Robust and Blind 3D Watermarking

Ming Luo

Submitted for the degree of Doctor of Philosophy

Department of Computer Science

THE UNIVERSITY of York

May 2010

Abstract

3D watermarking is a technique to hide some information into the 3D graphical model in such a way that the watermarked object is visually indistinguishable from the original one. A robust and blind 3D watermarking method should be able to detect the embedded message after a certain level of malicious attack without having the original model. 3D watermarking has a great potential of usage in the real world and it can be applied in the copyright protection, database management, graphics authentication and data transmission etc. This thesis proposes four novel robust and blind 3D watermarking methods based on spectral domain and spatial domain.

Chapter 2 comprehensively surveys the related literature in the fields of transformed domain methods, spatial domain methods and the watermarking metrics. Chapter 3 proposes a novel 3D watermarking methodology in the spectral domain. The mesh object is decomposed into a set of spectral coefficients which represent the energy of the mesh in different scales. The message is embedded by introducing constraints into the distributions of spectral coefficients.

Chapter 4 employs the geodesic distance to carry the bits based on the observation that the distribution of geodesic distance within a range is close to uniform. Two ways of embedding scheme are introduced. One is to modify the mean value and the distribution and the other is to change the variance. A novel Vertex Placement Scheme (VPS) is proposed to move the vertex in order to satisfy the watermarked geodesic distance, without causing significant distortion to the object.

Chapter 5 introduces two spatial domain methods which embed the message by changing the distribution of the vertex norms, *i.e.* the distance from vertex to the object centre. Two methods employ the same histogram mapping function as described in chapter 4. The first method minimizes the surface distortion by selecting a candidate point over the neighbourhood which introduces the minimum error. The second method employs the Levenberg-Marquardt optimization method to find the best possible solution to ensure that the surface distortion is truly minimum with respect to a novel surface error function.

The algorithms proposed in this thesis significantly improve the visual quality of the watermarked object while the watermark detection robustness is at a relatively high level. The robustness of the proposed methods is increasing from the methods presented in Chapter 3 to Chapter 5 while the surface distortion is decreasing for these methods. The second algorithm proposed in Chapter 5 achieves the best overall performance in the aspect of visual quality and robustness. In Chapter 6, we conclude the thesis by addressing the weakness and propose potential future research work.

Acknowledgments

Firstly, I would like to thank my parents for their endless love, encouragement and unconditional support during the six years of studying in UK.

Also, I would like to thank my supervisor, Dr Adrian Bors, for his helpful advice and valuable comments about my paper work. I also thank my assessor Dr Richard Wilson for his help on mathematics and programming.

My thanks go to all the rest of the people in the computer vision group for all the discussions, friendship and entertainment.

Special thanks go to my girlfriend Ms Bi as she brings fortune and real happiness to me in the last stage of PhD study and also a bright future in my life.

Declaration

I hereby declare that all the work in this thesis is solely my own, except where attributed and cited to another author. Most of the material in this thesis has been previously published by the author. A complete list of publications can be found on page iii.

List of Publications

Journal Papers

1. Ming Luo and Adrian Bors. Robust blind 3D watermarking minimizing the surface error, *IEEE Transaction on Visualization and Computer Graphics*, (submitted).

Conference Papers

1. Ming Luo, Kai Wang, Adrian Bors and Guillaume Lavoué, Local patch blind spectral watermarking method for 3D graphics. In *International Workshop on Digital Watermarking*, pages 211-226, 2009

2. Kai Wang, Ming Luo, Adrian Bors, F. Denis. Blind and robust mesh watermarking using manifold harmonics. In *Proceedings of the IEEE International Conference on Image Processing*, 2009 (to appear).

3. Ming Luo and Adrian Bors, Shape watermarking based on minimizing the quadric error metric, In *Proceedings of IEEE International Conference on Shape Modeling and Applications (SMI)*, pages 103-110, 2009

4. Ming Luo and Adrian Bors, Principal component Analysis of spectral coefficients for mesh watermarking, In *Proceedings of IEEE International Conference on Image Processing*, pages 441-444, 2009

Contents

List of Figures

List of Tables

List of Symbols

$\mathcal{O}$	It represents a 3D manifold mesh object.
$\mathcal{V}$	The vertex set of the mesh object $\mathcal{O}$
$\mathcal{F}$	The face set of the mesh object $\mathcal{O}$
$\mathbf{v}_i$	The coordinate vector of the i^{th} vertex of the mesh. In this thesis, we use the lower and bold case letter to represent the coordinate vector of a vertex or a point in 3D space.
V_i	Label of the i^{th} vertex of the mesh. In this thesis, weuse the upper case letter to label a vertex or a point in 3D space.
$\mathbf{F}_i$	The i^{th} face of the mesh.
$\mathcal{M}$	The message is going to be embedded in the object.
$\hat{\mathcal{M}}$	The retrieved message from the watermarked mesh object.
$\mathbf{L}$	The Laplacian matrix.
μ	The mean value of a statistical variable.
$\mathbf{C}$	Spectral coefficients.
$T_{\mathcal{O}}(\mathbf{x}, \mathbf{y})$	Geodesic distance between two point $\mathbf{x}$ and $\mathbf{y}$ on the manifold.
g	The geodesic distance of a vertex.
ρ_i	The Euclidean distance between the object centre and the i^{th} vertex.

Chapter 1

Introduction

Nowadays, Internet is an important part in our life. There are numerous types of digital data over the Internet, such as text, images, audio tracks, videos and 3D graphical objects etc. These are used more and more in industrial, medical and entertainment applications. Digital data are so widely used because they are easy to store, transfer and duplicate with high quality. However, the convenience also facilitates the access of malicious users to produce unauthenticated and pirate copies of the original work.

There are two typical digital data protection techniques: cryptography and watermarking. Cryptography completely changes the appearance of the data and as a result nobody would be able to decode the message without the secret key. Cryptography is often used in the transmission stage. Users have to decode the message before they can read or use the data. In contrast, watermarking preserve the observable quality of the data, for example the image fidelity, audio and video quality, in such a way that people can use the data without being aware of the existence of the embedded message. Watermarking can be used both in transmission and for data usage. In addition, cryptography aims to modify every single bit of the original data. In Digital Watermarking the aim is to embed a code consisting of bits into a cover media, representing image, audio, video or graphics information.

Digital watermarking is generally considered as a copyright protection technique. Digital watermarking consist in embedding information into the "cover media" in such a way that the embedded "stego media" is perceptually no different from the

original one. In principal, any kind of information that can be encoded in binary format can be embedded into the cover media, for example, copyright, creation information, content description etc. The message can be used in various ways for different purposes such as copyright protection, authentication and database indexing etc.

In the last decade, digital watermarking become an active research area and many watermarking techniques have been proposed for audio [22], [73] images [14], [121] and videos [27], [137]. The conventional 1D/2D digital data watermarking are relatively mature and their applications begin to appear in the commercial market, for example, you can watermark your own photo using the Adobe Photoshop. On the other hand, 3D watermarking is still in its infancy and there is no "accepted" solution especially for robust and blind 3D watermarking methods.

Watermarking methods have close relation with the research area of computer vision, pattern recognition, computer graphics, theory of transmission of information and approximation theory. Various techniques have been used for watermarking purpose. Many transformed domain methods embed information in the spectral coefficients of the mesh. While spectral graph theory has been used extensively in computer vision [144] and pattern recognition [88, 105]. Wavelet decomposition, which is an important multi-resolution analysis method developed in computer graphics community [113, 125], is also used in watermarking purposes. Hausdorff distance is used for evaluating the similarity between the watermarked object and the original object [34].

In this chapter, the fundamentals of 3D watermarking will be presented first. This includes fundamental watermarking concepts, watermarking properties and the difference between the 3D watermarking and traditional 1D/2D methods. Then, we envisage the applications of 3D watermarking techniques. The contributions of this thesis are given in the third part of this chapter. And finally, the structure of this thesis is given at the end of the chapter.

1.1 Fundamentals of 3D watermarking

1.1.1 Terminology

Before we enter into the details, it is necessary to clarify the specific terminology used throughout this thesis.

- **Cover medium**: An original digital medium (3D graphical object in our case) without being watermarked or processed is called cover medium (or cover object).
- **Stego medium**: When the cover medium is watermarked by some watermarking algorithms, watermarked medium object is then called stego medium.
- **Watermark**: Watermark is the message being inserted to the cover object. We use $\mathcal{M}$ to represent the message to be embedded, while $\hat{\mathcal{M}}$ denotes the retrieved message from the object that is watermarked or attacked.
- **Watermark embedding**: It is the process of inserting the watermark into the cover object.
- **Watermark detection**: It is the process of retrieving the embedded message.
- **Robustness**: We measure the robustness of a watermarking algorithm using the Bit Error Rate, *i.e.* the ratio between the correctly detected bits and the total number of embedded bits.
- **Distortion**: It means the similarity between the watermarked object and the original one. In this thesis, we use two criteria, both objectively and subjectively, to evaluate the distortion introduced by watermarking.

1.1.2 3D representations

A 3D object can be represented in various ways: using voxels, polygonal meshes, constructive solid geometry or as an implicit set of parametrized equations, such as nonuniform rational B-splines (NURBS) etc. However, triangulated meshes are considered as the denominator of the 3D objects since any other representations can

be converted into a triangular mesh very easily. In this thesis, we consider graphical objects are represented by triangular meshes.

A triangle mesh object $\mathcal{O}$ consists of a set of vertices describing its geometry, $\mathcal{V} = \{\mathbf{v}_i \in \mathcal{O} | i = 1, \ldots, |\mathcal{O}|\}$, where $|\mathcal{O}|$ is the number of vertices of mesh $\mathcal{O}$, and a set of triangles connecting the vertices, $\mathcal{F} = \{\mathbf{F}_i \in \mathcal{O} | i = 1, \ldots, |\mathcal{F}|\}$, where $|\mathcal{F}|$ is the number of faces of the mesh $\mathcal{O}$. A set of attributes such as colour, texture, shading may be associated with each vertex or face. However, we will not consider these attributes for watermarking because they can be easily removed or modified by users. In this thesis, we consider that a triangle mesh represents a two-manifold object which means that the local neighbourhood of any vertex is topologically equivalent to a disk. In other words, a non-boundary edge of the mesh $\mathcal{O}$ must connect two neighbouring faces only.

1.1.3 Watermarking properties

A 3D watermarking system contains five main requirements including the robustness of the embedded message, capacity of the payload, blindness, security and non-distortion of the 3D surface. From the robustness point of view, the watermarking algorithms are classified as robust watermarking and fragile watermarking. A robust method aims to detect the embedded message even after the object suffered from a serious level of attacks. This category of methods are often designed for the purpose of copyright protection. On the other hand, a fragile message should disappear totally when any attack happens to the 3D mesh model. A good fragile watermarking algorithm should be able to locate the region being modified. Fragile watermarking is used for the mesh authentication and tamper detection.

Steganography represents the procedure to hide information into a cover media, usually of audio, image, video or graphics format. The aim in steganography is to embed a large amount of information. This kind of methods is usually used for transmitting data and the 3D object is considered as a message carrier in this case.

From the blindness perspective, methods can be considered as blind or non-blind depending on whether the original object is required or not as reference in the detection stage. Blind methods are more difficult to implement than non-blind methods

but at the same time they are more practical having wider potential applications.

The fourth point is that the security of the embedded message should only rely on the private key but not on the method. Theoretically, it should be impossible to retrieve the embedded message without the private key. The security is relatively easy to achieve as it can be enforced in different stages of the watermarking process.

Last but not least, the distortion of the object watermarked by any methods should be perceptually invisible with respect to the original surface. Currently, robustness and distortion are the most challenging issues of the robust and blind 3D watermarking study. In this thesis, I focus on the watermarking method that is robust and blind, while aiming to introduce a minimal distortion in the 3D surface of the graphical objects used as cover media.

Figure 1.1 illustrates the relation among the three factors: robustness, capacity and distortion with the security taken into account. Trying to improve any one part may limit the effectiveness of the others. For instance, we may obtain a higher robustness if we relax the requirement of the distortion and embed fewer bits. But the trade-off is that the object may be very distorted from the original one. Increasing the capacity means that one bit of message will be carried by less vertices. Thus, this may reduce the robustness of the watermarking algorithm. Security means how much is the opportunity that the embedded message can be recovered and removed by malicious users. It is relatively not as vital as the other three factors in watermarking methods but it can not be ignored. How to find a proper balance among these aspects is the most challenging issue in the research of 3D watermarking algorithms.

1.1.4 Difference with conventional digital data

There are three major differences between the 3D graphical object and the traditional 2D/1D data so that the techniques developed for 2D/1D watermarking can not be directly imported to 3D object watermarking.

Firstly, a 3D object is not regularly sampled. It is not possible to represent a 3D object by a single dataset. For example, an image can be interpreted as a matrix. The irregular sampling issue makes the spectral analysis techniques such as

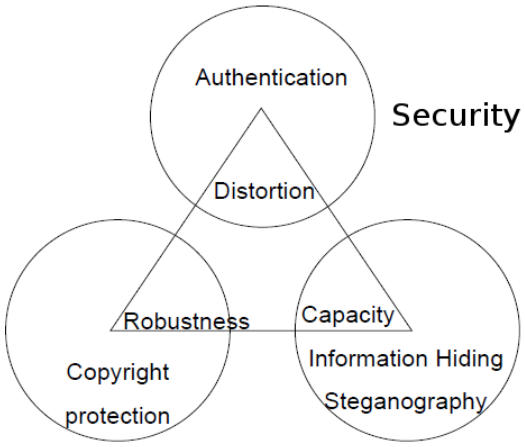


Figure 1.1: Property Triangle of 3D Watermarking

Fourier Transform, Discrete Cosine Transform and Wavelet Transform, which rely on regular sampling, more difficult to apply. Meanwhile, these techniques have been successful in the traditional 2D/1D data.

Secondly, a 3D object can be represented in an infinite ways both on geometry and connectivity. For instance, a mesh can be represented as a 1-to-4 connectivity (one vertex is connected with four neighbours), or 1-to-6 connectivity. Sometimes, regular connectivity is considered as regular sampling in 3D meshes [44, 67, 113]. Simplification techniques [50] preserves the shape of the mesh surface by using a smaller number of vertices. Uniform resampling schemes use totally new samples to represent the same object, etc. This indicates that either geometry or connectivity is weak and not reliable.

Last but not least, there is no stable intrinsic order sequence of the 3D data. Image data can be ordered easily according to the row or column information and audio and video data are streamed according to the time series. As both the geometry and the connectivity information are irregular in 3D objects, there is no explicit sequence order of 3D data. Thus, it is difficult to apply the existing successful spectral watermarking schemes, such as the one proposed in [38], on 3D meshes.

1.1.5 3D watermarking is a cutting edge area

3D object watermarking is a cutting-edge research area involving various fields such as computer vision, statistics, data-hiding, signal processing, approximation theory,

optimization theory and computer graphics, etc. Thus, 3D watermarking algorithms are very different from one to another. Signal processing techniques are mostly used in the spectral domain algorithms such as mesh spectral analysis [100], mesh wavelet decomposition [67].

In the other category, the message is embedded directly into the spatial domain. Most of the robust and blind algorithms belong to this category. This class of methods usually use statistical features of the mesh geometry and it is close to the computer graphics. Currently, the spatial domain methods based on the geometry features of the 3D model consists of the state of the art in 3D watermarking and it is a more promising area than methods based on the transformed domain.

The approximation theory is an important theoretical background of measuring the surface distortion. The algorithms proposed in Chapter 5 explicitly use an error function which is closely related with the approximation theory to control the surface distortion. Consequently, the distortion is minimized through an optimization procedure.

1.1.6 Attacks of 3D meshes

As mentioned in the previous section, a robust and blind 3D watermarking algorithm should be able to detect the embedded message even after the mesh object is suffered from a certain level of attacks. In general, there are two types of attacks. One is the distortionless attack that means any technique not modifying the geometry or the connectivity of the mesh. The following attacks belong to this category: mesh registration, rotation, uniform scaling, object description file shuffling and geometry transformation. Distortionless attack is often considered as 3D processing that is not harmful to the 3D model. Most of existing watermarking algorithms are resistant to distortionless attacks. On the other hand, distortion attacks may change the geometry or the connectivity or both properties of the mesh. In this thesis, we test the robustness of the proposed algorithms against five attacks including additive noise, Laplacian smoothing, mesh simplification, quantization and uniform resampling. Additive noise is a common attack which randomly modifies the geometry of the mesh. Mesh simplification aims to represent the same object using

less vertices. Laplacian smoothing is to remove the sharp feature and noise of the mesh. Quantization attack produces blocking effects on the object surface. Uniform resampling is a kind of combination attack of mesh simplification and Laplacian smoothing. Uniform resampling takes samples on the tangent plane of the surface and remesh the new samples to produce the new object.

1.2 Applications of 3D Watermarking

There are many potential applications of 3D watermarking because of two reasons. Firstly, any information can be embedded into the object. Secondly, the watermarked models can be used as original ones, because watermarking methods aim to embed the message without modifying the appearance of the 3D model. The content of the embedded information can be used in various ways. The most straightforward application is to protect the 3D object. Copyright information, such as author or creation date etc, can be embedded in order to protect the intellectual property of the 3D model. In a virtual 3D object market, an artist creates some wonderful 3D models. Then he can save the information such as website, price, even barcode to the object. Once the author finds some unauthorized copy over the Internet, he can claim his copyright by retrieving his own watermark code from the 3D model. Fragile watermarking can be used as an authentication or tampering detection tools. The watermark will disappear when a watermarked object is modified and ideally the detected information can tell where and how the mesh is modified.

In a 3D database, we can incorporate the information such as, database index, mesh description, category etc, into the mesh. In a large database, some objects may have the similar semantics with little different features such as two hands shown in Figure 1.2. In this case, the watermark can distinguish the two objects to accelerate the recognition or retrieval of relevant data about the given object.

It would be very difficult to replicate a 3D scene if it contains thousands of objects of various types and sizes. But if every object is embedded with a message describing its location in the scene, it would make the rendering like a jigsaw puzzle game. Using a steganography method [24], we may be able to incorporate the information of colour, material or texture of each vertex into the mesh itself. This

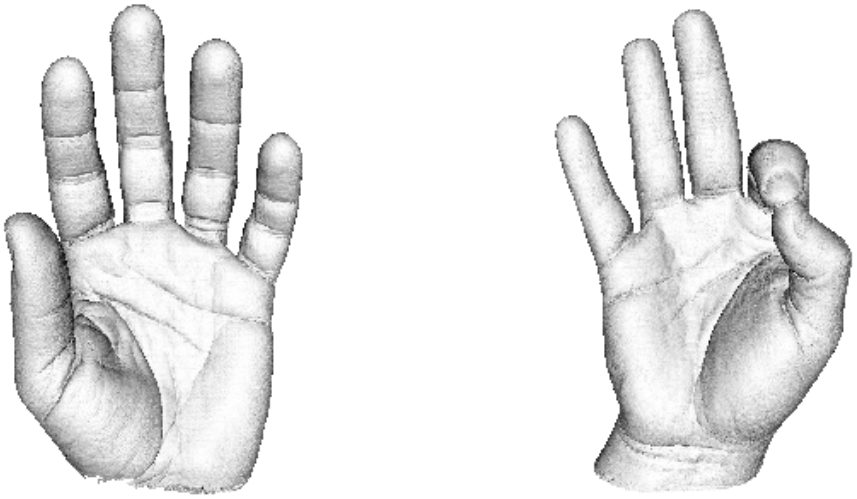


Figure 1.2: Two hands

can save lots of storage space and rendering time.

All in all, a lot more applications of using 3D watermarking algorithms in the real world can be imagined. However, there is currently no 3D algorithm that is accepted in general and applied especially in commercial and industrial market. My opinion is that the 3D algorithms proposed so far, especially the robust and blind algorithms, all introduce observable distortion and do not find a proper balance between the visual distortion and robustness.

1.3 Contributions

There are four major contributions in my thesis.

- A novel robust and blind watermarking algorithm based on the spectral domain is proposed in Chapter 3. It uses the statistical characteristics of the spectral coefficients to embed the message.

- A novel robust and blind watermarking algorithm based on geodesic distances is proposed in Chapter 4. It solves a reverse problem of calculating geodesic distance and guarantees a minimum distortion when modifying the geodesic distance.

- A novel object surface error function is proposed in Chapter 5. This function

can be used to describe the similarity between two local surfaces.

- A novel optimization watermarking methodology minimizing the error function while satisfying the watermarking constraint is proposed in Chapter 5. This algorithm, from the experiments, achieves the best perceptual and objective mesh quality, over all robust and blind watermarking methods, with respect to the original object surface while possesses a very high robustness. The optimization method bridge the gap between the surface error metric and the watermarking embedding.

In Chapter 3, the message is embedded by modifying the statistical features of the spectral coefficients of a 3D model. The method is the first one that analyze the spectral coefficients statistically and which is blind in the detection stage.

The method in Chapter 4 is motivated by the observation that the distribution of geodesic distances within a region of the surface is uniform. The message is embedded by changing the statistical features of the geodesic distribution to embed the message. We propose a novel method to solve the problem how to move a vertex in order to satisfy a given geodesic distance on the mesh.

There are two watermarking methods proposed in Chapter 5 and both share the same motivation with Cho's method in [31]. The first method called Quadric Selective Placement (QSP) method proposes a discretization method to minimize the surface distortion, while the second method named as L-M method proposes to use an optimization procedure to optimize the distortion with respect to a novel surface error function.

The robustness of the methods proposed in each chapter, from Chapter 3 to Chapter 5, is increasing, while their corresponding distortion produced in the graphical objects is decreasing.

1.4 Thesis Structure

The thesis is structured as follows:

- Chapter 2 gives a comprehensive literature review of the 3D watermarking algorithms.

- Chapter 3 proposes a robust and blind watermarking algorithm using the spectral domain.
- Chapter 4 proposes a robust and blind watermarking algorithm based on the statistical features of the geodesic distances.
- Chapter 5 proposes two robust and blind watermarking algorithms based on the optimization with respect to surface error functions.
- Chapter 6 gives the conclusion of the thesis and the future work.

