 – 446, oct 1998. 30

[91] Wei-Bo Wang, Guo-Qin Zheng, Jun-Hai Yong, and He-Jin Gu. A numerically stable fragile watermarking scheme for authenticating 3D models. *Computer-Aided Design*, 40(5):634 – 645, 2008. 30

[92] Z. Yu, H. H. S. Ip, and L. F. Kwok. A robust watermarking scheme for 3D triangular mesh models. *Pattern Recognition*, 36(11):2603–2614, 2003. 30

[93] M. Ashourian, R. Enteshari, and J. Jeon. Digital watermarking of three-dimensional polygonal models in the spherical coordinate system. *Computer Graphics International Conference*, 0:590–593, 2004. 32

[94] R. Darazi, R. Hu, and B. Macq. Applying spread transform dither modulation for 3D-mesh watermarking by using perceptual models. In *2010 IEEE International Conference on Acoustics Speech and Signal Processing (ICASSP)*, pages 1742–1745. IEEE, 2010. 32

[95] A. G. Bors. Watermarking mesh-based representations of 3-D objects using local moments. *IEEE Trans. on Image Processing*, 15(3):687–701, 2006. 32

[96] J. Bennour and J.C. Dugelay. Protection of 3D object visual representations. In *IEEE International Conference on Multimedia and Expo. (ICME)*, pages 1113–1116. IEEE, 2006. 33

[97] M. Luo and A.G. Bors. Surface-preserving robust watermarking of 3-D shapes. *IEEE Trans. on Image Processing*, 20(10):2813–2826, 2011. 33

[98] C.-H. Lin, M.-W. Chao, C.-Y. Liang, and T.-Y. Lee. A novel semi-blind-and-semi-reversible robust watermarking scheme for 3D polygonal models. *Visual*

Computer Journal (Computer Graphics International 2010), 26(6-8):1101–1111, 2010. 34

[99] K. Wang, G. Lavoué, F. Denis, and A. Baskurt. A comprehensive survey on three-dimensional mesh watermarking. *IEEE Trans. on Multimedia*, 10(8):1513–1527, 2008. 35, 36

[100] H. Hoppe. Progressive mesh. In *Proc. ACM SIGGRAPH'96*, volume 96, pages 99–108, 1996. 35

[101] M. Lounsbery, T. D. DeRose, and J. Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, 1997. 35

[102] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3D polygons using multiresolution wavelet decomposition. In *Proc. Int. Workshop on Geometric Modeling: Fundamentals and Applications'98*, volume 5, pages 296–307. Citeseer, 1998. 35

[103] F. Uccheddu, M. Corsini, and M. Barni. Wavelet-based blind watermarking of 3D models. In *Proceedings of the 2004 workshop on Multimedia and security*, pages 143–154, Magdeburg, Germany, 2004. ACM. 35

[104] S. Valette and P. Prost. Wavelet-based multiresolution analysis of irregular surface meshes. *IEEE Trans. on Visualization and Computer Graphics*, 10(2):113–122, 2004. 35

[105] M.S. Kim, S. Valette, H.Y. Jung, and R. Prost. Watermarking of 3D irregular meshes based on wavelet multiresolution analysis. In *Proceedings of the 4th international conference on Digital Watermarking*, pages 313–324, Berlin, Heidelberg, 2005. Springer-Verlag. 35

[106] W. H. Cho, M. E. Lee, H. Lim, and S. Y. Park. Watermarking technique for authentication of 3-D polygonal meshes. In *Proc. Int. Workshop on Digital Watermarking'05*, pages 259–270. Springer, 2005. 35

[107] K. Wang, G. Lavoué, F. Denis, and A. Baskurt. A fragile watermarking scheme for authentication of semi-regular meshes. *Proc. of the Eurographics Short Papers*, 8:5–8, 2008. 36

[108] K. Yin, Z. Pan, J. Shi, and D. Zhang. Robust mesh watermarking based on multiresolution processing. *Computers & Graphics*, 25(3):409–420, 2001. 36

[109] I. Guskov, W. Sweldens, and P. Schröder. Multiresolution signal processing for meshes. In *Proc. ACM SIGGRAPH*, pages 325–334. ACM Press/Addison-Wesley Publishing Co., 1999. 36

[110] B Bollobás. *Modern graph theory*. Springer Verlag, 1998. 36, 113, 132

[111] G. Taubin, T. Zhang, and G. Golub. Optimal surface smoothing as filter design. *European Conference on Computer Vision (ECCV)*, pages 283–292, 1996. 36

[112] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. In *Proc. ACM SIGGRAPH'00*, pages 279–286. ACM Press/Addison-Wesley Publishing Co. New York, NY, USA, 2000. 36