Chapter 2

Literature Review

Since the 3D watermarking was firstly introduced by Ohbuchi [96,97], it is becoming an active research area during the last decade. 3D watermarking is inspired from the image watermarking and video watermarking [20, 37, 58, 101, 112, 117, 130, 147]. However, as explained in Section 1.1.4 on page 5, the techniques in 2D watermarking can not be directly applied to 3D watermarking.

Generally speaking, the 3D watermarking can be classified into transformed domain watermarking and the spatial domain watermarking from the perspective of the embedding domain. Then the transformed domain methods can be further split into spectral methods and multiresolution methods. In this chapter, I will firstly comprehensively survey the transformed domain methods followed by that of the spatial domain methods. We focus mainly on the robust methods and briefly mention the others. Then we introduce the assessment methodology of the 3D watermarking algorithms.

2.1 Spectral domain algorithms

The methods of mesh spectral analysis are inspired by the development of spectral graph theory [26], signal processing and the kernel principal component analysis and spectral clustering in the computer vision and machine learning [146]. The mesh spectral analysis of a given mesh object $\mathcal{O}$ with N vertices generally has the following three steps in common:

1. A square Laplacian matrix $\mathbf{L}$ of size $N \times N$ is constructed. The Laplacian matrix which is a discretization of a continuous operator represents a discrete linear operator based on the connectivity of the input mesh.

2. The second step is almost identical for all methods. This consists of to eigen-decomposing the matrix $\mathbf{L}$.

3. Process the calculated eigenvalues usually by embedding constraints or by adding noise, *i.e.* frequency coefficients, and the eigenvectors, *i.e.* the orthonormal eigenspace.

The Laplacian matrix $\mathbf{L}$ is a square matrix which characterizes the pairwise information (also called affinity in the literature) between any two vertices on the mesh $\mathcal{O}$, *e.g.*, $L_{i,j}$ reflecting the weight between the i^{th} vertex and the j^{th} vertex. The Laplacian operator has a strong physical meaning which is equivalent to a second order differential operator in physics in the study of wave propagation, heat diffusion, electrostatics and fluid mechanics. Because the matrix encodes the one ring neighbourhood information of the mesh, it can also be considered as a convolutional kernel from the signal processing perspective. According to the different requirements, the Laplacian matrix can be used to simulate different continuous operators. Not only the connectivity information can be considered but also the geometric information can be embedded in the matrix as well.

Because the Laplacian matrix $\mathbf{L}$ is square and positive semi-definite, it means the eigen-decomposition produces a set of non-negative eigenvalues and a set of orthonormal eigenvectors. Chung [26] stated that Laplacian eigenvalues are closely related to almost all major graph invariants. In other words, the eigenvalues contain most of the information about the characteristics of the shape. Inspired by such properties, the graph spectra are used for shape matching and retrieval in computer vision [21, 81] and for indexing [64, 107]. On the other hand, eigenvectors provide a more refined shape characterization [146]. Furthermore, the eigenvectors have much wider applications including object segmentation [85,111,136], clustering [10,15,127, 128], parametrization [53, 148] and shape matching [21, 48, 108, 114].

Instead of using the eigenvalues and eigenvectors directly, the eigenvectors can be used similarly to Fourier descriptors. And the spectral coefficients can be obtained

by projecting the mesh geometry, *i.e.* the vertex coordinates, onto the orthonormal eigenspace defined by the eigenvectors. The coefficients also contains the energy and global information of the mesh global information. They can be used in various ways such as geometry compression [68], mesh watermarking [89, 90, 98, 100] and as Fourier descriptors [129].

In the following of this section, the watermarking methods are classified according to the type of basis functions used in the spectral analysis. Methods based on Combinatorial Laplacian are firstly introduced. Most of the spectral 3D watermarking methods belong to this branch. Methods based on manifold harmonics is followed and lastly the other types of spectral methods.

2.1.1 Combinatorial Laplacian methods

A combinatorial Laplacian is a matrix operator that solely depends on the connectivity of the mesh. It treats the pairwise relation as a binary delta function, *i.e.* if $\mathbf{v}_i$ is connected with $\mathbf{v}_j$, the corresponding entry is 1 otherwise, is 0. The idea was firstly introduced by Taubin [118] to approximate low pass filters. Kaini *et al* [68] compress the mesh geometry making use of the eigenprojections. Zhang [145] studies several variants of combinatorial Laplacian and their properties for spectral geometry processing and JPEG-like mesh compression.

Most of the spectral watermarking methods so far tend to embed the message in the spectral coefficients called eigenprojections in some papers. This is because the basis functions, *i.e.* eigenvectors, of the combinatorial Laplacian operator are stable and insensitive to the geometry changes since only the connectivity is considered in the matrix. Thus, after watermarking, the connectivity is not changed so the watermarked coefficients can always be detected. Some of the watermarking methods [2, 98] tend to remesh the mesh object ensuring the connectivity is consistent.

2.1.1.1 Theoretical background

We first briefly review the theoretical background of spectral analysis using the combinatorial Laplacian based on the work proposed by Karni *et al* [68]. Given a

mesh object $\mathcal{O}$ containing N vertices, the Laplacian matrix of dimension $N \times N$ is built according to its connectivity as follows:

$$L_{i,j} = \begin{cases} |\mathcal{N}_{\mathbf{v}_i}| & \text{if } i = j \\ -1 & \text{if } i \neq j \text{ and } \mathbf{v}_i \text{ adjacent to } \mathbf{v}_j \\ 0 & \text{otherwise} \end{cases} \tag{2.1}$$

where $|\mathcal{N}_{\mathbf{v}_i}|$ represents the valence of the vertex $\mathbf{v}_i$, *i.e.* the number of its neighbours directly connected to it. Then, the Laplacian matrix is eigen-decomposed as:

$$\mathbf{L} = \mathbf{q}^T \mathbf{\Omega} \mathbf{q} \tag{2.2}$$

where $\mathbf{\Omega}$ is the diagonal matrix containing the eigenvalues and $\mathbf{q}$ is the matrix consisting of the eigenvectors. The eigenvector matrix $\mathbf{q}$ is sorted in the ascending order according to the magnitude of its corresponding eigenvalues in the diagonal matrix $\mathbf{\Omega}$. While the eigenvalues in $\mathbf{\Omega}$ are considered as frequencies, $\mathbf{q}$ constitutes an orthonormal basis of the mesh $\mathcal{O}$. The spectral coefficients are calculated by projecting the vertex coordinates on the basis functions defined by the eigenvectors $\mathbf{q}$:

$$\mathbf{C} = \mathbf{q}\mathbf{V} \tag{2.3}$$

where $\mathbf{V}$ is the matrix containing the geometry of the vertex coordinates. The spectral coefficients of low frequencies, *i.e.* the coefficients correspond to the small eigenvalues in $\mathbf{\Omega}$, reflects the general shape or the large scale information of the mesh. In contrast, the high frequency coefficients indicate the details or the small scale information of the mesh. Figure 2.1 shows a set of spectral coefficients. 90% of the mesh energy is contained in the low frequency, while the energy in the high frequency is much lower. To reverse the transformation process, the geometry can be recovered as:

$$\mathbf{V} = \mathbf{q}^T \mathbf{C} \tag{2.4}$$

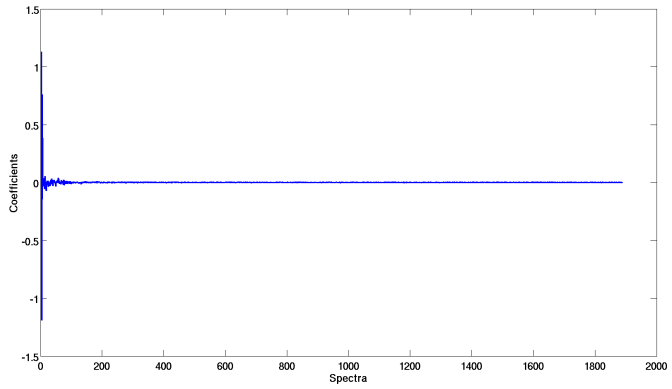


Figure 2.1: A plot of spectral coefficients.

2.1.1.2 Non-blind methods

Ohbuchi *et al* proposed a non-blind method in 2001 in [100] based on Karni's analysis from [68]. This is the first 3D watermarking method based on the spectral domain. The method applies the spectral analysis employing the basis functions of the combinatorial Laplacian. The message is embedded by slightly modifying the low frequency and medium frequency coefficients. In the detection stage, both the original object and the watermarked object need to be spectrally decomposed. The embedded information is retrieved by comparing the difference of the spectral coefficients between the original and the watermarked ones. However, any modification to the combinatorial Laplacian would result in different eigenvectors. So the algorithm is sensitive to any of the attacks that modifies the connectivity of the mesh. Furthermore, the method is computational expensive. The matrix eigen-decomposition requires $O(N^3)$ complexity. Thus, although any mesh can be spectral decomposed theoretically, it is not feasible to do so in a large mesh in practice.

In 2002, Ohbuchi *et al* extended their previous work in [98] in three directions. The mesh size was reduced by splitting it into several patches. Each patch is used to carry a set of bits. A more efficient numerical method called Arnoldi [52] is employed to eigen-decompose the Laplacian matrix. The Arnoldi method can calculate the leading spectral coefficients as required, instead of calculating the full set of the eigenvectors. Finally, the 3D object is remeshed before detection in order to recover the original connectivity such that the Laplacian Matrix is identical to the original

one. The method proposed in 2002 is resistant to the connectivity alteration attacks like mesh simplification and cropping because the connectivity is enforced to be the same in the detection stage. This method is computationally more efficient as not only the matrix size is reduced but also the numerical routine for eigen-decomposition is improved. In 2004, Ohbuchi *et al* and Cotting *et al* proposed two similar methods in [36, 99] to extend the ideas from the other papers to the point sampled object. Although the object is in point cloud format, the intrinsic connectivity is built before constructing the Laplacian Matrix.

Lavoué *et al* proposed a similar method as Ohbuchi's to watermark subdivision surfaces [77, 79]. The message is embedded in the control mesh (called also base mesh) of the subdivided mesh. Control mesh is the lower-resolution version of the original mesh after the wavelet decomposition. In the message retrieval, the control mesh synchronization need to be done on the attacked model so as to detect the message. There are two improvement of Lavoué's method over Ohbuchi's methods. Firstly, Lavoué proposed a so-called Low Frequency Favoring (LFF) modulation scheme. The full range of spectrum can be used for embedding by employing the LFF scheme. The LFF takes the magnitude of the spectral coefficients into account and adaptively embed the watermark. For the high frequency coefficients, *i.e.* a small numerical value, the embedding strength is adjusted to a smaller value. Moreover, the capacity of the watermark and the imperceptibility is optimized using error correcting codes. A large message can be encoded using a relatively small number of bits. The method claimed a 20% improvement of the watermark robustness over Ohbuchi's method [98].

All these methods are non-blind and the bit carriers are the low frequency and medium frequency coefficients. The main strength of these methods is the relatively high robustness. Nevertheless, the premises is made that the original object must be present in the message retrieval stage. There are three disadvantages. Firstly, the original object is required to recover the original connectivity. This involves extra steps and computational cost. Secondly, the computational cost is higher than spatial domain methods in general. Thirdly, it is hard to control the distortion. Although there are embedding strength parameters in order to control the visual distortion, there is no explicit relation between the coefficients and the vertex co-

ordinates. Therefore, decreasing the embedding strength is the only way to reduce the visual distortion. Furthermore, the distortion is large because the change of low frequency will change the general shape of the object.

2.1.1.3 Blind methods

Cayre and Alface *et al* proposed a blind algorithm [23] based on the spectral domain in 2003. A mesh object can be considered as a three dimensional signal, *i.e.* (v_x, v_y, v_z), we can have the corresponding spectral coefficient triplet (C_x, C_y, C_z). Every triplet is considered as an embedding primitive. The triplet is sorted in the ascending order and the maximum $C_{max} = max(C_x, C_y, C_z)$ and minimum value $C_{min} = min(C_x, C_y, C_z)$ are regarded as the modulation range. The mean value $Mean = (C_{max} + C_{min})/2$ is used to distinguish the bits 1 and 0 intervals. When embedding a 1 bit, the medium coefficient is moved into the interval of values corresponding to the bit 1 and vice versa. Figure 2.2 shows an example of the triplet embedding. The embedding message is inserted repetitively into the low and medium frequency to ensure the robustness. The method is the first blind algorithm based on the spectral domain, but its robustness is very limited.

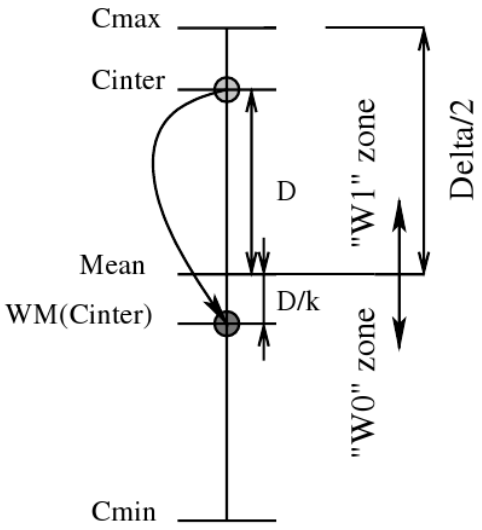


Figure 2.2: C_{inter} is moved into the 1 bit interval when embedding 1 bit. Figure is taken from [23].

Alface *et al* in 2005 [2] proposed to segment the 3D object into patches for reducing the embedding complexity while the core embedding method is still the same as

Cayre's method [23]. Firstly, the feature points are automatically selected through a multi-scale estimation of the curvature tensor field. Then, the algorithm proceeds by partitioning the mesh shape using a geodesic Delaunay triangulation of the detected feature points. Each of these geodesic triangle patches is then parametrized and remeshed by a subdivision strategy to obtain a robust base mesh. The remeshed patches are watermarked in the spectral domain and original mesh points are finally projected on the corresponding watermarked patches. The automatic feature point detection and the patch generation are the main contribution of Alface's method. The core watermarking process is basically identical with Cayre's method. Thus, it suffers from the low robustness problem as well.

All these methods are blind The main embedding idea is to encode information into the coefficient triplet. A lot of efforts are made on the preprocessing steps such as the robust feature points detection and the patch generation. And the spectral decomposition is made on the remeshed and parametrized model. The robustness of those methods somehow depend on the robustness of the pre-processing stages more than the embedding itself. Same as in Ohbuchi's methods mentioned in the previous section, Alface's methods embed the message in the low and medium frequencies. In other words, it embeds the message into the "shape" of the object. The algorithms show certain robustness. And lastly, the methods strongly rely on the pre-processing of the robust feature points, patch generation, parametrization and remeshing.

2.1.2 Manifold harmonics

Although the combinatorial Laplacian has the perfect reversibility and it is simple to implement, the lack of the geometry information makes it inadequate to describe the feature of an object. There is another kind of discrete Laplacian which deals with the geometry properties of the mesh, called Manifold Harmonics, proposed by Vallet [126]. Its transformation is called Manifold Harmonics Transform (MHT).

The Manifold Harmonics injects the geometry information by calculating the cotangent (cotan) weights of the one ring neighbourhood. The weight between $\mathbf{v}_i$ and $\mathbf{v}_j$ is measured by the cotan angle opposite to the edge formed by the two vertices [41, 93]. The cotan weight derived from Finite Element Modeling [135] has

been proved a close relationship with the surface curvature [93]. They converge to the continuous Laplacian under certain conditions as explained in [11, 60]. Nonetheless, the cotan weights are calculated by the dual cell area of each vertex, which is non-symmetric. Thus, the cotan weights can not be used for the spectral analysis directly. Lévy tried empirical symmetrization in [82]. Vallet *et al* clarify these issues based on a rigorous Discrete Exterior Calculus (DEC) [55] formulation and recover symmetry by expressing the operator in a proper basis [126]. The symmetry property ensures its eigenfunctions are both geometry aware and orthogonal as well.

2.1.2.1 Theory background

In this section, we clarify the theoretical issues of the Manifold Harmonic Transform. Similar to the Laplace operator in Euclidean space, the Laplace-Beltrami operator Δ is defined as the divergence of the gradient for functions defined over a manifold $\mathcal{O}$ with its metric tensor. The eigenfunction and the eigenvalue pair (H^k, λ_k) of Δ on manifold $\mathcal{O}$ satisfy:

$$-\Delta H^k = \lambda_k H^k \tag{2.5}$$

The above eigen-problem is then discretized and simplified within the finite element modeling framework as the following matrix equation:

$$-Q\mathbf{h}^k = \lambda_k D\mathbf{h}^k \tag{2.6}$$

where $\mathbf{h}^k = [H_1^k, H_2^k, \ldots, H_n^k]^T$, the $N \times N$ matrix D is diagonal and called lumped mass matrix as:

$$D_{i,i} = \left(\sum_{t \in N_{\mathbf{F}_i}} |t| \right) /3 \tag{2.7}$$

where $N_{\mathbf{F}_i}$ is the number of neighbouring faces of vertex $\mathbf{v}_i$. t is a neighbour of vertex $\mathbf{v}_i$. $|t|$ gives the area of the triangle. The matrix Q called stiffness matrix is also of size $N \times N$:

$$\begin{cases} Q_{i,j} = \left(cot(\alpha_{i,j}) + cot(\beta_{i,j})\right) /2 \\ Q_{i,i} = -\sum_j Q_{i,j} \end{cases} \tag{2.8}$$

where $\alpha_{i,j}$ and $\beta_{i,j}$ are the two angles opposite to the edge V_iV_j. The Manifold Harmonics Basis can be calculated by eigen-decomposing the matrix Q in equa-

tion (2.6). The frequencies are represented by the corresponding eigenvalues. Let us define vector $\mathbf{x} = (x_1, \ldots, x_N)$ (respective $\mathbf{y}$ and $\mathbf{z}$) containing the x coordinates of the mesh. With the Manifold Harmonics Basis, the k^{th} spectral coefficient can be calculated as:

$$c_k^x =< \mathbf{x}, \mathbf{h}^k >= \sum_{i=1}^{n} x_i D_{i,i} H_i^k \tag{2.9}$$

Thus, the amplitude of the spectral coefficients is defined as:

$$c_k = \sqrt{(c_k^x)^2 + (c_k^y)^2 + (c_k^z)^2} \tag{2.10}$$

The object can be exactly reconstructed by using the inverse manifold harmonics transform. For coordinates $\mathbf{x}$ (resp. $\mathbf{y}$, $\mathbf{z}$), we have

$$x_i = \sum_{k=1}^{n} c_k^x H_i^k \tag{2.11}$$

With the geometry information embedded in the operator, the spectrum obtained from the MHT nicely captures shape characteristics of the object. However, on the other hand, the side effect is that when the geometry of the mesh is changed, *e.g.* watermarked, the approximation matrix $\mathbf{Q}$ will be changed. Thus, if we apply the MHT again on the modified mesh, we can no longer retrieve the watermarked coefficients again. The causality problem is the major obstacle of using the MHT to design a watermarking method. People tend to use the iteration methods to recheck the coefficients to ensure a successful embedding [86].

Another major contribution of Vallet's work is a band-by-band spectrum computation algorithm and an out-of-core implementation that can compute thousands of eigenvectors for meshes with up to a million vertices. These make the spectral analysis directly usable in practice on a large mesh object, besides its common use as a theoretical tool.

2.1.2.2 Blind methods

Since the Manifold Harmonics Basis incorporates more geometry information of the mesh object, it captures more shape information rather than when considering topol-

ogy only. The spectrum obtained from the MHT is very stable and consistent for the other object representations. It means that the attacks like mesh simplification, resampling and remeshing, which do not alter the shape of the object, will not affect the spectrum very much. Because this feature of the MHT, it becomes a popular transformation technique to devise robust watermarking schemes. In this section, I will briefly introduce two recent robust and blind algorithm based on the manifold harmonics transform proposed by Vallet *et al* [126].

Liu *et al* [86] proposed a robust and blind algorithm based on the manifold harmonics in 2008. The method takes the medium frequency coefficients as the embedding domain. The authors experimentally show that the medium frequency changes affect the spectrum very little and can be accepted for watermarking purpose. Every ten coefficients are grouped as a embedding primitive used to carry one bit of message. Two embedding methods were proposed, one is called progressive embedding and the other is non-progressive embedding. The progress embedding picks one coefficient magnitude from the primitive and calculate the mean value of the other nine coefficient magnitudes. The selected embedding candidate magnitude is moved more than the mean value for embedding a 1 bit and less than the mean value for embedding a 0 bit. In order to overcome the causality problem, *i.e.* the watermarked coefficients can not be exactly recovered from the watermarked model, the method iteratively check the coefficients until it satisfies the embedding condition. While the assumption is made that a small change on the medium range spectrum will not affect the shape of the spectral coefficients significantly. The experiments show certain robustness against various attacks when 5 bits of message are embedded.

Wang *et al* [133] proposed another robust and blind algorithm based on the MHT in 2009. Unlike Liu's method where only 5 bits of message a embedded, Wang's method is able to carry 16 bits of message. The scalar Costa scheme [46] which is a quantization algorithm is used to modulate the low frequency coefficients to embed the message. The unique code-book generated from the Costa scheme ensures a good security of the algorithm. The spectral coefficients are repetitively embedded and iteratively checked to avoid the causality problem. The author argued that although the low frequency changes introduce a large numerical error on the

mesh object, human eyes are not sensitive with respect to distortion of the large scale changes [78]. The method was compared with Liu's method [86] and Cho's method [31] and shows a good result on both visual quality and robustness.

2.1.3 Other transformed methods

Other transformed domain methods includes Singular Spectrum Analysis (SSA) [94], Discrete Cosine Transform (DCT) [65], spherical parametrization [83], Oblate Spheroidal Harmonics [75].

The Singular Spectrum Analysis is assuming a time series of the object geometry. The time series is virtually the vertex order in the object file. Murotani *et al* [94] proposed a non-blind algorithm based on the SSA in 2003. The spectrum is then computed using the SSA for the trajectory matrix derived from the vertex series and used for watermarking. The original object is required in the detection stage to retrieve the watermark. The experiments show the algorithm is robust against the similarity transforms and the additive noise. The algorithm is a spectral domain method but obviously the assumption of the time series in SSA is not robust and the watermark can be easily destroyed. Any attack that modifies the vertex order, for instance a vertex reordering, will fail the algorithm.

Jeon *et al* [65] applied the Discrete Cosine Transform (DCT) to devise a 3D watermarking algorithm. The algorithm generates a set of triangle strips according to a secret key. The strips are then transformed into the spectral domain using DCT. The mid-frequency band of AC coefficients are used to carry the watermark in order to balance the trade-off between the robustness and the imperceptibility. The authors claim three advantages of using triangle strips. 1. The user who doesn't know the starting face for creating triangle strips can not distinguish a watermark pattern. 2. The triangle strips also have the property of mesh partition, it can be considered as a subset of the mesh object. 3. Finally, inserting the message into multiple strips strengthen the robustness. As proved experimentally, the method is rather robust against the additive noise attack and geometry compression. However, it is not resistant against any attacks that alter the connectivity of the mesh.

Li *et al* [83] proposed a non-blind method based on the spherical parametriza-

tion in 2004. The geometry of a 3D object is transformed into spherical signals using a global spherical parametrization and an evenly sampling scheme. Spherical harmonic transformation is then applied to generate frequency coefficients for embedding watermarks. The algorithm shows a good robustness against the additive noise attack.

In 2005, Wu *et al* [139] argued that the Combinatorial Laplacian spectral method does not encode any geometric information in the discrete operator. In addition, the inverse of a large matrix is computationally unfeasible. Wu *et al* introduced a new set of geometry dependent orthonormal basis functions derived from the Radial Basis Functions. By using the scheme, the main features of the mesh object can be recovered by using just a few spectral coefficients. The advantage of the new basis functions is that its computation is significant faster than the Laplacian based functions. However, the same as the other non-blind spectral methods, the message detection relies on the mesh registration, resampling and remeshing. As a consequence, the robustness benefits from those extra steps.

In 2009, Konstantinides *et al* [75] proposed a blind and robust method based on the Oblate Spheroidal Harmonics. The transform is based on the use of one of the many variants of oblate spheroidal harmonics; namely the Jacobi ellipsoidal coordinates [54, 120]. The algorithm realigns the mesh object by translating the object onto the mass centre, uniformly normalization and PCA rotation. However, the robustness of these traditional alignment methods can be severely affected by attacks. Thus, a smoothing scheme is proposed prior to the alignment. This is based on the observation that attacks like noise, resampling, remeshing and mesh simplification tends to alter the high frequency properties, while the smoothing tends to eliminate the high-frequency attributes, the smoothed versions of the attacked mesh and the intact one converge to roughly the same one. Patches are then generated on the smoothed surface while the patch centre is established as the intersection between an randomly-generated ray and the object surface. The radius of the patch is defined according to the geodesic distance. While the patch is generated on the smoothed surface, the points are sampled on the original object by projection from the smoothed version to the original one. When the preprocessing steps are all completed, the patches are spectral transformed and the watermark is embedded

in the spectral coefficients. The algorithm is compared with the state of the art algorithm proposed by Cho *et al* [31] and the results show a better robustness and better visual quality. However, the algorithm involves too many preprocessing steps like reorientation, patch generation and sampling etc. Moreover, the capacity of the algorithm is quite low and it is tested for embedding only 7 bits of message is tested in the experiments.

Ai *et al* [1] introduced a method that firstly find out the feature points from the rapid changing regions. The mesh is uniquely segmented into Voronoi patches using those feature points. Each patch is used to generate a range image. A Discrete Cosine Transform is then applied on the range image and the bit message is inserted into the high frequency of the image. The algorithm is robust against various attacks including mesh simplification, additive noise and cropping etc. This method directly applies 2D image watermark techniques to the 3D methods by generating the range image of the mesh object.

2.2 Multi-resolution methods

2.2.1 Regular wavelet decomposition

Except the classic Fourier-like analysis, another traditional approach used in signal processing area is the multiresolution analysis. The basic idea behind multiresolution analysis is to decompose a complicated function into a "simpler" low resolution part, together with a collection of perturbations, called wavelet coefficients [87]. While in the case of a 3D mesh object, the original 3D mesh itself is considered as a function. The object is analyzed using the so-called lazy wavelet transform [113]. In the transform, the object is filtered with a wavelet function. A base mesh is then generated *i.e.* the base mesh is the analogy of the low-resolution function and it should be a good approximation of the original denser one. The information that is lossy in the base mesh is stored in the wavelet coefficients. Thereafter, the 3D object is iteratively analyzed using the different scale of basis functions. The functions with different scales are orthogonal. The object can be decomposed into different level of details as shown in Figure 2.3. The scheme proposed by Lounsbery *et al* [87] requires

that the mesh must fit a 4-to-1 subdivision connectivity scheme, *i.e.* a vertex can only connect with six neighbours. Because of the restricted requirement of the mesh, the wavelet transform described by Lounsbery *et al* [87] is also called regular mesh wavelet transform. Any mesh can be easily converted to the regular mesh using the method proposed by Eck *et al* [45].

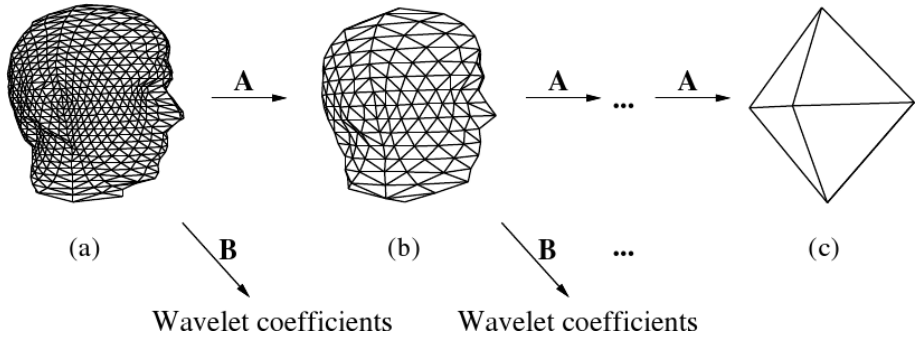


Figure 2.3: Wavelet decomposition [87]

To formulate the wavelet transform in a more rigorous manner, we have:

$$\begin{aligned} \mathbf{V}^j &= \mathbf{A}^j \mathbf{V}^{j+1} \\ \mathbf{W}^j &= \mathbf{B}^j \mathbf{V}^{j+1} \end{aligned} \tag{2.12}$$

assuming that $\mathbf{V}^{j+1}$ denotes the matrix whose row corresponds to the vertex coordinates at the resolution level $j+1$. Then $\mathbf{V}^j$ is the one level lower resolution. $\mathbf{W}^j$ is the wavelet coefficients which is the lossy information from resolution level $j+1$ to j. $\mathbf{A}^j$ and $\mathbf{B}^j$ are called the analysis filters at resolution level j producing the base mesh (base function) and the wavelet (lossy information), respectively. The transform can be reversed by adding the lossy information contained in the wavelet coefficients back to the base mesh as:

$$\mathbf{V}^{j+1} = \mathbf{P}^j \mathbf{V}^j + \mathbf{Q}^j \mathbf{W}^j \tag{2.13}$$

where $\mathbf{P}^j$ and $\mathbf{Q}^j$ are called synthesis filters. An interesting mathematical relation between the synthesis filters and the analysis filters is defined as:

$$\left[\mathbf{P}^j | \mathbf{Q}^j\right] = \left[\frac{\mathbf{A}^j}{\mathbf{B}^j}\right]^{-1} \tag{2.14}$$

Kanai *et al* firstly employed the wavelet framework and developed a non-blind 3D watermarking algorithm in 1998. They argue that the human eye is not sensitive to the small geometric changes in the bumpy areas. While the norm of the wavelet coefficient vector $|\mathbf{w}_i^j|$ indicates the degree of the bumpiness of the small area near around $\mathbf{w}_i^j$ [67]. The larger the norm, the bumpier the area. Thus, only the wavelet coefficient vectors that are larger than a certain user-specified threshold are selected as the watermarking domain. Furthermore, the geometric error is also strictly controlled so that the largest change of the wavelet can not beyond certain level. The vector is modulated to carry the watermark and the message can be recovered by comparing the wavelet of the original mesh.

Uccheddu *et al* [124] proposed a blind algorithm. The watermark is embedded by modulating the norm of the wavelet coefficient vector. The change of the norm is determined by the look up table generated by a secret key.

Cho *et al* [32] proposed a fragile watermarking algorithm using the wavelet transform. And Wang *et al* [131, 132] proposed a hierarchical watermarking algorithm using the wavelet. Three watermarks can be inserted in different appropriate resolution levels obtained by the wavelet transform. The robust watermark is stored in the lowest resolution by modifying the norms of the wavelet coefficient vectors. The fragile watermark is embedded in the high resolution level. And the high-capacity watermark is inserted in one or several intermediate levels.

In conclusion, the watermarking methods based on the regular wavelet transform [87] such as proposed by Kanai *et al*, Uccheddu *et al*, Cho *et al* and Wang *et al*, are rather similar. The basic idea is to modulate the wavelet coefficient norms to embed the message. The weight of the modulation is controlled using different quantization methods or lookup tables. There are four main advantages of using the wavelet transform.

1. As the norm of wavelet vector implicitly characterize the bumpiness of the local surface, and human eyes are not sensitive to the changes in the bumpy areas, it is easy to define the area that is more suitable for watermarking than the others.

2. The watermarks can be embedded in different resolution levels. Furthermore,

as the low resolution represents the low frequency and high resolution contains more about the high frequency information, various resolution levels can be watermarked for different purpose as Wang *et al* did in [131, 132].

3. Not only the wavelet coefficient vector can be watermarked, but also the base mesh. This gives a broader range of the embedding domain.

4. The lazy wavelet transform enables researchers to define a clear geometric relation between the surface distortion and the upper bound of the modification of each wavelet coefficient vector.

On the contrary, because the strict limitation of the regular mesh wavelet decomposition, there are two obvious disadvantages of this class of methods:

1. The transform works only on the restricted topological class of the mesh, *i.e.* the mesh must be in a 4-to-1 subdivision connectivity schemes. Every vertex can only have six neighbours.

2. This class of methods are not robust against any connectivity attacks like mesh simplification, cropping and remeshing etc.

2.2.2 Other multi-resolution methods

There are other multi-resolution analysis methods on mesh with arbitrary connectivities rather than the restricted 4-to-1 connectivity. Progressive mesh [61] was proposed by Hoppe. The method employs the restricted edge collapse operation which is chosen deterministically with the goal of preserving the surface of the original mesh. This operator is applied iteratively until it reaches a coarse base mesh and it is accompanied with a sequence of vertex split operations. Praun *et al* developed a method based on the progressive mesh [104]. Firstly a number of basis functions are constructed. The boundary of the basis functions are defined in the coarse base mesh of the progressive representation. Then, the vertex within each region is displaced according to a interpolation function. In the message extraction, the observed mesh should be re-synchronized and resampled with the original mesh.

Valette *et al* extends the wavelet decomposition to the irregular mesh in 2004 [125]. In this paper, a set of well-designed wavelet codebook is constructed in order

to match an arbitrary local connectivity scheme. Instead of contract four triangles into one as the regular wavelet transform, the irregular transform is able to deal with 2-to-1, 3-to-1 and 4-to-1. The extension work is also used for watermarking. Kim *et al* [70] propose a method based on Valette's irregular wavelet. The idea is not surprisingly the same as the regular wavelet methods, *i.e.* the norm of the wavelet coefficient vector is modulated to embed the message. However, because the irregularity of the connectivity of the mesh, Valette's wavelet method in fact needs a starting triangle to proceed the decomposition process. Thus, the wavelet decomposition will not produce the same base mesh or the wavelet sequence if the connectivity is changed. Therefore, the algorithm is not robust against the mesh simplification attacks.

2.3 Spatial domain algorithms

Spatial domain methods consist of embedding the message in the geometry or the connectivity of the mesh directly. Thus, there are three characteristics of the spatial domain methods. First of all, it is easy to apply constraints on the mesh and the constraint can be easily recovered and detected blindly. Secondly, because the geometry and connectivity define the appearance of the surface, it enables the user to explicitly control the watermarking distortion on the object surface. Finally, spatial domain method does not have the extra transformation steps, they are much more computationally efficient than the transformed domain methods. These three features determines that the spatial domain is more suitable for blind or fragile watermarking as well as for steganography applications.

From the purpose or the application point of view, the 3D methods can be classified into three sets: robust watermarking, steganography and fragile watermarking. Almost all transform domain methods are robust watermarking algorithms with a few exceptions of the wavelet methods. In fact, although the wavelet transform analyze the object using a set of orthogonal basis functions, the manipulations are directly on the geometry. On the other hand, spatial domain is used in all the three classes of algorithms. In my research, all my methods are blind and robust watermarking algorithms. Therefore, in this literature review of the spatial do-

main methods, I will mainly focus on the robust methods in the spatial domain. Steganography and the fragile watermarking will be briefly reviewed for completion.

2.3.1 Robust methods

Benedens *et al* in 1999 proposed one of the first robust 3D watermarking methods based on the spatial domain in [16]. This method groups the vertex normals as the watermarking bins and each bin is used to carry one bit of message. The message is embedded by carefully modifying the normal distribution of each bin. The experiments show that the algorithm has a good performance against the mesh simplification attack. Because the mesh simplification attack tends to preserve the surface and thus the vertex normals are not likely to be changed a lot. While it is more problematic in the noise attack which randomly modifies the geometry of the surface.

Harte and Bors published a paper [57] in 2002 and proposed to embed the message using the local moments. The work was completed by Bors in 2006 [19]. Two schemes were proposed. One is to use the ellipsoid to model the local moments, while the other places two parallel planes locally according to the geometry of a vertex neighbourhood. The message is encoded by examining the position of the candidate vertex with respect to the ellipsoid or the two parallel planes. The sequence order is checked after the embedding to make sure the message can be recovered afterwards. The algorithm is robust against the additive noise attack and smoothing up to a certain level.

Yu *et al* proposed a spatial domain method based on the distance from vertex to the object centre [141] in 2003. The vertices are firstly scrambled and divided into sections using a secret key. The distances of vertices within each section are then modified in order to embed one bit of message. The magnitude of the distance change is adapted with respect to its original length to control the surface distortion. In the detection stage, the observed model is synchronized and re-sampled with the original model. The message is retrieved by comparing the difference between the original length and the watermark length. The algorithm is robust against various attacks including mesh simplification, additive noise and cropping etc. However,

this method requires the original object to recover the message.

Zafeiriou *et al* proposed two spatial domain methods in 2005 [142]. The first method uses individual vertex as the embedding primitive and it is robust against the similarity transformation. The second one converts the vertices from the Euclidean coordinate system (x, y, z) to the spherical coordinate system (ρ, θ, ϕ) first. The vertices are clustered into groups according to their θ values. The assumption is made on the distribution of the ρ component to be a Gaussian distribution within each cluster. The message is embedded by modifying left variance or right variance of the Gaussian distribution. The second algorithm shows a good robustness and visual quality. Zafeiriou's method is probably the first method in the spatial domain that utilize the geometric statistical feature of the object to embed the watermark.

Cho *et al* in 2007 proposed a similar statistical method [31] combining the ideas of Yu *et al* and Zafeiriou *et al*. In this work, the vertices are firstly clustered into groups according to the distance from the vertex to the object centre *i.e.* ρ component of the (ρ, θ, ϕ) spherical coordinate system. The observation tells that the distribution of the ρ component is uniform within each bin. Two histogram mapping functions are introduced to modify the mean value and variance value of the distribution respectively as shown in Figure 2.4. The histogram mapping functions ensure the statistical condition of the distribution is satisfied while the Euclidean movement of the vertex is minimum. The method proposed by Cho *et al* is probably the most robust 3D watermarking algorithm that does not require the original object to retrieve the watermark.

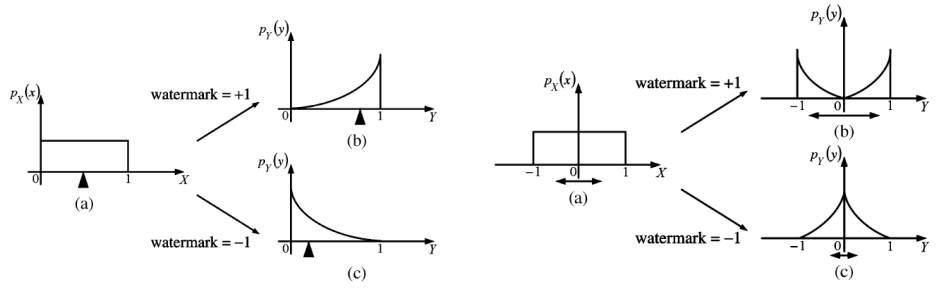


Figure 2.4: (a) change the mean value of the distribution (b) change the variance of the distribution. Taken from [31].

Alface *et al* in 2007 [3] applied Cho's method locally and proposed a framework to withstand the cropping attack. A number of robust feature points are automatically detected and used as the source points. From those source points, geodesic waves are propagated and the boundary of patches are defined as the intersections between those geodesic waves. Each patch is used as a single message carrier. Alface *et al* argue that the message can still be recovered as long as there is at least one intact patch existing in the attacked mesh model. However, when the cropping attack removes one of the feature points, the propagation intersection will be different. Thus, neither the patch is recovered nor the watermark. Thus, the method is not in fact totally solve the cropping problem. Furthermore, the robust feature point proposed in their papers [2, 3] must calculate the geodesic maps for every single vertex. So it is very computationally expensive and the complexity is approximately $O(N^2 \log N)$. So the method is of limited use in practice.

Hu *et al* [62] in 2009 extends Cho's work by minimizing the mean square error between the original mesh and the watermarked mesh under several constraints. The assumption is made that the distortion is the same, but the robustness is improved. In this method, the object centre is enforced to be the same as the original one. The watermarked norms in each bin are carefully calculated to make sure a higher robustness. The main improvement is made that the vertex is guaranteed to be within the bin after watermarking and the gravity centre is not changed.

All the 3D watermark methods I developed are compared with Cho's method. This is because Cho's method achieves the almost best robustness over all the other methods in the literature. Furthermore, the two methods described in Chapter 5 are extension and generalization of Cho's method. These two methods preserve the robustness of Cho's method, while the surface distortion is significantly reduced.

2.3.2 Other spatial domain methods

Except the robust watermarking algorithm, steganography and fragile watermarking are the other two kinds of algorithms in the watermarking family. As we have reviewed in the previous sections, robust watermarking is a technique that aims to detect the embedded information even when the stego medium suffered from a cer-

tain level of attacks. It is designed for copyright protection purpose. Steganography and fragile watermarking are designed with different motivations in mind.

Steganography is a data transmission and storage technique. In this scenario, the capacity is the most important criteria to evaluate a steganography algorithm but not the robustness. This branch of algorithms usually use every vertex as a embedding primitive. The sequence of the embedded message can be determined by the connectivity [17]. The capacity can be increased by quantization [23], subdivision [122], angles [30] and multilayer embedding [24].

Fragile watermarking, on the contrary to the robust watermarking, is designed that the watermark should disappear when any attacks happens to the stego medium. Therefore, it is used for authentication. In the message retrieval, if the information is detected completely and correctly, then the object is intact, otherwise the mismatching message tells people the object is attacked or modified unintended. A good fragile watermarking should be able to tell where the object is touched and what kind of attacks it is suffered. The basic idea of fragile watermarking is to slightly move the vertex to a predefined relationship. The relationship can be defined with respect to its neighbours [33, 134, 140] or the centre of the object [138]. There are two common problems that are frequently encountered in the fragile watermarking: the causality problem and the convergence problem. The causality problem raised when the neighbouring relationship of a former processed vertex is influenced by the perturbing of its latter processed neighbouring vertices. The convergence problem means that the original model has been severely distorted before some vertices reach the predefined relationship. The two problems should be carefully handled when designing a fragile watermarking method.

2.4 Robust 3D watermarking assessment

Figure 1.1 illustrates the three most important aspects in the robust 3D watermarking algorithm: distortion, robustness and the capacity. For a robust and blind watermarking algorithm used for the purpose of copyright protection, most methods accept that the payload of the embedded watermark is 64 bits. Thus, most of the evaluation work are focused on the other two parts *i.e.* distortion and robustness. In

this section, I will present the assessment approaches that are used in the literature.

2.4.1 Distortion evaluation

Evaluating the similarity of two 3D objects has been widely studied in the research of the mesh simplifications. In general, there are Six classes of methods to measure the error between the original mesh object and the target object.

1. Hausdorff distance comparison
2. Volume based measurement
3. Energy minimization measurement
4. Curvature based measurement
5. The projection image comparison
6. Human perceptual distance.

The Hausdorff distance is widely used to measure the similarity between two sets of data including image comparison [63] and 3D comparison [12, 34, 56, 66, 74]. The symmetric Hausdorff distance between an original mesh object $\mathcal{O}$ and a processed object $\hat{\mathcal{O}}$ is defined as:

$$H(\mathcal{O}, \hat{\mathcal{O}}) = max(max(min(d(\mathbf{v}, \hat{\mathbf{v}}))), max(min(d(\hat{\mathbf{v}}, \mathbf{v}))), \quad \forall \mathbf{v} \in \mathcal{O}, \quad \forall \hat{\mathbf{v}} \in \hat{\mathcal{O}} \tag{2.15}$$

The most popular implementation called Metro which is a Hausdorff distance measurement was done by Cignoni *et al* [34]. It was originally designed to measure the similarity between the simplified object and the original object. Now, researchers in 3D watermarking community also prefer to use the Metro tool to assess the distortion introduced by the watermark embedding. As the Hausdorff distance is not symmetric, two distances are evaluated: forward $E_f(\mathcal{O}, \hat{\mathcal{O}})$ and backward $E_b(\hat{\mathcal{O}}, \mathcal{O})$ root mean square errors between $\mathcal{O}$ and $\hat{\mathcal{O}}$. The maximum root mean square (MRMS) value is then used as the distortion measure:

$$E(\mathcal{O}, \hat{\mathcal{O}}) = \max\{E_f(\mathcal{O}, \hat{\mathcal{O}}), E_b(\hat{\mathcal{O}}, \mathcal{O})\} \tag{2.16}$$

where:

$$E_f(\mathcal{O}, \hat{\mathcal{O}}) = \frac{\sum_{\mathbf{v}\in\mathcal{O}} \min_{\hat{\mathbf{v}}\in\hat{\mathcal{O}}} \|\mathbf{v} - \hat{\mathbf{v}}\|}{|\mathcal{O}|} \tag{2.17}$$

$$E_b(\hat{\mathcal{O}}, \mathcal{O}) = \frac{\sum_{\hat{\mathbf{v}}\in\hat{\mathcal{O}}} \min_{\mathbf{v}\in\mathcal{O}} \|\hat{\mathbf{v}} - \mathbf{v}\|}{|\hat{\mathcal{O}}|} \tag{2.18}$$

It is generally accepted that the Hausdorff distance is the best approximation measurement to evaluate the similarity between two 3D objects. Therefore, in the rest of our thesis, we will use the MRMS as the objective metric to evaluate the distortion introduced by our watermarking methods.