[113] F. Cayre, P. Rondao-Alface, F. Schmitt, B. Macq, and H. Maâtre. Application of spectral decomposition to compression and watermarking of 3D triangle mesh geometry. *Signal Processing: Image Communication*, 18(4):309–319, 2003. 36

[114] E.E. Abdallah, A.B. Hamza, and P. Bhattacharya. Spectral graph-theoretic approach to 3D mesh watermarking. In *Proceedings of Graphics Interface*, pages 327–334. ACM, 2007. 36

[115] J. Wu and L. Kobbelt. Efficient spectral watermarking of large meshes with orthogonal basis functions. *The Visual Computer*, 21(8):848–857, 2005. 37

[116] F. Hartung and B. Girod. Digital watermarking of raw and compressed video. In *Proc. European EOS/SPIE Symposium on Advanced Imaging and Network Technologies*, volume 2952, pages 205–213. Citeseer, 1996. 38

[117] M.D. Swanson, B. Zhu, B. Chau, and A.H. Tewfik. Object-based transparent video watermarking. In *IEEE First Workshop on Multimedia Signal Processing*, pages 369 –374, Jun 1997. 38

[118] M.D. Swanson, B. Zhu, and A.H. Tewfik. Multiresolution scene-based video watermarking using perceptual models. *IEEE Journal on Selected Areas in Communications*, 16(4):540–550, 1998. 38

[119] P. Bas and B. Macq. A new video-object watermarking scheme robust to object manipulation. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 526–529. IEEE, 2001. 38

[120] M. Barni, F. Bartolini, and N. Checcacci. Watermarking of MPEG-4 video objects. *IEEE Trans. on Multimedia*, 7(1):23–32, 2005. 39

[121] A. Piva, R. Caldelli, and A. De Rosa. A DWT-based object watermarking system for mpeg-4 video streams. In *Proceedings of IEEE International Conference on Image Processing*, volume 3, pages 5–8. IEEE, 2000. 39

[122] M. Noorkami and R.M. Mersereau. Compressed-domain video watermarking for H. 264. In *Proceedings of International Conference on Image Processing*, volume 2, pages II – 890–3. IEEE, 2005. 39

[123] G. Qiu, P. Marziliano, A.T.S. Ho, D. He, and Q. Sun. A hybrid watermarking scheme for H. 264/AVC video. In *Proceedings of the 17th International Conference on Pattern Recognition*, volume 4, pages 865–868. IEEE, 2004. 39

[124] E.I. Lin, A.M. Eskicioglu, R.L. Lagendijk, and E.J. Delp. Advances in digital video content protection. *Proceedings of the IEEE*, 93(1):171 –183, Jan 2005. 39

[125] G. Doërr and J.L. Dugelay. A guide tour of video watermarking. *Signal processing: Image communication*, 18(4):263–282, 2003. 39

[126] J.F. Tilki and AA Beex. Encoding a hidden digital signature onto an audio signal using psychoacoustic masking. In *Proc. 7th Int. Conf. Digital Signal Processing Applications & Technology*, pages 476–480, 1996. 39

[127] M.D. Swanson, B. Zhu, A.H. Tewfik, and L. Boney. Robust audio watermarking using perceptual masking. *Signal Processing*, 66(3):337–355, 1998. 39

[128] In-Kwon Yeo and Hyoung Joong Kim. Modified patchwork algorithm: a novel audio watermarking scheme. *IEEE Trans. on Speech and Audio Processing*, 11(4):381 – 386, july 2003. 39

[129] X.Y. Wang and H. Zhao. A novel synchronization invariant audio watermarking scheme based on DWT and DCT. *IEEE Trans. on Signal Processing*, 54(12):4835–4840, 2006. 39

[130] Shaoquan Wu, Jiwu Huang, Daren Huang, and Y.Q. Shi. Efficiently self-synchronized audio watermarking for assured audio data transmission. *IEEE Trans. on Broadcasting*, 51(1):69 – 76, march 2005. 40

[131] S.K. Lee and Y.S. Ho. Digital audio watermarking in the cepstrum domain. *IEEE Trans. on Consumer Electronics*, 46(3):744–750, 2000. 40

[132] J.T. Brassil, S. Low, and N.F. Maxemchuk. Copyright protection for the electronic distribution of text documents. *Proceedings of the IEEE*, 87(7):1181 –1196, Jul 1999. 40

[133] M.J. Atallah, C.J. McDonough, V. Raskin, and S. Nirenburg. Natural language processing for information assurance and security: an overview and implementations. In *Proceedings of the 2000 workshop on New security paradigms*, pages 51–65. ACM, 2001. 40

[134] M. Atallah, V. Raskin, M. Crogan, C. Hempelmann, F. Kerschbaum, D. Mohamed, and S. Naik. Natural language watermarking: Design, analysis, and a proof-of-concept implementation. In *Information Hiding*, pages 185–200. Springer, 2001. 40