Volume based measures have first been developed by Alliez *et al* [5]. The error between two meshes $\mathcal{O}$ and $\hat{\mathcal{O}}$ is defined as $V(\mathcal{O}, \hat{\mathcal{O}})$, where V is a Lebesque formula. The main argument is that if $\hat{\mathcal{O}}$ is the best approximation of $\mathcal{O}$ then the volume between both meshes is minimized.

Energy minimization measurements are mostly used in mesh simplification algorithms that use the edge collapse strategy. The energy is nothing but a scalar measuring the difference between the two surfaces. Garland proposed a Quadric Error Metric (QEM) which calculate the sum of squared distances between two surfaces locally [50]. Hoppe used the elastic equations to measure the similarity [61]. In fact, the error functions proposed by Garland and Hoppe have implicit connections with the Hausdorff distortion, Volume and Curvature etc. QEM is by far the most popular metric that is used in mesh simplification due to its accuracy and efficiency.

Curvature Based Distance has been proposed by Kim *et al* [71]. They argue that the visual distortion should be measure according to the human vision which is sensitive to curvature direction changes. The local error is then decomposed into three distinct components, distance, tangential and discrete curvature. Lavoué also proposed a perceptually-driven roughness measurement [78,80] to evaluate the visual similarity between two meshes. This work tries to simulate the subjective human eye behavior rather than calculating the objective distance between two meshes. Similarly, Corsini *et al* [35, 51] proposed a objective roughness assessment.

A natural way to evaluate the similarity between two meshes is human eyes. Therefore, Bian *et al* developed a novel human perceptual distance metric to evaluate

the distortion introduced by mesh processing in [18]. The experiments show that the difference estimates are well correlated with human perception of differences.

The last class of distortion measurement is to compare the similarity between the 2D projection images of 3D meshes [4,84,106]. This kind of measurements have the same inspiration as the Lavoué's perceptually driven roughness measurement [78,80] and they are evaluating the distortion from the human eye perspective.

There is also another distortion measurement worth to mention called the Signal to Noise Ratio (SNR) [142]. The error is calculated as the ratio between the sum of the original signal, *i.e.* the sum of the vertex coordinates and the sum of the noise, *i.e.* the Euclidean distance between vertices. However, as a mesh object is a 2D manifold, Euclidean distance between every pair of vertices does not reflect the real distortion between two surfaces. Therefore, we do not consider this error function as an effective one to evaluate the distortions between two 3D objects.

Finally, as we have mentioned earlier, we will use the MRMS error produced by the Metro tool which is the most popular and efficient Hausdorff distance implementation to measure the surface error between two objects. The Hausdorff distance gives the most accurate and comprehensive description about the similarity of two objects, and it is widely used in computer vision and computer graphics.

2.4.2 Robustness measurement

The other most important property of a robust watermarking method is, no surprisingly, the robustness. The robustness means how much of the probability that we can retrieve the same message as embedded when the mesh object is suffered from some attacks. The most common and simple robustness measurement is the Bit Error Rate (BER) (also called detection ratio), *i.e.* the ratio between the number of bits that are correctly detected and the total number of bits embedded as: Let $\mathcal{M}$ denote the original message to be embedded while $\hat{\mathcal{M}}$ is the detected message. Then BER is defined as:

$$BER = \frac{\sum_{i=1}^{|\mathcal{M}|} \mathcal{M}_i == \hat{\mathcal{M}}_i}{|\mathcal{M}|} \tag{2.19}$$

where $|\cdot|$ gives the cardinality of the set.

The Receiver Operating Characteristic (ROC) or simply the ROC curve is a measurement for a binary classifier system to discriminate the probabilities of false alarm P_{fa} and false rejection P_{fr}. The false alarm means that the detection asserts the object is watermarked when the object is not in fact. Whilst the false rejection describes the probability that the detection tells that the object is not watermarked but it is actually. The operating point where $P_{fa} == P_{fr}$ *i.e.* the Equal Error Rate (EER) can be used as a quantitative estimation of the watermark detection performance.

In the area of 3D watermarking, people generally consider that the BER is sufficient to describe the robustness of a 3D watermarking algorithm. The algorithms described in this thesis are all based on the statistical feature changes instead of a single value modulation, ROC and EER are not necessary. Therefore, we use only the BER (Detection Ratio) to evaluate the robustness of our watermarking algorithms.

2.5 Discussion of the watermarking literature

It is necessary now to briefly summarize the 3D watermarking methods. We here discuss the advantages and disadvantages of the transformed domain methods and the spatial domain methods.

2.5.1 About the transformed domain

There are various of transformation methods proposed in the last decade such as spectral decomposition, multiresolution analysis, DCT and Radial Basis Function etc. Informally speaking, the methods based on the transformed domain try to analogize the techniques from the 2D data to 3D data. Although the transformed domain algorithms are relatively successful in the conventional data type, they do not gain the same success in 3D. The most important reason is that a 3D object is not regular sampled, the connectivity is not regular either.

Different spectral decomposition methods use different set of orthogonal basis functions of the mesh. The combinatorial Laplacian operator interprets purely the topological information of the mesh object. The connection between vertices are

characterized as a binary relation. So the combinatorial Laplacian is the analogy between the graph Laplacian and Fourier transform. It implicitly assumes the uniform sampling of the mesh as pointed-out in [93]. Any geometric changes to the vertices do not affect the basis function. Therefore, it is suitable for watermarking due to its perfect reversibility. However, the major problem is also that the operator solely depends on the connectivity. Then any change modifying the connectivity like mesh simplification and remeshing will destroy the basis functions. As a consequence, the spectra will be different.

Manifold Harmonics Transform (MHT), on the other hand, is another spectral technique that use the discrete Laplace-Beltrami operator to encode the geometric and topological information of the mesh. The relation between vertices are interpreted using a cotan weight which reflects not only the connectivity but also the local surface curvatures. With the geometric information embedded in the operator, the spectrum obtained by MHT is more descriptive and informative. On the other hand, the basis functions are closely related to and can be affected by the geometry due to the construction of the operator. So when watermark embedding modifies the geometry, it may not be possible to retrieve the embedded message due to the causality problem. The researchers using MHT to watermark the object argue that the spectra will not change too much as long as the object is not strongly distorted because the spectra is so descriptive.

There are also other spectral analysis methods, but in general, the ideas are rather similar. The message is embedded by modulating the low or medium frequency coefficients as the argument says the low frequency embedding is less distorted and more robust. The human eye system is not sensitive to the low frequency changes [75]. And the robustness is high because the large scale change of the object is hardly to be modified. Many of the methods require the original object as a reference in the detection stage, which is not practical in most applications.

As a summary, the advantages of the spectral domain watermarking are security and robustness. It is debatable if the distortion is low or not. Firstly, the objective distortion is large, like MRMS value. Secondly, it seems no subjective distortion from a low resolution image (like images in the research paper). But the difference is quite obvious on the computer when you see the real object with rotation and

dragging stuff. Some researchers claim that as a blind method, people will not be able to tell any distortion without the original object. But this is not acceptable from a rigorous scientific perspective.

On the contrary, the computational efficiency is the main disadvantage of the transformed domain methods. A number of techniques were developed for improving the computation including mesh segmentation [89,98] and better numerical methods [42, 76, 116, 126]. Fortunately, watermarking method is supposed to be an offline application rather real time. Secondly, due to the obscure relationship between the geometry and the spectral coefficients, there is lack of mechanism to explicitly control the distortion with respect to the existing distortion evaluation tools [34] etc.

Multiresolution is another kind of transformed domain methods. In this branch, the regular wavelet decomposition is the most popular framework for watermarking 3D object [131,132]. Although the wavelet transform has very rigorous and profound mathematical framework, the actual construction of the wavelet coefficients is rather easy and intuitive. Therefore, the wavelet methods do not have the problem of the high computation cost. Also, the vector norm of the wavelet coefficient explicitly characterize the local surface roughness. It provides a natural way to select the embedding region and strength. Thus, the wavelet method has been used for fragile, steganography and robust watermarking. In contrast, the disadvantage of wavelet method is obvious: it requires the mesh to be in a restricted connectivity scheme. Even for the irregular wavelet method, the connectivity must be identical with the original object in order to retrieve the watermark. So the watermark based on the multiresolution method has limited applications in the real world.

2.5.2 About spatial domain and distortion measurement

So far, the spatial domain has been used in fragile watermarking [140], steganography [23] and robust watermarking algorithms [31]. There are basically two ways to embed the watermark. The first approach we name it as "single embedding" consists of using a single vertex as an embedding primitive and implement some constraint to carry the message. The second one named "statistical embedding"

consists of modifying statistical features. Single embedding is mostly used in the fragile watermarking and steganography because it is not robust to attacks. However, its distortion is low and easy to control. Statistical embedding, on the other hand, consist of using the statistical description as the embedding primitive. So it is generally more robust. The trade-off is the relative high distortion.

In the current study of robust and blind watermarking in spatial domain, the most important problem that I address is the surface distortion. Especially when we modify the statistical features of an object, it is not easy to control the distortion except adjust the parameter of the embedding strength. The distortion can be lower with a lower strength parameter, but also the robustness will be lower in this case

In section 2.4.1, we introduced many existing surface distortion measurements. However, there is no spatial domain watermarking method that explicitly uses any one of the similarity measure to control the distortion. In addition, because the measures are designed for various applications rather than watermarking, they are not very suitable to control the watermarking distortion.

Chapter 3

Spectral Watermarking of 3D meshes

3.1 Introduction

Mesh spectral analysis is derived from the spectral graph theory, signal processing and the kernel principal component analysis. The spectral coefficients are obtained by projecting the mesh geometry onto a set of orthonormal basis functions obtained from the structure of the mesh. The low frequency spectral coefficients reflect the large scale information of the mesh, while the high frequency coefficients indicate the local scale information. The mesh spectral analysis has been used extensively in many area including: shape matching and retrieval [21, 81], indexing [64, 107], segmentation [111, 136], clustering [128], parametrization [53] and geometry compression [68].

Mesh spectral analysis has also been used in 3D mesh watermarking as reviewed in Section 2.1. The computational complexity is higher than that of spatial domain methods, even when using a decomposition scheme [124]. This shortcoming limits the application of spectral methods to large meshes [91]. Thus, the application of most spectral methods would require a mesh segmentation procedure in order to reduce the computational complexity [2, 97].

There are two main advantages of the spectral domain watermarking methods. The robustness of the spectral methods is high especially the non-blind methods.

Each coefficient is reflecting the energy of its corresponding basis function of the mesh. Thus, every single spectral coefficient is related with all the vertex geometry. A light attack of the mesh is not likely to modify the basis functions and the spectral coefficients. Secondly, the security is easily enforced in the spectral methods. It is highly unlike to recover the embedded message by studying the pattern of the mesh geometry without knowing the secret key and transforming the mesh into the spectral domain. Thus, it makes the malicious attacks more difficult to find the message.

The most prominent drawback is that spectral analysis requires very high computational cost. The computational efficient is incomparable with the spatial domain methods even using better decomposition scheme [126]. This shortcoming limits the spectral methods to apply on the large meshes [90] without reducing the mesh size. Thus, most of the spectral methods involve a mesh segmentation in order to reduce the computational complexity [2, 98].

From the distortion point of view, the spectral methods introduce more visible distortion on the smooth regions rather than the bumpy regions. Although the change of the spectral coefficients spreads the error over all the vertices, there is very few study about the relation between the spectral coefficients and the mesh geometry. Therefore, it is not possible to introduce an explicit mechanism to control the distortion introduced by the spectral watermarking.

In this chapter, we introduce a novel robust and blind 3D watermarking algorithm based on the spectral domain. This work is mostly based on our paper published in IWDW 2009 [89] which is an extension of [90] in ICIP 2008. We firstly introduce the idea of using the statistical features of spectral coefficients to embed the watermark. We use the combinatorial Laplacian matrix as the operator in the mesh spectral analysis. The high frequency coefficients are splitted into bands and each band is used as a bin to embed one bit of message. Each bin of coefficients forms a point cloud whose statistical characteristics are analyzed using the Principal Component Analysis (PCA). In the proposed approach, the distribution of the spectral coefficients is constrained to a sphere when embedding a bit of zero and to a squashed ellipse when embedding a bit of one. With the perfect reversibility of the spectral decomposition scheme by using the combinatorial Laplacian matrix,

the message can be easily recovered by checking the ratio between the first principal component and the last principal component of the high frequency coefficients.

The proposed method consists in applying spectral decomposition locally, in well defined patches of the graphical object. The proposed methodology has the following stages. Firstly, the object is robustly aligned along the principal axis calculated using the analytic volumetric moments. Then the object is decomposed into patches (*i.e.* connected spatial regions) of equal areas defined along the first and second principal axes. Lastly, the spectral analysis is performed on each patch and the spectral coefficients are extracted. We use the embedding scheme proposed in [90] to embed one bit of message into one patch generated in the last step. Unlike the previous work [90] which split the high frequency coefficients into bands, the robustness of the localized version [89] uses the whole high frequency of each patch as a embedding primitive. The results are compared with Cho's algorithm [31] and it shows that the proposed algorithm introduces less distortion.

There are two main contributions of our spectral methods. Firstly, unlike the other spectral domain methods which embed the message in the low and medium frequency non-blindly [77, 79, 98, 100]. Our method utilizes the high frequency coefficients to embed the watermark blindly. And the experiments show the frequency changes can also obtain relatively good visual quality and robustness. Secondly, the robustness is enforced by statistical analyzing the spectral coefficient rather than embedding the same bit into coefficients repetitively.

The reminder of the chapter is organized as follows. In Section 3.2 we describe the volumetric method for aligning the 3D graphical object, while in Section 3.3 we describe the algorithm for generating the equal area patches. The alignment method will also be used in Chapter 4 as the initialization of the geodesic calculation. In Section 3.4 the spectral graph theory is briefly introduced. Section 3.5 gives the details of the proposed watermark insertion and extraction based on spectral coefficients analysis. The experimental results and comparison with the state of the art are provided in Section 3.6. The conclusions of this study are drawn in Section 3.7.

3.2 Robust object alignment

The first step of watermarking is usually to register the object with respect to the original model [98, 104]. The original object is used as a reference model so that the new object can be translated and rotated on the coordinate system same as the original object. This category of methods are frequently used in the non-blind watermarking method. Thus, it is not suitable for our blind method. Object alignment is to find a stable pose for a 3D object which is invariant to translation, rotation and uniform scaling. This is common and important issue in 3D mesh processing methodology such as in watermarking, object matching [69], object retrieval etc. There are many ways to align the object.

The model symmetry has also been studied extensively for aligning the object into a meaningful orientation for processing [25,69]. The reflectance plane symmetry is probably the most robust alignment method for a class of similar 3D objects. So far there is no watermarking method using this kind of alignment algorithms, because their computational requirements is high.

The mostly classical alignment method in 3D graphics is the Principal Component Analysis. This method treats the object as a point cloud and estimates its principal components. The PCA method is fast but its shortage is the low robustness. As PCA does not consider the topology or the manifold information of the mesh object, the principal axis calculated by the point cloud can be very easily biased by attacks such as mesh simplification, quantization and additive noise. Some algorithms [75] use the PCA to align the object in the preprocessing stage.

A 3D object can also be realigned using its surface area moment or the volume moment. The surface area moment and volume moment are proposed by Tuzikov *al* [123]. They calculate the continuous area and volume moment of the object. Principal component of the moments are analyzed and the object is realigned with respect to its principal components. Both moments capture more information of the model than PCA and they are robust as long as the object is not severely distorted. However, the area moment is not as robust against the noise attack which modifies the surface area. We propose to use a robust alignment scheme called volume moment alignment which was proposed in [143] considering the robustness

and its computational efficiency.

The volume moments of a 3D object are defined as:

$$M_{pqr} = \int\int\int x^p y^q z^r \rho(x,y,z) d_x d_y d_z \tag{3.1}$$

where p, q, r are orders, and $\rho(x,y,z)$ is the volume indicator function (it equals to 1 if (x,y,z) is inside the mesh and to 0 otherwise). For a triangular face $f_i = \{\mathbf{v}_{i1}, \mathbf{v}_{i2}, \mathbf{v}_{i3}\} = \{(x_{i1}, y_{i1}, z_{i1}), (x_{i2}, y_{i2}, z_{i2}), (x_{i3}, y_{i3}, z_{i3})\}$ on a mesh object, the moments are defined as :

$$\begin{aligned}
M_{000}^{f_i} &= \tfrac{1}{6}|x_{i1}y_{i2}z_{i3} - x_{i1}y_{i3}z_{i2} - y_{i1}x_{i2}z_{i3} + y_{i1}x_{i3}z_{i2} + z_{i1}x_{i2}y_{i3} - z_{i1}x_{i3}y_{i2}| \\
M_{100}^{f_i} &= \tfrac{1}{4}(x_{i1} + x_{i2} + x_{i3}) \cdot M_{000}^{f_i} \\
M_{200}^{f_i} &= \tfrac{1}{10}(x_{i1}^2 + x_{i2}^2 + x_{i3}^2 + x_{i1}x_{i2} + x_{i1}x_{i3} + x_{i2}x_{i3}) \cdot M_{000}^{f_i} \\
M_{110}^{f_i} &= \tfrac{1}{10}(x_{i1}y_{i1} + x_{i2}y_{i2} + x_{i3}y_{i3} + \tfrac{x_{i1}y_{i2}+x_{i1}y_{i3}+x_{i2}y_{i1}+x_{i2}y_{i3}+x_{i3}y_{i1}+x_{i3}y_{i2}}{2}) \cdot M_{000}^{f_i}
\end{aligned} \tag{3.2}$$

In fact this corresponds to the moment of the tetrahedron linking this face to the coordinate system origin. The global moments of a mesh are obtained by summing these elementary moments over all facets (with the appropriate contribution sign). The complete set of explicit volume moment functions can be found in [123]. The object centre is defined as

$$\mu = (M_{100}/M_{000}, M_{010}/M_{000}, M_{001}/M_{000}), \tag{3.3}$$

and the 3×3 matrix of the second order moments of the 3D object is constructed as:

$$\mathbf{\Psi} = \begin{pmatrix} M_{200} & M_{110} & M_{101} \\ M_{110} & M_{020} & M_{011} \\ M_{101} & M_{011} & M_{002} \end{pmatrix} \tag{3.4}$$

The principal axes of the object are the eigenvectors obtained by applying eigen-decomposition to the covariance matrix $\mathbf{\Psi}$:

$$\mathbf{\Psi} = \mathbf{W}^T \mathbf{\Delta} \mathbf{W} \tag{3.5}$$

where $\mathbf{\Delta} = \{\delta_1, \delta_2, \delta_3\}$ is the diagonal matrix containing the eigenvalues assuming $\delta_1 > \delta_2 > \delta_3$ and $\mathbf{W} = [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \mathbf{w}_3]^T$ is the matrix whose columns are the eigenvectors of $\mathbf{\Psi}$. The eigenvalues $\{\delta_1, \delta_2, \delta_3\}$ characterize the extension of the object along its principal axes whose directions are defined by the corresponding eigenvectors. In order to define a unique alignment, we propose two constraints. Firstly, the three axes must conform the right hand rule such that the direction of the third axis will be well defined as the cross product of the first two. Furthermore, the valid alignment satisfies the condition that the third order moments M_{300} and M_{030} of the rotated object are positive. By following these constraints and the right hand rule, the principal axis alignment is unique [143] and far more robust than the alignment calculated by only using vertex coordinates.

3.3 Object patch generation

Watermarking in spectral domain owns a series of advantages including increased watermark key security and good watermark related robustness. However, as shown in [90], the application of blind spectral watermarking is limited to rather small graphical objects due to the high computational complexity requirements of spectral decomposition for large meshes.

Thus, an object segmentation scheme is required for reducing the Laplacian matrix size in order to improve the computational efficiency. There are many segmentation or clustering methods that have been used in the watermarking literature. One popular category of patch generation algorithm is seed-generated, *i.e.* the patches are generated according to seeds on the object surface. Ohbuchi *et al* use a random seed to generate patches on the object surface [98]. Konstantinides [75] propose a similar method as Ohbuchi's. They firstly cast a random ray from the centre of the object. The intersection between the ray and the surface is defined as a seed. The patch is generated around the seed with a certain geodesic boundary. However, there are a obvious shortcoming with this kind of patch generation. The seeds can be close to each other so that the patches will be overlapped. The causality problem which means the new watermarked patch will remove the previously embedded watermarking will certainly happen in this case. Another approach of patch gen-

eration is feature point based segmentation method [2,3]. Alface *et al* [2] proposed to re-triangulate the object surface into patches by using the detected robust feature points based on its curvature characteristics. The feature points are connected by geodesic lines such that vertices in the same the region is considered as a compact patch. Alface *et al* [3] also proposed another feature point based segmentation method in 2007. In [3], the protrusion points are detected and used as seed points. Then, the patches are generated as a geodesic circle around the protrusion points. This category of patch generation methods strongly requires the robustness of the feature point detection algorithm. The feature points may be different when the surface is strongly attacked by random noise. In addition, the protrusion points detection are computationally very expensive, it may take a few hours for detection process as we shown in Section 4.4.6 on page 95.

In this chapter in order to apply the spectral algorithm to large 3D objects, we propose a simple and computational efficient method to split the mesh into segments (*i.e.* spatial compact regions) so that each segment is used for carrying one bit of the message. The number of patches is generated as a sum of the number of bits and a small integer number. The small integer can be generated randomly seeded by the secret key. So the total patches is a bit more than the message length in order to improve the security. After aligning the graphical object as described in Section 3.2 we trim away the extremes of the object as defined along its principal axis $\mathbf{w}_1$. In this way we increase the watermark security. Then the trimmed object is split into layers which are defined by planes perpendicular to $\mathbf{w}_2$, the second principal axis. Finally, the vertices and triangles from each layer are divided into connected patches of equal areas in a direction along the first principal axis $\mathbf{w}_1$. All these steps are detailed below.

Let us consider x_{max} and x_{min} the maximum and minimum value along the first principal axis $\mathbf{w}_1$. $\alpha \in [0, 0.15]$ is a value generated by the secret key which is used for trimming the extremities of the object. We define two boundary values along the first principal axis $\mathbf{w}_1$ such that:

$$\begin{cases} B_{min} = (x_{max} - x_{min}) \cdot \alpha + x_{min} \\ B_{max} = (x_{max} - x_{min}) \cdot (1 - \alpha) + x_{min} \end{cases} \tag{3.6}$$

All the vertices whose x coordinates are outside this range, *i.e.*, $\mathbf{v}_x < B_{min}$ and $\mathbf{v}_x > B_{max}$ will be excluded from the watermark embedding process. Let us define the trimmed object $\mathcal{O}_T$ as:

$$\mathcal{O}_T = \{\mathbf{v} | B_{min} \leq v_x < B_{max}, \ \ \forall \mathbf{v} \in \mathcal{O}\} \tag{3.7}$$

The total area of the trimmed object considered for watermarking, denoted as A_t, is defined as the sum of all polygons, usually triangles, which are located on the surface of the trimmed object $\mathcal{O}_T$.

$\mathcal{O}_T$ is then split into κ layers defined by planes perpendicular onto the second principal axis, *i.e.* $\mathbf{w}_2$. κ is chosen as a ratio as:

$$\kappa = \delta_2 / \delta_3 \tag{3.8}$$

Other schemes to define κ can also be used, for example, fixed $\kappa = 2$. Thus, there are $\kappa + 1$ boundary values defined by:

$$y_i = y_{min} + \frac{y_{max} - y_{min}}{\kappa} \cdot i \tag{3.9}$$

where $i = 0, \ldots, \kappa$. The vertices of the trimmed object are thus split into κ layers and for each layer we have:

$$L_i = \{\mathbf{v} | y_{i-1} \leq \mathbf{v}_y < y_i, \ \ \forall \mathbf{v} \in \mathcal{O}_T\} \tag{3.10}$$

where $i = 1, \ldots, \kappa$. These layers are then divided so as to obtain a set of N patches of equal areas.

The desired area for each patch is calculated according to the following equation:

$$A_p = \frac{A_t}{N} \tag{3.11}$$

where $N > M$ and M is the number of bits to be embedded in the object. The reason for which more patches are generated than the number of bits to embed is because a segment may be eliminated if its area is smaller than a pre-defined

threshold, this may happen due to surface folding while we also aim to increase the watermark security by deliberately excluding specific patches. For example, the last patch of each layer is likely to have an area smaller than A_p and thus if watermarked may result into a non-uniform embedding capacity.

Next, we sort all vertices of all layers L_i in ascending order of their x coordinates, *i.e.* along the first principal axis $\mathbf{w}_1$. Vertices are then iteratively added into a patch from left to right of the sorted sequence; when the area A_p is attained for the current growing patch, a new patch is initiated.

Let us denote $\mathcal{P}_j$ as the j^{th} patch generated, where $j = 1, \ldots, N$. The first M patches $\mathcal{P}_1, \ldots, \mathcal{P}_M$ are used for watermarking. The patches to be watermarked can be picked up randomly according to the watermark key. This patch segmentation mechanism is summarized in the Algorithm 1.

When one triangle is crossing several boundaries of layer planes, that triangle will be split and only the area of its section which is within the layer L_i will be accounted into the current patch area. For example, as shown in Figure 3.1, triangle $\triangle ABC$ is located at the intersection of two different layers; B_{max} indicates the trimming boundary value. Only the area of the red region, *i.e.* polygon $ADEF$ is accumulated in the current patch area. Ideally, one segment should contain only one compact 3D patch surface. However, if an object has a complex topology, *e.g.* the graphical object contains many holes, one patch may include several small and isolated surface regions. Watermarking such discontinuous patches will result into visible distortions and may cause visible artifacts on the graphical object surface following spectral watermarking. Therefore, after the segmentation we do not consider for watermarking those patches which contain discontinuous areas that are smaller than a predefined threshold.

There are several advantages for the proposed layer segmentation algorithm. Firstly, the patches generated using this algorithm are highly secure; indeed the percentage of the trimmed extremities is generated according to a secret key and it is therefore impossible to recover the patches without the knowledge of this secret key. By increasing the number of layers we can provide more compactness to the patches. Patches which are closer to a square-like shape are more appropriate for spectral watermarking since they provide area compactness and a high connectivity

Algorithm 1 Patch Segmentation Algorithm

```
1: seg = 0; // Patch index
2: for i = 1 to κ do
3:    Let V be the sorted sequence of ∀v ∈ L_i in ascending order of the x coordinate
4:    A = 0;
5:    for j = 1 to |L_i| do
6:       v_j = V[j];
7:       Add v_j into patch P_seg;
8:       for all Neighbouring face f incident to v_j do
9:          if f is not processed and each vertex of f has been assigned to a patch
            then
10:            Calculate the area A_in inside the layer L_i;
11:            Increment the area accumulator A+ = A_in;
12:            if A > A_P then
13:               seg++; // Move to the next patch;
14:               Move v_j to patch P_seg
15:               A = A − A_p // Residue area is assigned to the next patch
16:            end if
17:         end if
18:      end for
19:   end for
20:   Mark all v in the last patch of the layer L_i as −1.
21: end for
```

which are both beneficial for watermarking. On the contrary, if only one layer is used, the algorithm splits the object into narrow strips which contain a lower level of mesh connectivity than a square-like patch. By adjusting the parameter κ, we therefore have the flexibility to adjust the size and the shape of the patches.

According to the experimental results the proposed patch segmentation algorithm is robust against most of the mesh attacks including additive noise, mesh simplification and Laplacian smoothing. Moreover, the process of watermarking the 3D object will hardly affect the segmentation in the detection stage. This robustness is a very strong point for a practical watermarking application. The proposed algorithm produces patches of equal areas. That also constitutes a very strong issue for watermarking; indeed since each patch will carry one bit of the watermark, each bit will have a high robustness. Two examples of segmentation (one layer and three layers) of the Venus head object are illustrated in Figure 3.2(a) and 3.2(b), respectively. From these figures it is clear that three layers segmentation produces more compact patches than a single layer segmentation.

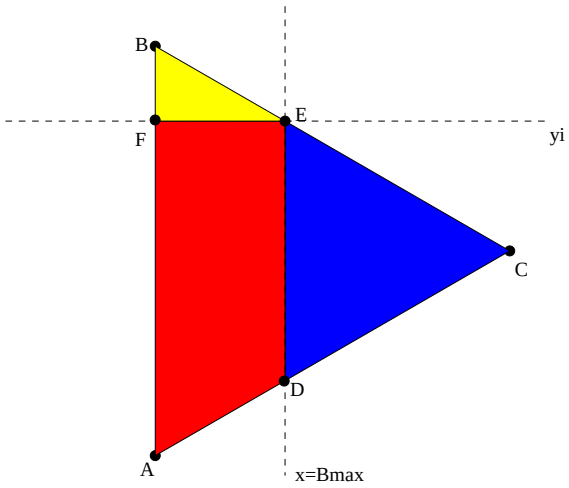


Figure 3.1: An example of triangles splitting over different patches, where each colour corresponds to a different patch.

3.4 Spectral decomposition of mesh patches

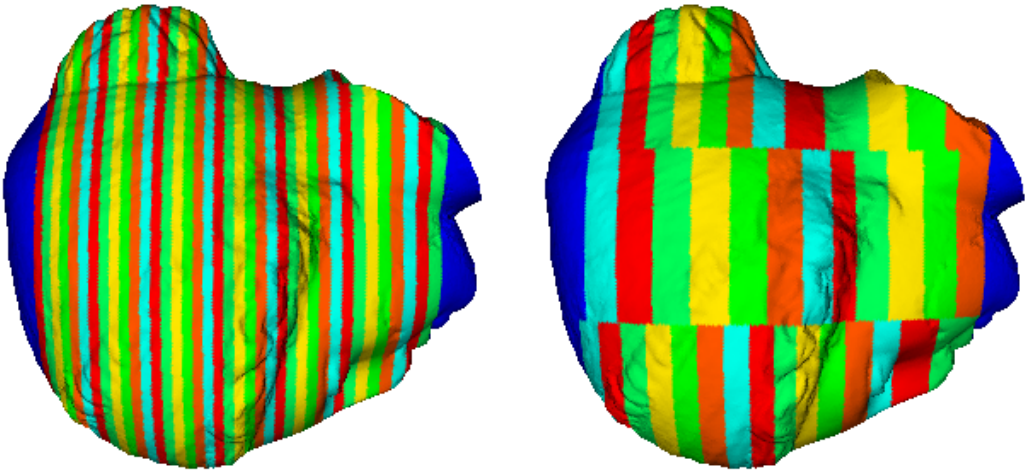
The theory background of the spectral decomposition of a mesh object using the combinatorial Laplacian matrix has been introduced in Section 2.1.1.1 Chapter 2. Here we apply the spectral analysis on the local patches of the mesh. A patch $\mathcal{P}_i$ consists of a set of vertices $\{\mathbf{v}_j, j = 1, \ldots, |\mathcal{P}_i|\}$ where $|\mathcal{P}_i|$ is the number of the vertices within the patch $\mathcal{P}_i$, and a set of edges characterizing the connectivity information. The Laplacian matrix $\mathbf{L}^i$ is calculated as the difference between the degree matrix and the adjacency matrix and has the following entries:

$$L^i_{j,k} = \begin{cases} |\mathcal{N}_{\mathbf{v}_j}| & \text{if } j = k \\ -1 & \text{if } j \neq k \text{ and } \mathbf{v}_j \text{ adjacent to } \mathbf{v}_k \\ 0 & \text{otherwise} \end{cases} \tag{3.12}$$

where $|\mathcal{N}_{\mathbf{v}_j}|$ represents the degree (valence) of the vertex $\mathbf{v}_j$ (the number of neighbouring vertices $\mathbf{v}_k \in \mathcal{N}_{\mathbf{v}_j}$). The Laplacian matrix is eigen-decomposed as :

$$\mathbf{L}^i = \mathbf{q}_i^T \Omega_i \mathbf{q}_i \tag{3.13}$$

where Ω_i is a diagonal matrix containing the eigenvalues of the Laplacian, and $\mathbf{q}_i$ is a matrix containing its eigenvectors. The eigenvectors of $\mathbf{L}^i$ constitute an orthogonal basis and the associated eigenvalues are considered as frequencies. Here



(a) One layer $\kappa = 1$ (b) Three layers $\kappa = 3$

Figure 3.2: Patch segmentation of the Venus head graphical object. The blue regions represent the trimmed extremities along the first principal axis $\mathbf{w}_1$.

we assume that $\mathbf{q}_i$ and $\mathbf{\Omega}_i$ are sorted in ascending order according to their corresponding eigenvalues in the diagonal matrix $\mathbf{\Omega}_i$. The spectrum is provided by the projections of each vertex coordinate on the directions defined by the basis function $\mathbf{q}_i$. The spectral coefficients $\mathbf{C}_i$ are calculated as:

$$\mathbf{C}_i = \mathbf{q}_i \mathbf{V}_i \tag{3.14}$$

$\mathbf{V}_i$ is the set of spatial coordinates of the vertices of the patch.

The transformation can be reversed and the patch vertices can be recovered as:

$$\mathbf{V}_i = \mathbf{q}_i^T \mathbf{C}_i \tag{3.15}$$

As the construction of the Laplacian matrix solely depends on the connectivity but not the geometry of the mesh, the changes of the spectral coefficients $\mathbf{C}_i$ only affect the geometry $\mathbf{V}_i$. Therefore, the connectivity and the basis functions $\mathbf{q}_i$ remain the same. This enables us to perfectly retrieve the message after watermarking which only modifies the spectral coefficients.

3.5 Watermarking using constraints on the PCA axes of the spectral coefficients

3.5.1 Watermark Embedding

The spectral coefficients can be divided into "low frequency" and "high frequency". The low frequency reflects the large scale information of the patch while the high frequency corresponds to the details of the patch. Changing the low frequency coefficients may result in severe deformation and the shearing of the object. In contrast, changing the high frequency could introduce noisy like effects on the patch surface. In this chapter, we propose to embed the watermark in the high frequency coefficients so as to minimize the introduced geometric distortion. In the following we consider the distribution of the highest 70% of the spectral coefficients for watermarking. The high frequency coefficients of each patch form a point cloud in the 3D space. The shape of this point cloud is analyzed in the following way by using Principal Component Analysis (PCA). The mean and covariance matrix of each set of points are calculated as:

$$\mu_i = \frac{\sum_{j=1}^{n} \mathbf{C}_{i,j}}{n} \tag{3.16}$$

$$\Sigma_i = \frac{1}{n} \sum_{j=1}^{n} (\mathbf{C}_{i,j} - \mu_i)^T (\mathbf{C}_{i,j} - \mu_i) \tag{3.17}$$

where n is the number of frequency coefficients. The covariance matrix Σ_i is decomposed as:

$$\Sigma_i = \mathbf{U}_i^T \Lambda_i \mathbf{U}_i \tag{3.18}$$

where Λ is the diagonal matrix containing the eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ where we assume $\lambda_1 > \lambda_2 > \lambda_3$. $\mathbf{U}_i$ is the transformation matrix whose columns are the eigenvectors of Σ_i. The eigenvalues $\{\lambda_1, \lambda_2, \lambda_3\}$ determine the extension (variance) of the point cloud along the axes defined by the eigenvectors. The spectral coefficients of a patch are shown as a signal for the x axis in Figure 3.3(a) and as a 3D distribution in Figure 3.3(b).

The watermark embedding method has three steps. Firstly, the point cloud of spectral coefficients $\mathbf{C}_i$ is rotated so that its axes coincide with the orthogonal axes

defined by the eigenvectors:

$$\mathbf{D}_i = \mathbf{C}_i \mathbf{U}_i \tag{3.19}$$

such that the variances along the three axes will not be correlated.

The cloud of 3-D points of $\mathbf{C}_i$ is then "squashed" for embedding a bit of 1 and "inflated" to a sphere for embedding a bit of 0, by using the ratio between the eigenvalues :

$$\frac{\lambda_1}{\lambda_k} = K \quad \begin{cases} K > 1 & \text{for a bit of 1} \\ K = 1 & \text{for a bit of 0} \end{cases} \tag{3.20}$$

where $k \in \{2, 3\}$. In order to enforce these constraints, the variance along the second and third axis is changed without affecting the variance corresponding to the largest eigenvalue :

$$\hat{D}_{i,k} = D_{i,k} \sqrt{\frac{\lambda_1}{K\lambda_k}} \tag{3.21}$$

where $k \in \{2, 3\}$, λ_1 is the largest variance and $\hat{D}_{i,k}$ represents the modified component of the coefficient vector after embedding the watermark. For embedding a '1' bit, K is set to be larger than 1 and smaller than a reasonable value ensuring the point cloud is not over-squashed, and for embedding a '0' bit, K is set to be 1 in order to inflate the point cloud into a perfect sphere. Figures 3.3(c) and 3.3(d), illustrate the shape of the coefficients cloud after embedding a bit of 0 or 1, respectively.

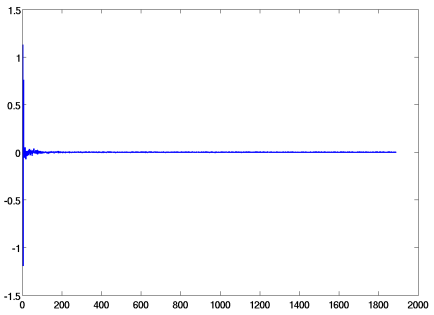
The watermarked spectral coefficients of high frequency are reconstructed as:

$$\hat{\mathbf{C}}_i = \hat{\mathbf{D}}_i \mathbf{U}_i^T \tag{3.22}$$

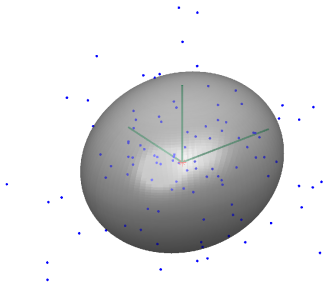
Finally, we enforce the changes back to the watermarked coefficients $\hat{\mathbf{C}}_i$. The watermarked object is obtained by applying the reverse transformation of equation (3.15) using the watermarked coefficients $\hat{\mathbf{C}}_i$.

3.5.2 Watermark extraction

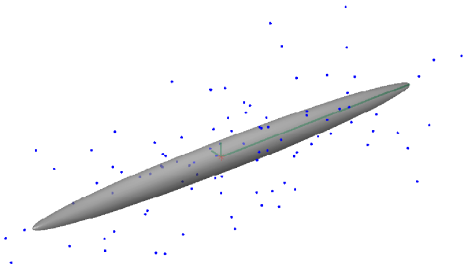
The proposed spectral PCA watermarking detection stage does not require the original object for retrieving the watermark. The detector needs to know the secret key, the watermarking algorithm and the number of bits that was embedded in order to



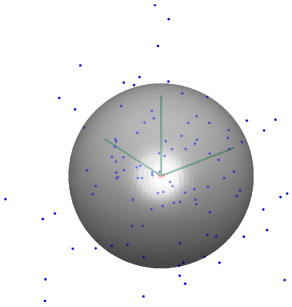
(a) Spectral coefficients



(b) Ellipse of the original coefficients



(c) Ellipse with 1 embedded



(d) Ellipse with 0 embedded

Figure 3.3: Enforcing constraints into spectral coefficients of meshes.

recover the watermark message. The secret key is used to generate the α and the length of the message is used to generate the patches. As a statistical method, it is impossible to generate the patches without knowing the length of the message. First of all, the mesh object is aligned and segmented as explained in Sections 3.2 and 3.3. Then, we apply the spectral analysis on each patch and extract the spectral coefficients in the same way as proposed in Section 3.4. Finally, we calculate the ratio between the largest and the smallest eigenvalues of the point cloud formed by the watermarked coefficients and retrieve the information bit as :

$$\begin{aligned} &\text{if } \frac{\hat{\lambda}_1}{\hat{\lambda}_3} > T \quad &&\text{then } bit = 1 \\ &\text{otherwise} \quad &&\text{then } bit = 0 \end{aligned} \tag{3.23}$$

where T is a threshold which depends on the embedding level K.

The other spectral domain 3D watermarking methods in the literature embed watermark by modulating a single spectral coefficient or repetitively a number of spectral coefficients. Unlike those methods, the method we proposed here is the first one that use the statistical feature of the spectral coefficients to embed watermark.

In comparison with the other methods, the method proposed in this chapter improves the robustness because a statistical feature is more difficult to be destroyed.

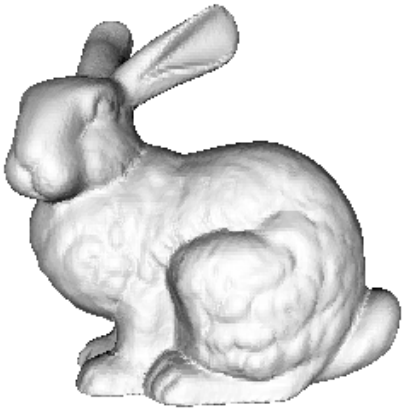
3.6 Experimental results

3.6.1 Models and parameters

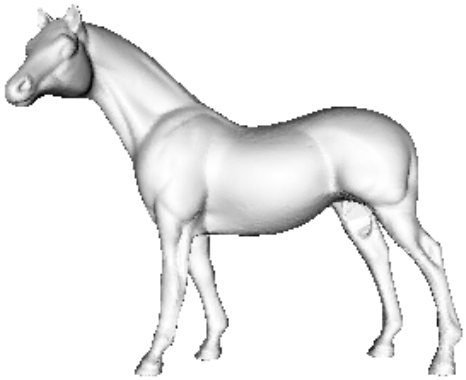
The proposed 3D watermarking algorithm is applied on four different mesh objects: Bunny with $34,835$ vertices and $69,666$ faces, Horse with $67,583$ vertices and $135,162$ faces, Buddha with $89,544$ vertices and $179,222$ faces and Venus head with $134,345$ vertices and $268,686$ faces. It can be observed that these objects contain many vertices and faces. Bunny is an object with bumpy surface. Horse contains large regions with very few variations. The Budda object is a topologically complex one. And finally, the Head object is a relatively large 3D model. The appearance of those models are shown in Figure 3.4. Each object is split into $N = 70$ patches grouped into $\kappa = 2$ layers as described in Section 3.3 while embedding a total of $M = 64$ bits. $\kappa = 2$ is selected empirically. By splitting them into patches as explained in Section 3.3 we reduce the required computational complexity for spectral watermarking. The watermarking algorithm parameters are set as: $\alpha = 0.1$, $K = 15$ and $T = 2.25$, as used in equations (3.6), (3.20) and (3.23), respectively. The $K = 15$ is chosen to make sure the point cloud is squeezed enough to distinguish with the perfect sphere. As the coefficients are the high frequency, the change of the coefficients is very small. The detection threshold $T = 2.25$ is chosen empirically by taking the average value of first principal component and third principal component of spectral coefficients point cloud over all the patches. $K = 15$ is also chosen empirically as it can sufficiently distinguish the shape of the coefficients point cloud as shown in Figure 3.3 (c) and (d).