[135] M. Atallah, V. Raskin, C. Hempelmann, M. Karahan, R. Sion, U. Topkara, and K. Triezenberg. Natural language watermarking and tamperproofing. In *Information hiding*, pages 196–212. Springer, 2003. 40

[136] X. Sun, G. Luo, and H. Huang. Component-based digital watermarking of chinese texts. In *Proceedings of the 3rd international conference on Information security*, pages 76–81. ACM, 2004. 41

[137] Y. Liu, X. Sun, and Y. Wu. A natural language watermarking based on chinese syntax. *Advances in Natural Computation*, pages 435–435, 2005. 41

[138] H.M. Meral, E. Sevinc, E. Unkar, B. Sankur, A.S. Ozsoy, and T. Gungor. Syntactic tools for text watermarking. In *Proceedings of the SPIE Conference on Security, Steganography, and Watermarking of Multimedia Contents*, volume 6505, pages 65050X–1 – 65050X–12, 2007. 41

[139] J. Fridrich, M. Goljan, and R. Du. Detecting LSB steganography in color, and gray-scale images. *IEEE Multimedia*, 8(4):22–28, 2001. 41

[140] S. Dumitrescu, Xiaolin Wu, and Zhe Wang. Detection of LSB steganography via sample pair analysis. *IEEE Trans. on Signal Processing*, 51(7):1995 – 2007, july 2003. 41

[141] J.J. Harmsen and W.A. Pearlman. Steganalysis of additive-noise modelable information hiding. In *Electronic Imaging*, pages 131–142. International Society for Optics and Photonics, 2003. 42

[142] A.D. Ker. Steganalysis of LSB matching in grayscale images. *IEEE Signal Processing Letters*, 12(6):441–444, 2005. 42

[143] X. Yu, T. Tan, and Y. Wang. Extended optimization method of LSB steganalysis. In *IEEE International Conference on Image Processing*, volume 2, pages II–1102–5. IEEE, 2005. 42

[144] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *Information Hiding*, pages 310–323. Springer, Springer, 2002. 42, 129

[145] Bin Li, Jiwu Huang, and Yun Qing Shi. Steganalysis of YASS. *IEEE Trans. on Information Forensics and Security*, 4(3):369 –382, sept. 2009. 42

[146] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Trans. on Information Forensics and Security*, 1(1):111–119, 2006. 43

[147] G. Xuan, Y. Shi, J. Gao, D. Zou, C. Yang, Z. Zhang, P. Chai, C. Chen, and W. Chen. Steganalysis based on multiple features formed by statistical moments of wavelet characteristic functions. In *Proceedings of Information Hiding Workshop*, pages 262–277. Springer, 2005. 43

[148] W.-N. Lie and G.-S Lin. A feature-based classification technique for blind image steganalysis. *IEEE Trans. on multimedia*, 7(6):1007–1020, 2005. 43

[149] M. Yoon, I. Ivrissimtzis, and S. Lee. Variational Bayesian noise estimation of point sets. *Computers & Graphics*, 33(3):226–234, 2009. 46, 47

[150] I. Ivrissimtzis. Effects of noise on quantized triangle meshes. In *Mathematical Methods for Curves and Surfaces, LNCS, Springer*, pages 274–284, 2010. 46

[151] Y. Yang and I. Ivrissimtzis. A logistic model for the degradation of triangle mesh normals. In *7th International Conference on Curves and Surfaces, LNCS*, pages 697–710. Springer, 2012. 47

[152] C. Touma and C. Gotsman. Triangle mesh compression. In *Graphics Interface*, pages 26–34, 1998. 47

[153] P. Alliez and M. Desbrun. Progressive compression for lossless transmission of triangle meshes. In *SIGGRAPH*, pages 195–202, 2001. 47

[154] B. Walter, S. Zhao, N. Holzschuch, and K. Bala. Single scattering in refractive media with triangle mesh boundaries. In *SIGGRAPH*, pages 1–8, 2009. 47

[155] S. Jin, R. R. Lewis, and D. West. A comparison of algorithms for vertex normal computation. *The Visual Computer*, 21(1-2):71–82, 2005. 47

[156] Q. Meyer, J. Susharpmuth, G. Susharpner, M. Stamminger, and G. Greiner. On floating-point normal vectors. *Computer Graphics Forum*, 29:1405–1409(5), 2010. 47

[157] M. Pauly, N. J. Mitra, and L. J. Guibas. Uncertainty and variability in point cloud surface data. In *Symposium on Point-Based Graphics*, pages 77–84, 2004. 47

[158] A. Kalaiah and A. Varshney. Statistical geometry representation for efficient transmission and rendering. *ACM Transactions on Graphics*, 24(2):348–373, 2005. 47