3.6.2 Watermarked spectral coefficients

Figure 3.3(b) shows a point cloud of original spectral coefficients of a patch. The ellipsoid which describes the distribution of the point cloud in the 3D space is characterized by three principal axis obtained in PCA. Figures 3.3(c) and (d) illustrate



(a) Bunny



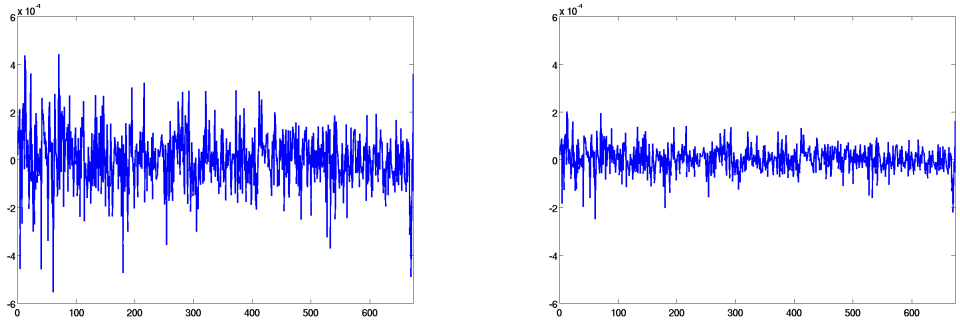
(b) Horse

(c) Budda

(d) Head

Figure 3.4: Models used in experiments.

the case when embedding a '1' and a '0' bit, respectively. As we can see from those figures, the shape of the ellipsoid mimic the shape of the Arabic number '1' and '0'. This is actually the inspiration of our algorithm. The spectral coefficients corresponding to the y axis coordinate, *i.e.* corresponding to the second component eigenvector, of an original and a watermarked segment are shown in Figures 3.5(a) and 3.5(b), respectively. It can be observed that the amplitude of the high frequency coefficients is shrink after watermarking a bit of '1'. However this kind of high frequency modulation only introduces small geometric distortions on the shape.



(a) Original (b) After embedding a bit of '1'

Figure 3.5: Spectral coefficients corresponding to the y axis component.

3.6.3 Comparative assessment of the object distortion

The proposed watermarking algorithm is compared with the state-of-art robust algorithm proposed by Cho *et al.* in [31]. We denote by ChoMean and ChoVar, the mean change and the variance change algorithms, described in [31]. We set the watermark parameter $\alpha = 0.05$ for the ChoMean and ChoVar methods according to their embedding algorithm from [31].

The distortion introduced by our spectral watermarking algorithm is compared objectively and visually. We use the MRMS proposed in [34] as the numerical objective comparison measurement. As we introduced earlier in Section 2.4.1, MRMS is the most popular Hausdorff error implementation for graphics so far. It is widely used by watermarking community and is generally accepted as the best quality measurement tool. Details can be found in Section 2.4.1 Chapter 2 and the equation can be found in equation 2.16 on page 34.

The comparison of the visual distortions is shown in Figure 3.6. It is easy to see that Cho's methods produce staircase-like distortions on the graphical object surface. The distortion is quite large and clearly visible. On the other hand, our method introduces less distortion than Cho's method. Some visible noise can be observed at the boundary of the patches. The distortion is more visible on the flat regions such as the body of the Horse and the belly of the Budda. While the distortion is not obvious on the regions with many variations. However, this level of distortion may limit the real application of this method. This verifies the distortion visibility study made By Bors in [19]. The numerical results are listed in Table 3.1.

From these results it is clear that the algorithm proposed in this chapter introduces less distortion than Cho's algorithms for all four objects from both geometric and visual points of view.

Object	Spectral	ChoMean	ChoVar
Bunny	0.25	0.49	0.28
Horse	0.43	0.67	0.44
Budda	0.26	0.46	0.26
Head	0.10	0.25	0.15

Table 3.1: Geometric distortions measured by MRMS ($\times 10^{-4}$).

3.6.4 Robustness comparison

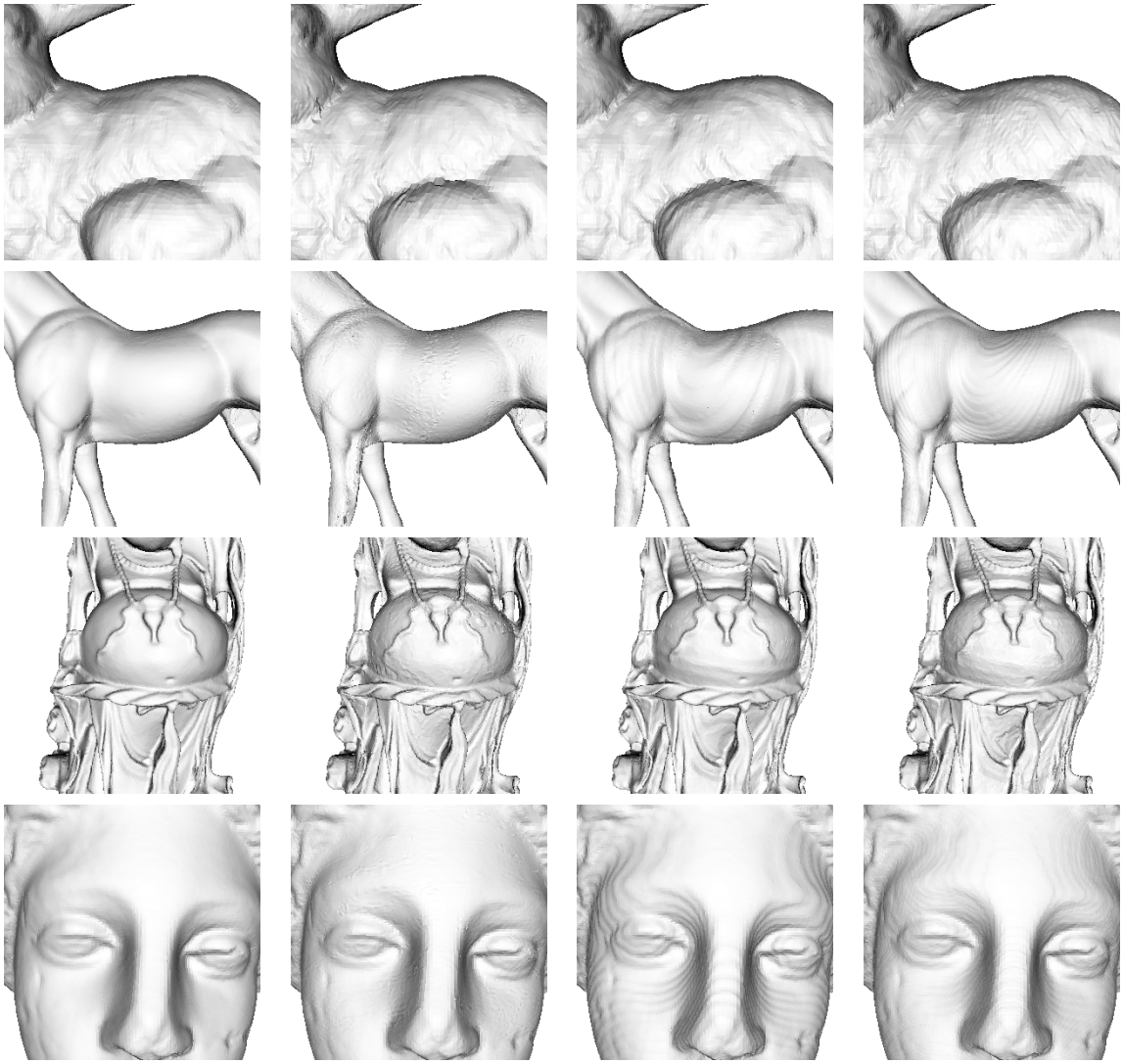
The robustness comparison results of the three algorithms against various attacks such as additive noise, mesh simplification, quantization, Laplacian smoothing and uniform resampling. As mentioned in Section 2.4.2, we consider that the set of these five attacks can sufficiently test the robustness of a watermarking algorithm.

Noise added to the vertex locations of the watermarked object can model a large category of possible attacks which may affect the ability to retrieve the watermark code. In the following we consider additive random noise according to the following distortion equation :

$$\tilde{\mathbf{v}}_i = \hat{\mathbf{v}}_i + \epsilon \|\hat{\mathbf{v}}_{max}\| \overrightarrow{\mathbf{p}} \tag{3.24}$$

where $\tilde{\mathbf{v}}_i$ represents the distorted watermarked vertex $\hat{\mathbf{v}}_i$, $\epsilon \in [0, 1]$ is the percentage of $\|\mathbf{v}_{max}\|$ which corresponds to the largest Euclidean distance measured from the object center to each vertex, $\overrightarrow{\mathbf{p}}$ is a unitary vector with random direction. The direction of $\overrightarrow{\mathbf{p}}$ spans the entire range of possible angles. Figure 3.7 shows the comparison between our algorithm and Cho's algorithms. In the case of the Budda and Head object, the results are rather similar for the three methods while Cho's algorithms demonstrate a higher robustness in the case of Bunny and Horse objects.

We use the Laplacian Smoothing method proposed in [119] for the smoothing attack. The Laplacian Smoothing calculates the position of a vertex by averaging its neighbours while the weight is considered according to its local connectivity. The

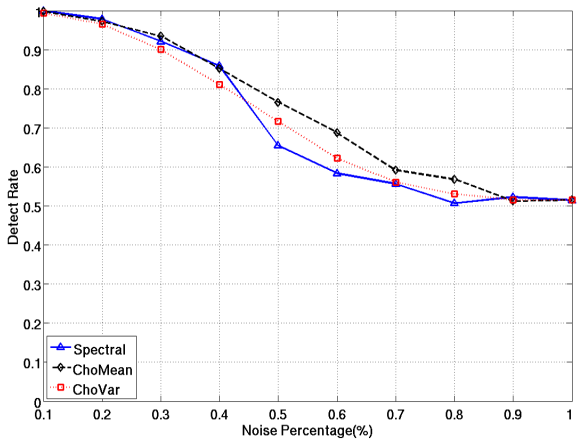


(a) Original (b) Spectral (c) ChoMean (d) ChoVar

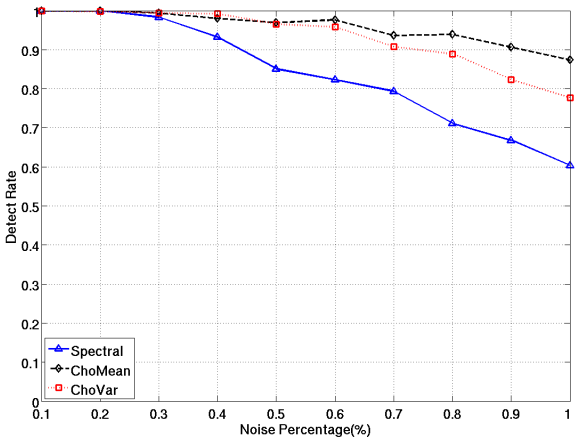
Figure 3.6: Visual distortions introduced by different watermarking algorithms.

method consider the local Laplacian as a high frequency filter. We consider the smoothing parameter $\lambda = 0.2$ and 10 iterations used in our experiments. The plots displaying the robustness of the watermarking methods are provided in Figure 3.8 for all four graphical objects. Basically the three algorithms have the similar robustness but our method performs better for fewer iterations in the smoothing attack.

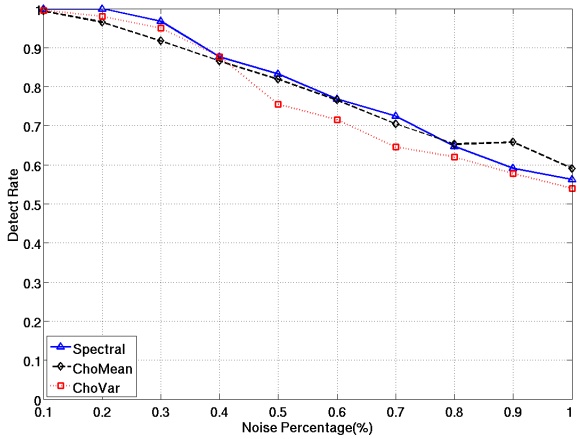
Mesh simplification is used in graphics for compressing, coding and in other applications [43]. In the case of watermarked objects mesh simplification is considered as a potentially destructive attack for the watermark. The quadratic metric simplification software described in [50] was used for testing the robustness at mesh simplification. This algorithm collapses two vertices connected by an edge into one



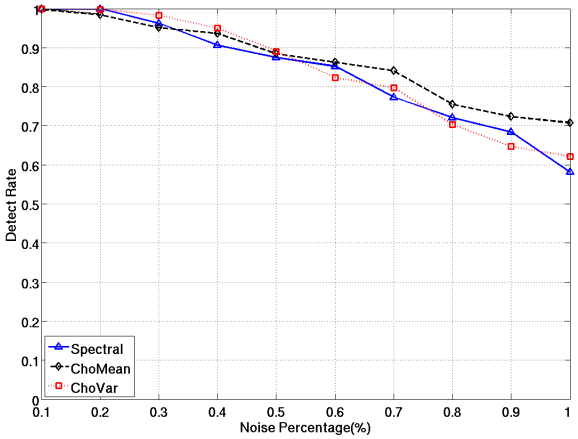
(a) Bunny



(b) Horse



(c) Budda

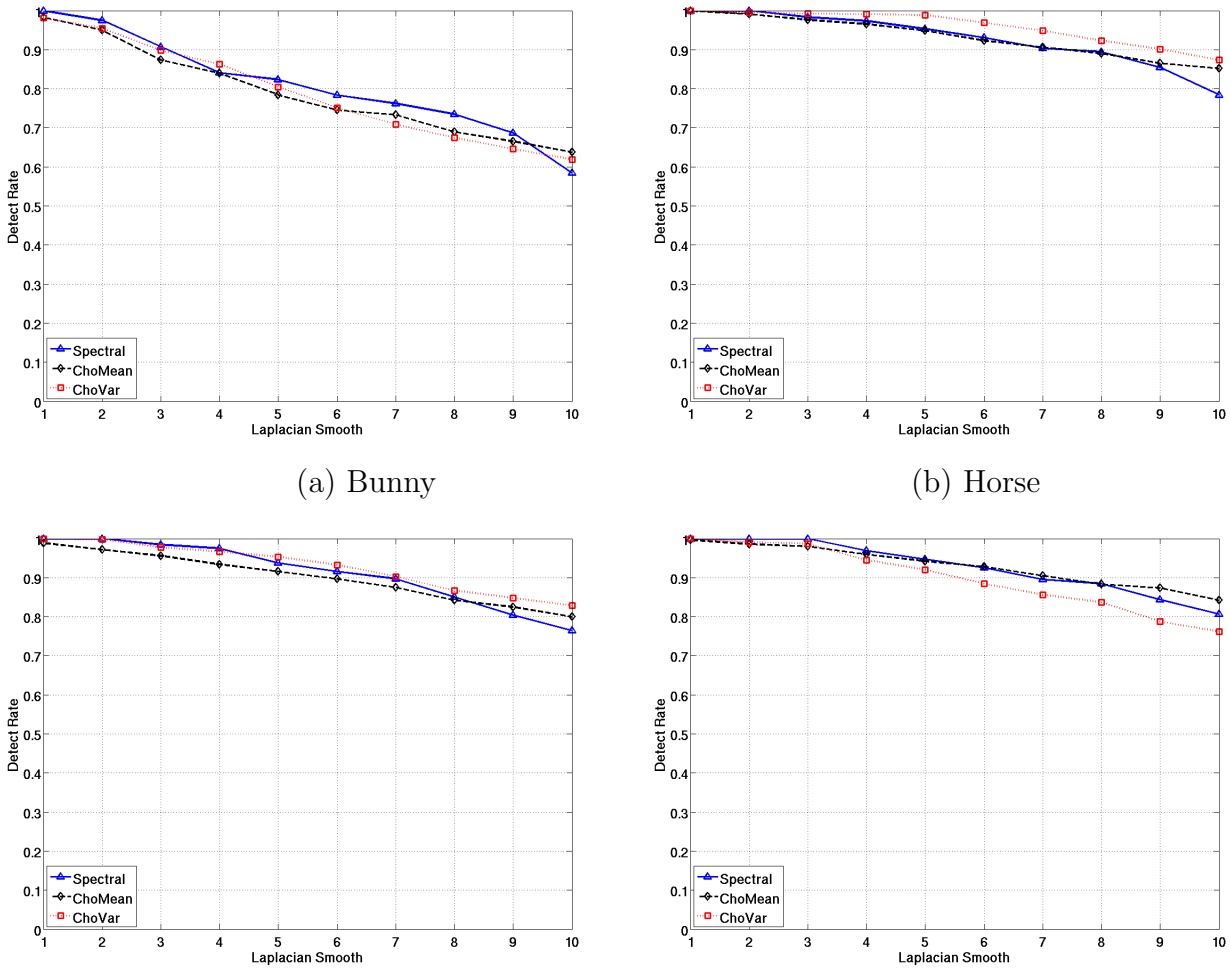


(d) Head

Figure 3.7: Robustness against the additive noise attack.

which position is chosen ensuring that the introduced error is minimum. This approach for mesh simplification was chosen because it represents a harder to resist attack for the watermark than other mesh simplification algorithms. Figure 3.9 contains the plots showing the resistance to mesh simplification attack for the objects when using all four methods. Cho's results are better in general when considering mesh simplification. The reason is that our algorithm modifies the high frequency for embedding watermark, while the mesh simplification removes the high frequency details of the mesh. As a result, the behaviour of the mesh simplification attack erases the watermark from the high frequency.

The quantization attack is defined as the difference between the maximum and minimum value of vertices along each axis when they are quantized by a specific



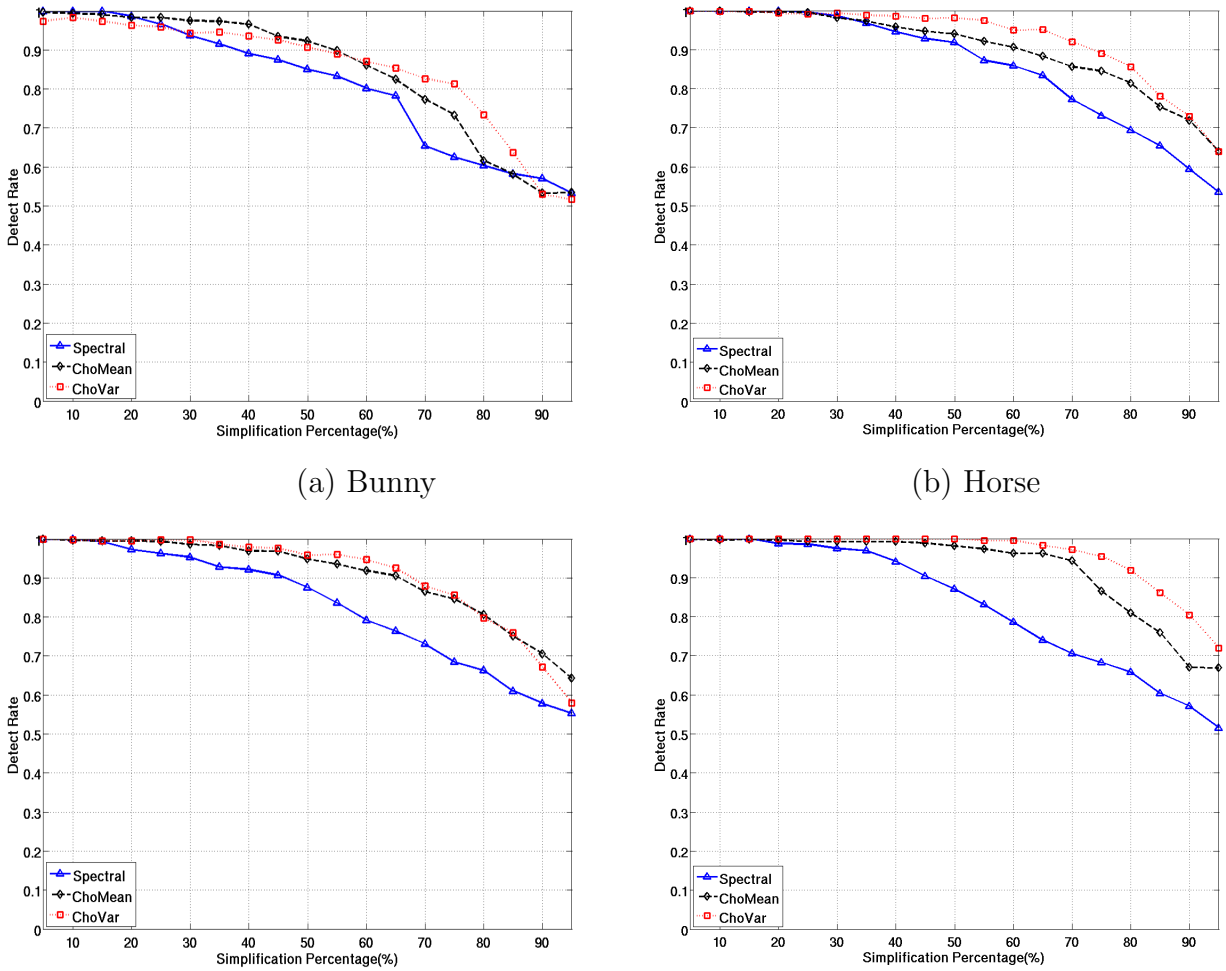
(a) Bunny (b) Horse

(c) Buddha (d) Head

Figure 3.8: Robustness against the Laplacian smoothing.

number of bits. We test the robustness of the algorithms from 11 bits quantization to 5 bits quantization. Figure 3.10 provides the plots of the robustness against the quantization attack. It can be observed from these plots that the proposed algorithm is better than Cho's algorithm in this attacking test.

We compare the robustness of watermarking methods against the uniform resampling attack by using the algorithm proposed in [13]. This attack consists of uniformly sampling random vertices from the graphical object surface and meshing them in a way that is not related to the original object. The vertices are sampled on the local tangent plane. Therefore, the new vertices are not guaranteed that they would lie on the initial surface. The number of sampled vertices represent $\{100\%, 80\%, 60\%, 40\%, 20\%\}$ from the total number of vertices in the original object.



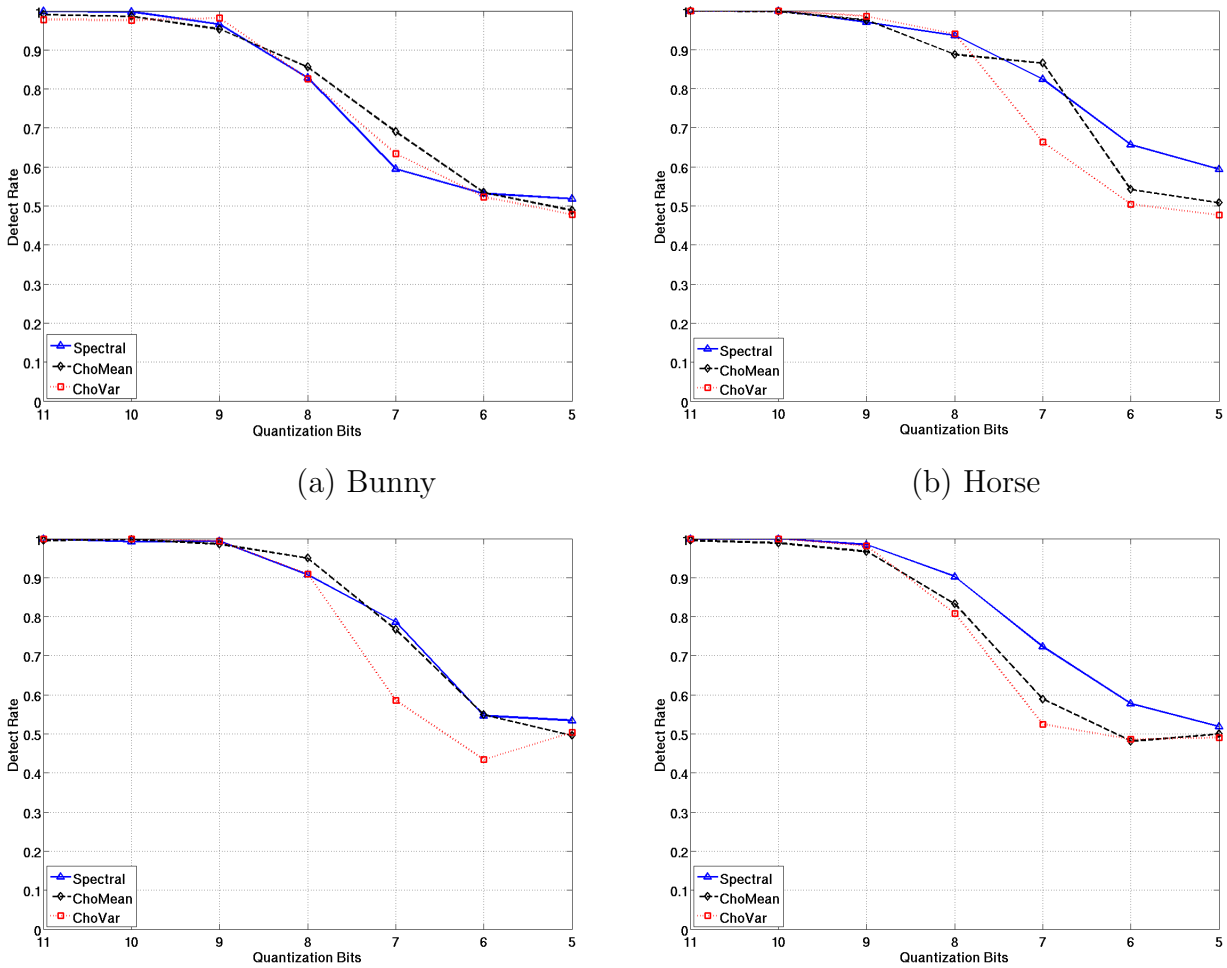
(a) Bunny (b) Horse

(c) Buddha (d) Head

Figure 3.9: Robustness against mesh simplification.

The results are shown in Figure 3.11. Our algorithm provides a better performance in the Budda object but worse than Cho's Mean methods in the other cases.

The comparison demonstrates the good trade-off of our method between the watermark robustness and the distortion. Indeed our algorithm introduces a lower distortion (both geometric and visual) than Cho's methods at the price of a lower robustness for certain attacks, while maintaining the same robustness for other attacks.



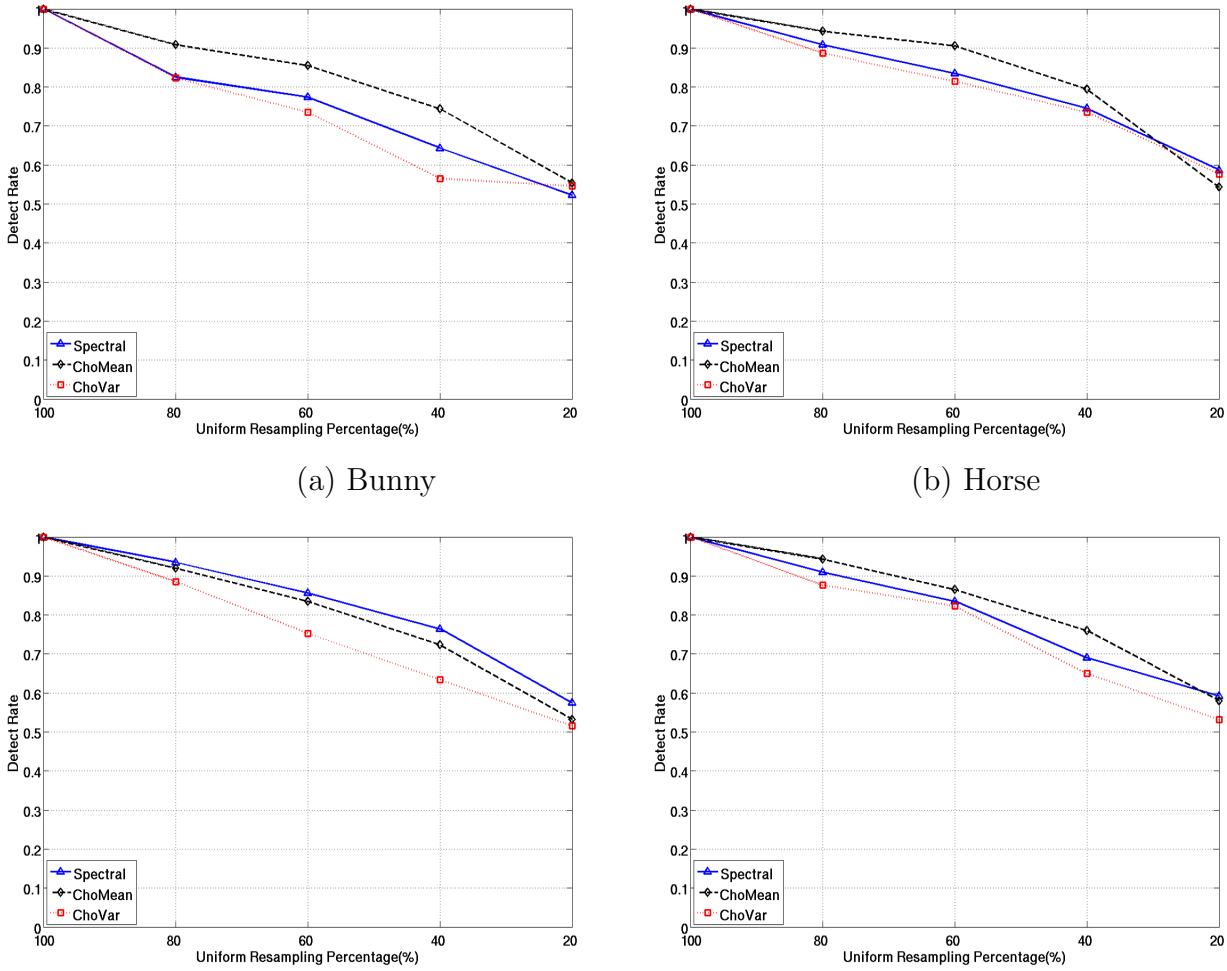
(a) Bunny (b) Horse

(c) Buddha (d) Head

Figure 3.10: Robustness against quantization.

3.6.5 Alignment and segmentation

In this study we evaluate the influence of the object centre location and the axes alignment orientation for the watermark detection performance. We use the Bunny graphical object in order to compare the robustness of the principal axis alignment for the following methods: volume moments, surface moments, PCA alignment. The surface moment alignment is similar to the volume moment alignment and both are described in section 3.2 as well as in [123]. The error in center location, after considering various mesh surface attacks, is given by the Euclidean difference between the object centres before and after the attack. The alignment error is measured by the angle difference between the object principal axes before and after



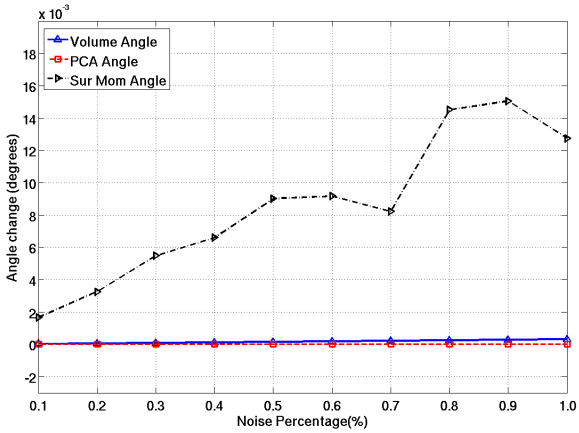
(a) Bunny (b) Horse

(c) Buddha (d) Head

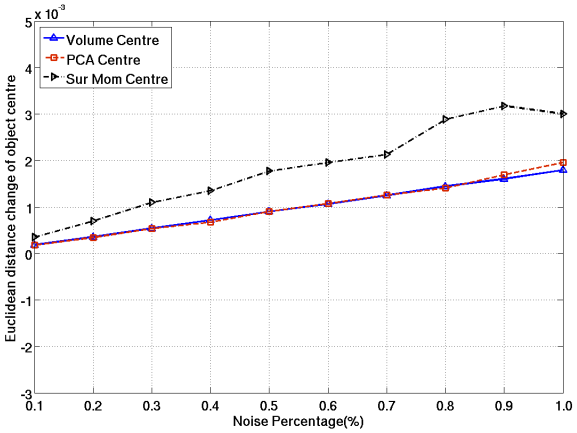
Figure 3.11: Robustness against uniform resampling.

the attacks. The results for the angle error between the principal axes and the resulting bias in the object centre location, when considering additive noise to the watermarked object, are shown in Figures 3.12(a) and 3.12(b), respectively. Results of the same error evaluations when considering mesh simplification are shown in Figures 3.12(c) and 3.12(d), respectively. Clearly, the volume alignment method has the highest robustness in all these cases.

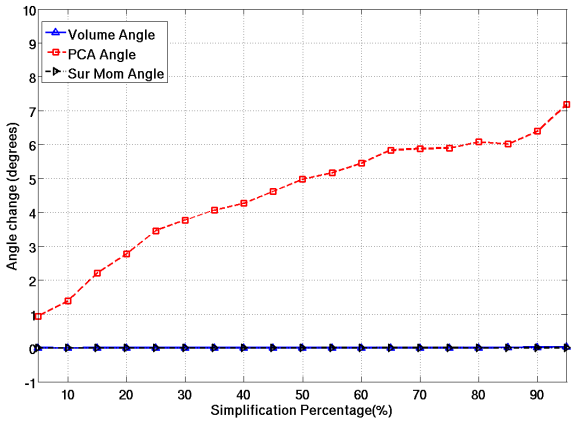
In Figure 3.13 we experimentally examine the robustness of the equal area segmentation method proposed in Section 3.3. The segmentation result on the original object is shown in Figure 3.13(a). The results obtained after considering additive noise, mesh simplification and Laplacian smoothing are shown in Figures 3.13(b), 3.13(c) and 3.13(d), respectively. For the simplification, we used the quadric error



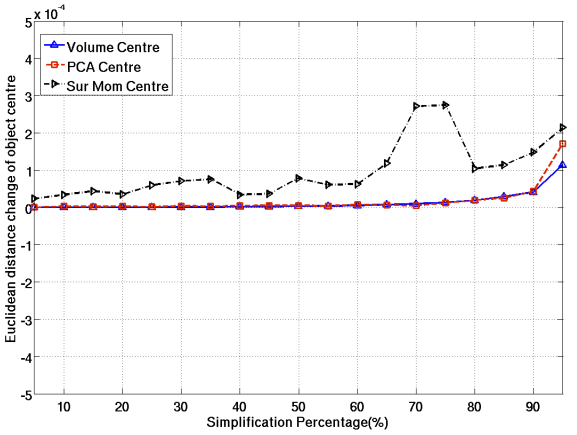
(a) Angle difference of principal axis



(b) Object centre difference



(c) Angle difference of principal axis



(d) Object centre difference

Figure 3.12: Alignment error under noise attack in (a) and (b), simplification attack in (c) and (d).

metric software described in [50], while for the mesh smoothing we employed the Laplacian filter proposed in [119] with a parameter $\lambda = 0.2$ and for 10 iterations. The segmentation is consistent almost perfectly under the simplification and the Laplacian smoothing. Some errors emerge at the leg of the horse under the additive noise attack but most of the segments remain identical to the original ones.

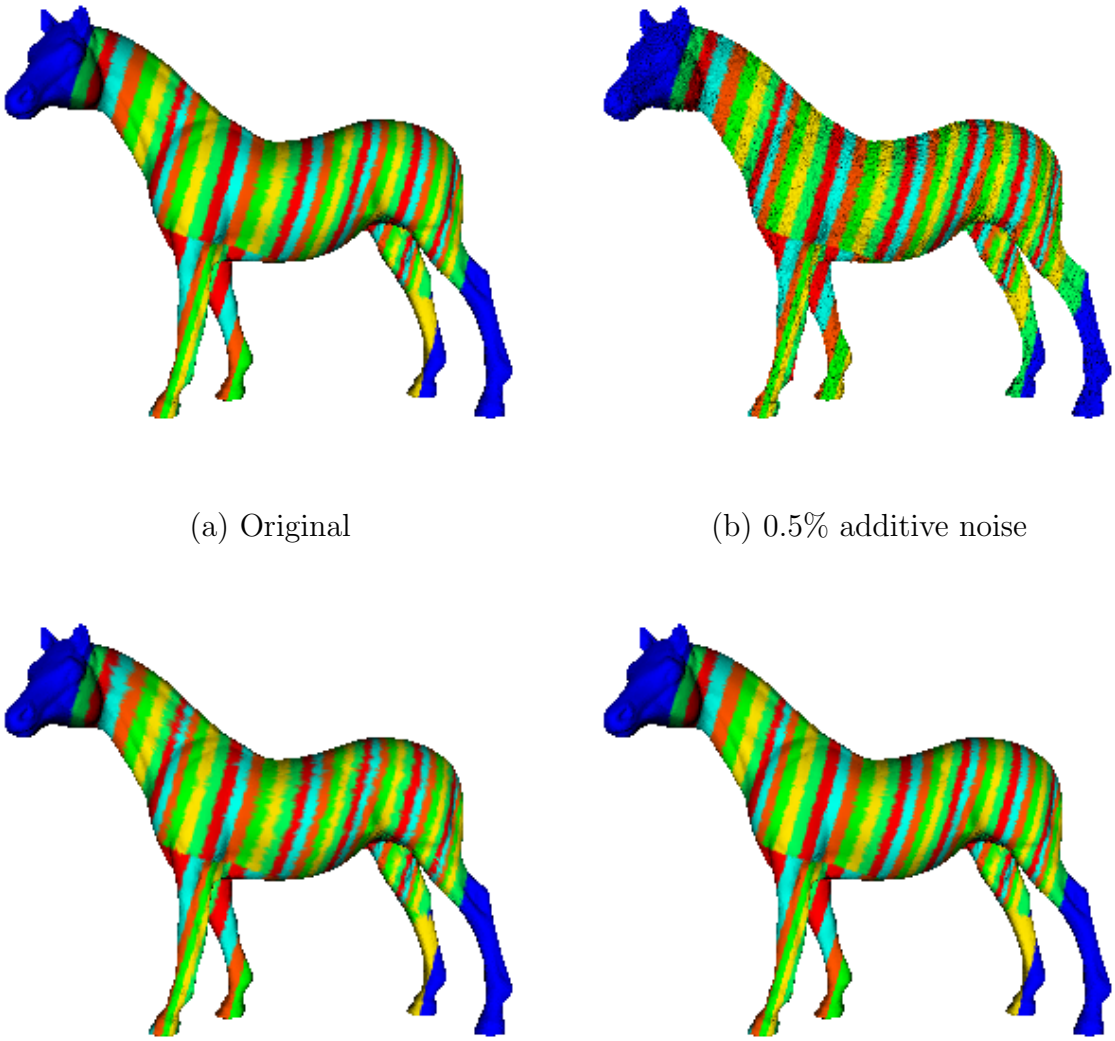
3.7 Conclusion

In this chapter, we proposed a new blind and robust spectral watermarking algorithm for 3D meshes based on the localized patches generation. The algorithm firstly orients the object using the Volumetric Principal Component Analysis. The object

is then split into patches of equal areas along the first and second principal axis. Each patch of the object is spectral decomposed. The spectral coefficients of each patch are treated as point cloud in 3D space. The message is embedded by enforcing a constraint on the ratio between the first principal component and the other two principal components of the point cloud.

The proposed watermarking methodology in the spectral domain is blind and robust and is based on embedding constraints in the distribution of spectral coefficients. Also, we proposed a novel patch generation method that is robust against most of the common mesh attacks. The proposed method does not suffer of the patch overlapping problem as other segmentation methods [98]. Furthermore, the patch generation method is very flexible and can be adjusted according to the object type and to the various requirements of the watermarking system. The security is ensured in several stages including patch generation, patch selection and the coefficients selection.

However, there are a few disadvantages of the watermarking algorithm. Although the proposed method provide competitive results on some cases in the robustness test, the algorithm is not as robust as Cho's method. And the algorithm is not robust against the re-watermarking attack. Also, the algorithm still produces visible distortion on some regions of the object. As a transform domain method, the proposed method is lack of mechanism to control the distortion produced by watermarking. However it still produces less distortion than the Cho's method. Extensive experiments have shown that the proposed method provides a good trade-off between distortion and robustness when compared with state-of-the-art spatial domain methods.



(a) Original

(b) 0.5% additive noise

(c) 50% simplification

(d) Smoothed after 10 iterations $\lambda = 0.2$

Figure 3.13: Robustness of the segmentation method

Chapter 4

Geodesic Watermarking

4.1 Introduction

Although the spectral method shows certain robustness and relatively low visual distortion, this category of methods suffers from the problem of high computational cost. In this chapter, we propose a spatial domain method which use the geodesic distribution to embed the watermark.

3D watermarking algorithms based on spatial domain have shown promising and good results. A watermarking method using distributions of differences of distances between randomly picked vertices and the average of their neighbourhood, both calculated from the principal axis of the object, was proposed in [142]. The distance from the object center to the vertices on its surface is considered as a statistical variable in [31] for a watermark embedding method which extends the approach from [142]. Two statistical methods are proposed in [31] by changing the statistical variable mean or variance. This method was shown to embed watermarks which are robust against most common distortion attacks such as additive noise, smoothing and mesh simplification. Alface *et al* [3] proposed to use robust feature points as watermarking references in order to cope with the stego-object cropping attack. All existing graphics watermarking methods create bump like changes on the surface of objects which are more or less visible.

The geodesic distance, which takes into account the local surface variation, has been shown to be the most appropriate measure for calculating the distance be-

tween two different points on a mesh [110]. The geodesic distance has been used in computer graphics [115] and computer vision [102]. The distribution of geodesic distance has not been investigated in the literature for watermarking 3D graphics. Various approaches have been adopted for calculating the geodesic distance including by using straightest geodesics [103]. However, this method does not guarantee the minimal path for its computation. In [115] the geodesic distance is calculated by considering an acceptable error bound. Fast Marching Method (FMM) was proposed for fast and accurate calculation of geodesic distances between two locations on the object mesh [72, 102, 109]. FMM evaluates geodesic distances with respect to a source location by considering a specific vertex ordering according to the upwind direction. Based on FMM we can change the geodesic distance of one vertex without affecting the upwind vertex distances. This avoids the backward causality problem encountered when considering other measures. Geodesic distances are invariant to translation, rotation and vertex reordering. With regard to uniform scaling, it changes the numerical value of a specific geodesic distance but not the ratio of two geodesic distances, thus this does not affect the geodesic distribution. Normalizing all geodesic distances to the interval $[0, 1]$ solves the uniform scaling problem.

In this chapter we propose a novel statistical 3-D watermarking method by using distributions of geodesic distances calculated from the mesh surface. We identify the location of a source as a reference for the FMM method using the intersection of a key based random vector passing through the object center with its surface. The reference system for aligning the graphical object is defined by using the Volumetric Principal Component Analysis (VPCA). We split the surface of the object into strips of equal geodesic width, calculated by using FMM when considering the source location as reference. Each strip contains vertices which have their geodesic distances to the source in a well defined interval range. Distributions of geodesic distances corresponding to the vertices from each strip are changed by using two histogram mapping methods in order to embed a single bit. A novel Vertex Placement Scheme (VPS) is proposed for moving vertices according to the watermark geodesic distances as required by the corresponding histogram mapping. The vertices are moved on the object surface perpendicularly on the geodesic front lines. The message is embedded

when all vertices comply with the marked geodesic distances. By changing geodesic distances as proposed in this study we preserve the object shape by minimizing the distance from the displaced vertices to the original mesh surface. We provide a set of constraints for the interval of allowable vertex changes. The minimum admissible change is a consequence of the study from [72], while we provide the condition for the maximum admissible change in order to avoid the turn over of mesh triangles.

The rest of the chapter is organized as follows. Section 4.2 introduce the geodesic distance calculation and the FMM method. In Section 4.3 we detail the steps of the proposed methodology corresponding to the initialization, histogram mapping and the vertex placement scheme followed by the mesh distortion assessment and the watermark detection algorithm. Section 4.4 provides the experimental results, while Section 4.5 details the conclusions of this study.