[159] X. Sun, P. Rosin, R. Martin, and F. Langbein. Noise analysis and synthesis for 3d laser depth scanners. *Graphical Models*, 71(2):34–48, 2009. 47

[160] Olga Sorkine, Daniel Cohen-Or, and Sivan Toledo. High-pass quantization for mesh encoding. In *Proceedings of the Eurographics/ACM SIGGRAPH Symposium on Geometry Processing*, pages 42–51. Eurographics Association, 2003. 47

[161] L. Schuchman. Dither signals and their effect on quantization noise. *IEEE Trans. on Communication Technology*, 12(4):162–165, 1964. 48

[162] R. Gray and T. Stockham. Dithered quantizers. *IEEE Trans. on Information Theory*, 39(3):805–812, 1993. 48

[163] Curtis Roads. *The Computer Music Tutorial*. The MIT Press, 1996. 48

[164] L. Akarun, Y. Yardunci, and A.E. Cetin. Adaptive methods for dithering color images. *IEEE Trans. on Image Processing*, 6(7):950 –955, 1997. 48

[165] Leif Kobbelt, Swen Campagna, Jens Vorsatz, and Hans-Peter Seidel. Interactive multi-resolution modeling on arbitrary meshes. In *SIGGRAPH*, pages 105–114, 1998. 48

[166] Igor Guskov, Wim Sweldens, and Peter Schröder. Multiresolution signal processing for meshes. In *SIGGRAPH*, pages 325–334, 1999. 48

[167] Hugues Hoppe. View-dependent refinement of progressive meshes. In *SIGGRAPH*, pages 189–198, 1997. 48

[168] W. Pasman and F. W. Jansen. Scheduling level of detail with guaranteed quality and cost. In *Web3D '02*, pages 43–51. ACM, 2002. 48

[169] Liang Hu, Pedro V. Sander, and Hugues Hoppe. Parallel view-dependent refinement of progressive meshes. In *I3D '09*, pages 169–176. ACM, 2009. 48

[170] H. Pham, editor. *Springer Handbook of Engineering Statistics*. Springer, 2006. 49

[171] Kai Wang, G. Lavoué, F. Denis, and A. Baskurt. Hierarchical watermarking of semiregular meshes based on wavelet transform. *IEEE Trans. on Information Forensics and Security*, 3(4):620–634, 2008. 81

[172] D Zwillinger. *CRC standard mathematical tables and formulae*. CRC Press LLC, 2003. 94

[173] P Cignoni, C Rocchini, and R Scopigno. Metro: measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998. 99

[174] Kai Wang, Guillaume Lavoué, Florence Denis, Atilla Baskurt, and Xiyan He. A benchmark for 3d mesh watermarking. In *Shape Modeling International Conference (SMI)*, pages 231–235. IEEE, 2010. 101

[175] G Taubin. Geometric signal processing on polygonal meshes. *Eurographics State of the Art Reports*, pages 81–96, 2000. 103

[176] http://www.blender.org/. 104, 124

[177] Euisun Choi and Chulhee Lee. Feature extraction based on the bhattacharyya distance. *Pattern Recognition*, 36(8):1703 – 1709, 2003. 107

[178] Xiaobai Chen, Aleksey Golovinskiy, and Thomas Funkhouser. A benchmark for 3D mesh segmentation. *ACM Trans. on Graphics (Proc. SIGGRAPH)*, 28(3):73:1–73:12, Aug 2009. 108

[179] O Sorkine, D Cohen-Or, Y Lipman, M Alexa, C Rössl, and H.-P Seidel. Laplacian surface editing. In *Proceedings of the Eurographics/ACM SIGGRAPH*

Symposium on Geometry Processing, pages 179–188, Nice, France, 2004. Eurographics Association. 113

[180] Gabriel Peyré. http://www.ceremade.dauphine.fr/∼peyre/, 2008. 119

[181] Jan Kodovský and Jessica Fridrich. Calibration revisited. In *Proceedings of the 11th ACM workshop on Multimedia and security*, pages 63–74, New York, NY, USA, 2009. ACM. 129

[182] Y. Higuchi. Combinatorial curvature for planar graphs. *Journal of Graph Theory*, 38(4):220–229, 2001. 132

[183] David Freedman and Persi Diaconis. On the histogram as a density estimator: L_2 theory. *Probability Theory and Related Fields*, 57(4):453–476, 1981. 134

[184] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3–14, January 2007. 134

[185] W.J. Krzanowski. *Principles of multivariate analysis*. Oxford University Press, 2000. 134

[186] P. Shilane, P. Min, M. Kazhdan, and T. Funkhouser. The princeton shape benchmark. In *Proceedings of Shape Modeling Applications*, pages 167–178. IEEE, 2004. 136