4.2 Geodesic distances on manifolds

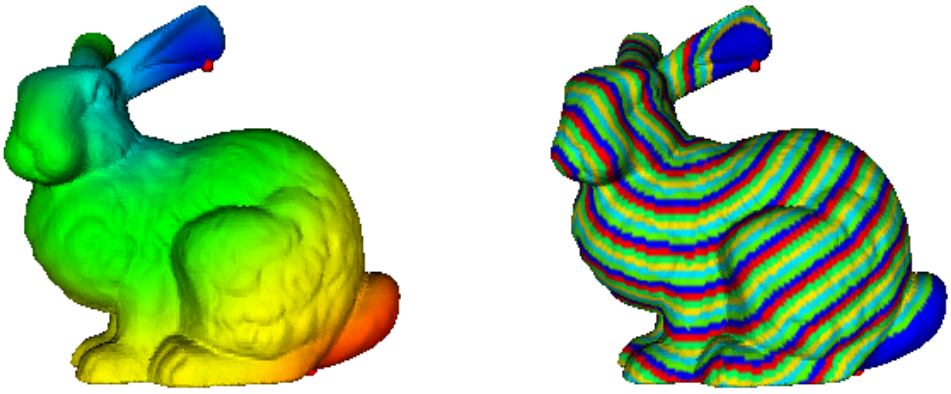
The proposed graphics watermarking methodology consists of splitting the object into regions and marking each region according to the watermark code. In order to ensure an increased robustness to possible attacks and to various graphics processing algorithms we propose to use distance measures which are consistent with the shape representation. Euclidean distance simply calculates the shortest distance between locations in space without considering any specific information about the graphical object. In contrast, the geodesic distance takes into account the shape of the object and calculates distances along the minimal path on the surface of the object [72, 109, 110]. By using geodesic distances we can embed information in the graphical shape by ensuring that the resulting watermarked shape is not distorted when compared to the original shape. In the following we outline a few concepts about calculating geodesic distances on manifolds which are used by the proposed watermarking methodology.

Let us consider $\mathcal{O}$, a graphical object which is represented as a mesh containing vertices $\mathcal{V} = \{\mathbf{v}_i \in \mathcal{O} | i = 1, \ldots, N\}$, where N is the total number of vertices. In the following we assume that the graphical objects are represented as triangulated manifolds. Other types of graphical objects can be converted into triangulated

manifolds. Let us consider a curve $\gamma(t) \in \mathcal{O}$, where $t \in [0, P]$ is a parameter, onto the surface of the object, which joins two different points $\mathbf{x}, \mathbf{y} \in \mathcal{O}$. The points are located on the surface of the graphical object, but they are not necessarily part of its vertex set. There are a multitude of ways for joining the two given points such that the connecting curve is completely contained on the surface of the graphical object. The geodesic distance $T_{\mathcal{O}}(\mathbf{x}, \mathbf{y})$ between the points $\mathbf{x}$ and $\mathbf{y}$ is defined with respect to the minimal length of the geodesic curve $\gamma(t)$ such that :

$$T_{\mathcal{O}}(\mathbf{x}, \mathbf{y}) = \min_{\gamma(t) \in \mathcal{O}} \int_0^P \sqrt{\gamma'(t)^T H(\gamma(t)) \gamma'(t)} dt \tag{4.1}$$

where $\gamma(0) = \mathbf{x}$, $\gamma(P) = \mathbf{y}$, $\gamma'(t)$ represents the local derivative of the parametric curve and $H(\cdot)$ is an intrinsic metric. In this study we consider $H(\cdot) = 1$. A geodesic map calculates the geodesic distances from all points on the object surface to a set of known source points. The geodesic map to a single source point is shown coded using colour on the Bunny object in Figure 4.1(a) where the reference point, representing the source for evaluating the geodesic distances, is indicated by a small red circle. The color varies from blue to red pseudo-coding the geodesic distance from the source point. In the following we consider a single source point $\mathbf{s} \in \mathcal{O}$ and for the sake of simplification we denote $T_{\mathcal{O}}(\mathbf{s}, \mathbf{x}) \equiv T(\mathbf{x})$, where $\mathbf{x} \in \mathcal{O}$ is an arbitrary location on the surface of the graphical object $\mathcal{O}$.



(a) Geodesic map. (b) Iso-geodesic mesh strip generation

Figure 4.1: Geodesic distance, highlighted using pseudo-colour, on the Bunny object.

The key idea for solving equation (4.1) is that the distance function satisfies the intrinsic Eikonal equation, which is a non-linear partial differential equation given by :

$$\|\nabla_{\mathcal{O}} T(\mathbf{s}, \mathbf{x})\| = F(\mathbf{x}), \quad \forall \mathbf{x} \in \mathcal{O} \tag{4.2}$$

such that there is a start location $\mathbf{s}$ with $T(\mathbf{s}) = 0$, where $F(\mathbf{x}) > 0$ and $\|\cdot\|$ represents the norm. Physically, the solution $T(\mathbf{x})$ is the shortest distance from $\mathbf{s}$ to $\mathbf{x}$ using only paths contained in $\mathcal{O}$, where $1/F(\mathbf{x})$ is the propagation speed at location $\mathbf{x}$. We assume $F(\mathbf{x}) = 1$ in the following, *i.e.* the propagation speed is constant all over the mesh. The idea for solving the Eikonal equation is to find an approximation to the gradient term which correctly deals with shape variations including folds and creases [109].

Various solutions have been proposed for calculating the geodesic distance [103, 110, 115]. An approximate solution for solving the eikonal equation (4.2) on triangulated manifolds is provided by the Fast Marching Method (FMM) [72,110]. FMM makes sure that every vertex is updated only once by progressively advancing the front of distance calculation in an upwind direction starting from the source (reference location) $\mathbf{s}$. At any time, when applying this method, the object vertices are split into three sets $\mathcal{D}$, $\mathcal{F}$ and $\mathcal{L}$ such that $\mathcal{D} \cup \mathcal{F} \cup \mathcal{L} = \mathcal{V}$, where $\mathcal{V}$ is the vertex set, and the intersection of any two of these sets is empty. The set $\mathcal{D}$ is used to store the vertex which geodesic distance has been calculated. $\mathcal{F}$ is the front line of the propagation, *i.e.* the geodesic distance of the vertex that is being calculated. $\mathcal{L}$ contains the rest of the vertices. Initially, the source points are labeled as $\mathcal{D} = \{\mathbf{s}\}$. Neighbours of the source points are marked as $\mathcal{F} = \mathcal{N}(\mathbf{s})$. Once a vertex $\mathbf{v}_i \in \mathcal{F}$ has its geodesic distance $T(\mathbf{v}_i)$ calculated, it is moved from set $\mathcal{F}$ to set $\mathcal{D}$. FMM, by using the geodesic distance propagation, as described above, ensures that changing the geodesic distance of a specific vertex will not affect the previously calculated geodesic distances for the other vertices. This property is important in order to ensure that geodesic distance calculation avoids the backward causality problem. The neighbours of $\mathbf{v}_i$ are moved to the front set $\mathcal{F}$, if $\{\mathbf{v}_j | \mathbf{v}_j \in \mathcal{N}(\mathbf{v}_i), \mathbf{v}_i \in \mathcal{D}, \mathbf{v}_j \notin \mathcal{D}\}$. All the other vertices are part of the third set, $\mathcal{L}$. The vertices from $\mathcal{F}$ define the propagation front line and they are ordered according to their geodesic distance to

the source point. One by one, in the order of their increasing geodesic distance to the source location, these vertices are added to the set $\mathcal{D}$. When the geodesic distance is calculated for a vertex $\mathbf{v}_j \in \mathcal{F}$, this should not change any geodesic distance $T(\mathbf{v}_i)$ already calculated for the vertices from the set $\mathcal{D}$. Only vertices from the set $\mathcal{L}$ are considered each time for inclusion in the set $\mathcal{F}$ and only vertices from $\mathcal{F}$ have their geodesic distance calculated. The procedure continues as a propagation wave until $\mathcal{L}$ and $\mathcal{F}$ are both empty. The details of how to calculate the geodesic distance can be found in section 4.3.4. The computational complexity of this algorithm is $O(N \log\ N)$, where N is the number of vertices. The local shape characteristics are intrinsically considered in the calculation of propagating fronts using geodesic distances in the FMM. In the following section we consider FMM for developing a new 3-D watermarking method which ensures that the presence of the watermark in the graphical object will be hidden.

4.3 The geodesic front propagation watermarking method

The proposed watermark embedding method has the following steps: finding the source location, segmenting the object surface into strips, mapping of geodesic distance histograms and the vertex placement scheme for watermark embedding.

4.3.1 Defining the source location

As described in Section 4.2 the FMM method requires a source location $\mathbf{s}$ which is considered as reference for calculating the geodesic distances. The source location should be defined robustly, such that we will be able to identify the same position even after the graphical object has suffered certain changes (such as those performed by an attacker with the intention to destroy the watermark). In general, there are two possible approaches to generate the starting source point. The first way is to find the starting position as the intersection between a direction cast from a reference point, *i.e.* the object centre, to the mesh surface. The direction is generated according to a secret key. Alternatively, the starting point can be chosen

by using a robust feature point detection algorithm such as in [3]. The direction of the intersection scheme is generated according to a secret key such that it is very hard for a malicious attacker to find the starting point without knowing the secret key thus increasing the security of the technique. However, this scheme relies on the robust location of the object center as well as on the principal axis orientation required for the object alignment with the watermark coordinate system. Any attack that affects the object center and the principal axis orientation may lead to errors in the watermark detection stage. On the other hand a feature point location method may have high computational complexity requirements while lacking security since an attacker can easily guess such source locations. In the following we describe a blind method for defining the source location using graphical object moments.

We use the same volume moment scheme detailed in Section 3.2 to align the object. A direction $\overrightarrow{\mathbf{\Gamma}}$ is casted from the object centre μ (shown in equation 3.3) according to a secret key. We use the secret key as a seed, and then generate a random 3D vector based on this seed. The starting point is defined as the intersection between the direction of this vector from the object center μ and the mesh surface as:

$$\mathbf{s} = \{\mu + t\ \overrightarrow{\mathbf{\Gamma}} \cap \mathcal{O}\} \tag{4.3}$$

where $t \in I\!R$ is a parameter scalar. There are two extreme cases corresponding to when there is no intersection with the object surface and when there are multiple intersections. In the former case we proceed to generate additional directions until an intersection with the object surface is found. In the latter case the intersection which is the furthest away from the object center μ is chosen as the source location.

4.3.2 Iso-geodesic mesh strip generation

Let us define $T_{min} = \min(\{T(\mathbf{x}), \forall \mathbf{x} \in \mathcal{O}\})$ and $T_{max} = \max(\{T(\mathbf{x}), \forall \mathbf{x} \in \mathcal{O}\})$ as the minimum and maximum geodesic distances calculated for the object $\mathcal{O}$ from a source location $\mathbf{s}$. Generally, the number of vertices whose geodesic distance is close to T_{min} and T_{max} would be too small in order to be statistically relevant so they are not considered for watermark embedding. Therefore, we trim the range of

acceptable geodesic distances to the range :

$$T(\mathbf{v}_j) \subset ((1-\varepsilon)T_{min} + \varepsilon T_{max}, \varepsilon T_{min} + (1-\varepsilon)T_{max}) \tag{4.4}$$

where $\varepsilon \in (0, 0.2)$ characterizes vertices which are close to extremes, according to their geodesic distance from the source location, in order to be excluded from further watermark embedding calculations. Then, for a watermark code of M bits, the object mesh is segmented into M strips, each used for embedding a single bit. Consequently, the geodesic distance width for each strip is defined as :

$$T_b = \frac{(1-2\varepsilon)\ (T_{max} - T_{min})}{M}. \tag{4.5}$$

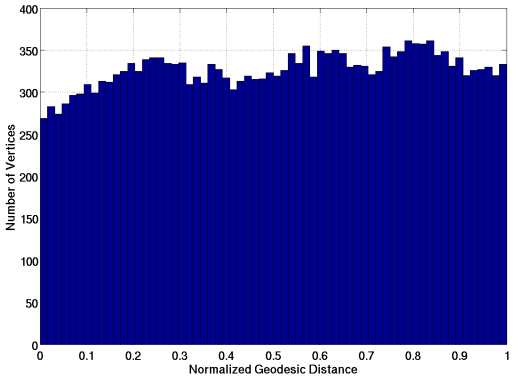
Let us consider $\mathcal{B}_i$ as the set of vertices which are located in a specific range of geodesic distances calculated from the source location $\mathbf{s}$ and characterizing a mesh strip on the object surface :

$$\mathcal{B}_i = \{\mathbf{v}_j \in \mathcal{O} \mid T_{min} + (i-1)T_b \leq T(\mathbf{v}_j) < T_{min} + iT_b\} \tag{4.6}$$

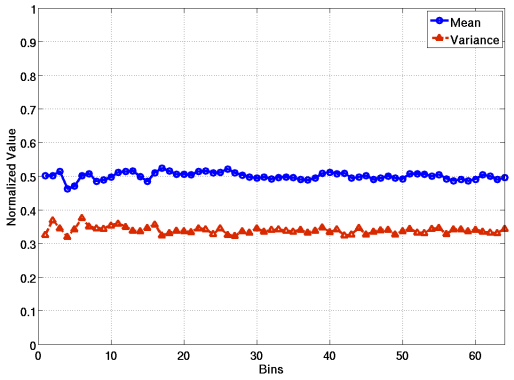
for $i = 1, \ldots, M$. T_b should be large enough in order to define regions which contain a statistically consistent number of vertices available for watermark embedding. Figure 4.1(b) shows how the graphical object Bunny is split into strip regions. The region around the location of the source, which is trimmed away, is pseudo-colored with blue, and each strip is used for embedding a single bit.

An interesting observation is that the geodesic distance distributions corresponding to vertices contained in each strip are close to uniform which means that the expected mean and variance are around 0.5 and 1/3, respectively. Figure 4.2(a) displays a typical histogram of geodesic distances of the vertices on the Bunny object. Figure 4.2(b) shows the mean and variance values corresponding to each bin of the Bunny object segmented into 64 regions as described in the previous sections. As can be observed from this plot, the mean and variance value of most of the bins conforms to the assumptions of uniform distributions of geodesic distances. These properties have been observed for a large category of objects when assuming segmentation into iso-geodesic strips containing a statistically significant number of

vertices.



(a) Histogram of geodesic distances for vertices.



(b) Mean and variance of geodesic distances from each iso-geodesic strip.

Figure 4.2: Statistics of geodesic distances from strips extracted from the Bunny object when segmented into 64 regions.

4.3.3 Statistical watermarking by histogram mapping

After splitting the graphical object into strips of equal geodesic width according to equations (4.5) and (4.6), each strip is associated with a bit from the watermark code. The distributions of geodesic distances for vertices in each strip are close to uniform and consequently they are considered as a random variable suitable to be used for embedding messages by introducing specific histogram asymmetries [31, 142].

In the proposed method we embed a bit into each iso-geodesic distance strip. Let us define the statistical variable representing the geodesic distance from a vertex to the source location $\mathbf{g}_i = \{g_{ij} = T(\mathbf{v}_j), \forall \mathbf{v}_j \in \mathcal{B}_i\}$. We record the minimum and maximum geodesic distances within each strip, *i.e.* $T_{i,min} = T_{min} + (i-1)T_b$ and $T_{i,max} = T_{min} + iT_b$, in order to be used later for the inverse normalization. The statistical variable g_{ij} is firstly normalized to the range $[0, 1]$ by:

$$\tilde{g}_{ij} = \frac{g_{ij} - T_{i,min}}{T_{i,max} - T_{i,min}} \tag{4.7}$$

where g_{ij} and $\tilde{g}_{ij}$ are the jth elements of $\mathbf{g}_i$ and $\tilde{\mathbf{g}}_i$, respectively.

In the following we consider two histogram mapping functions for embedding information as in [31]. The first histogram mapping function changes the mean value

of the statistical variable $\tilde{g}_{ij}$ which corresponds to the vertex $\mathbf{v}_j \in \mathcal{B}_i$. Assuming that g_{ij} is uniformly distributed, the expected mean value of $\tilde{\mathbf{g}}_i$ is 1/2. In order to embed one bit of message, the mean value of $\tilde{\mathbf{g}}_i$ is changed as follows:

$$\hat{\mu}_i = \begin{cases} \frac{1}{2} + \alpha & \text{if } B_i = 1 \\ \frac{1}{2} - \alpha & \text{if } B_i = 0 \end{cases} \tag{4.8}$$

where α is the watermark strength factor, influencing the visual distortion and robustness, and B_i, for $i = 1, \ldots, M$, is the bit to be embedded in the ith mesh strip.

In order to fulfill the relationship from (4.8) the first histogram mapping function implements the following :

$$\tilde{g}'_{ij} = \tilde{g}_{ij}^{\beta} \quad \begin{cases} \beta \in (0,1) & \text{if } B_i = 1 \\ \beta \in (1,\infty) & \text{if } B_i = 0 \end{cases} \tag{4.9}$$

where $\tilde{g}'_{ij}$ is the resulting histogram mapping variable corresponding to $\tilde{g}_{ij}$ and $\beta > 0$ is a parameter which models its shape and which depends on the watermark strength α [31]. Finally, the watermarked geodesic distance $\hat{g}_{ij}$ is obtained by mapping $\tilde{g}'_{ij}$ back to the original interval as :

$$\hat{g}_{ij} = \tilde{g}'_{ij}(T_{i,max} - T_{i,min}) + T_{i,min} \tag{4.10}$$

The second embedding method changes the variance of the statistical variable g_{ij}. In this case, g_{ij} is normalized to the range $[-1, 1]$:

$$\tilde{g}_{ij} = 2 \cdot \frac{g_{ij} - T_{i,min}}{T_{i,max} - T_{i,min}} - 1 \tag{4.11}$$

The expected variance of variable $\tilde{g}_{ij}$ is 1/3 for a uniform distribution. We embed one bit of message by modifying the variance of $\tilde{g}_{ij}$ according to :

$$\hat{\sigma}_i^2 = \begin{cases} \frac{1}{3} + \alpha & \text{if } B_i = 1 \\ \frac{1}{3} - \alpha & \text{if } B_i = 0 \end{cases} \tag{4.12}$$

The second histogram mapping function for modifying each element from the set $\mathcal{B}_i$

is defined as :

$$\tilde{g}''_{ij} = sign(\tilde{g}_{ij})|\tilde{g}_{ij}|^{\beta} \quad \begin{cases} \beta \in (0,1) & \text{if } B_i = 1 \\ \beta \in (1,\infty) & \text{if } B_i = 0 \end{cases} \tag{4.13}$$

Accordingly, the watermarked geodesic distances are obtained by the inverse normalization function:

$$\hat{g}_{ij} = \frac{1}{2} \cdot (\tilde{g}''_{ij} + 1) \cdot (T_{i,max} - T_{i,min}) + T_{i,min} \tag{4.14}$$

The details of how to calculate β in equation (4.9) and (4.13) given the value of α in equation (4.8) and (4.12) can be found in Cho's paper [31].

A crucial requirement for graphics watermarking is to produce undetectable changes in the object surface. In the following section we describe and discuss a new watermark embedding method called vertex placement scheme. The proposed method changes the location of vertices inside each iso-geodesic strip such that their geodesic distances correspond to the watermarked distributions obtained as described above.

4.3.4 Changing vertex geodesic distances by vertex placement scheme

The previous section described the modalities of mapping histograms of geodesic distances corresponding to each surface strip, by using equation (4.8) when shifting the mean or (4.12) when changing the variance, in order to embed one bit. Various 3-D watermarking methods embed information in the 3-D structure of the object by introducing bias in the statistical measures characterizing the local object symmetry according to the bit to be embedded. Such approaches produce changes with respect to a predefined reference location usually resulting in bump like distortions on the object surface. In this chapter we propose to watermark the graphical object by displacing the vertex locations on its surface, along the direction of the FMM propagation front obtained as explained in Section 4.2. In this approach we compute the solution to the Eikonal equations (4.2), resulting in a piecewise linear approximation to the geodesic function.

In the following we describe how to displace vertices in order to conform with the distributions of the watermarked geodesic distance variables from (4.9) or (4.13) while not visibly perturbing the surface. Let us consider the framework from [72] by using similar notations and figures. The proposed watermark embedding procedure for a particular triangle $\triangle ABC$, $A, B, C \in \mathcal{V}$ is called the vertex placement scheme (VPS). The study can be easily extended for all the vertices inside the strip $\mathcal{B}_i$ and to the entire object $\mathcal{O}$. Let us consider the vertices A and B as having their geodesic distances $T(A)$ and $T(B)$, $T(A) < T(B)$. When the angle $\angle C = \theta$ inside triangle $\triangle ABC$ is acute then the update scheme is monotone, *i.e.* $T(A) < T(B) < T(C)$. Let us assume that the lengths of the triangle sides are $a = \|BC\|$, $b = \|AC\|$ and $c = \|AB\|$ as shown in Figure 4.3(a) and denote the geodesic distances between its vertices, calculated along the front propagated with respect to the source location $\mathbf{s}$, as :

$$t = T(C) - T(A) \tag{4.15}$$

$$u = T(B) - T(A) \tag{4.16}$$

$$h = T(C) - T(B) = t - u \tag{4.17}$$

Kimmel and Sethian have shown in [72] that the value of t can be calculated using FMM by assuming known $T(A)$, $T(B)$ and the geometry of $\triangle ABC$, according to the equation :

$$(a^2 + b^2 - 2ab\cos\ \theta)t^2 + 2bu(a\cos\ \theta - b)t + b^2(u^2 - a^2\sin\ \theta) = 0 \tag{4.18}$$

The solution t must satisfy two conditions:

$$\begin{cases} u < t \\ a\cos\theta < \frac{b(t-u)}{t} < \frac{a}{\cos\theta} \end{cases} \tag{4.19}$$

$u < t$ means that $T(C) > T(B) > T(A)$ which conforms to the monotone property. The second condition of equation (4.19) means that $T(C)$ must be updated from

within $\triangle ABC$. Thus, the complete update procedure is given as:

$$T(C) = \begin{cases} min\{T(C), t + T(A)\} & \text{if conditions (4.19) are fulfilled} \\ min\{T(C), b + T(A), a + T(B)\} & \text{otherwise} \end{cases} \tag{4.20}$$

The Fast Marching Method does not calculate exact solution of geodesic distance. When condition (4.19) is not satisfied, the geodesic distance will be approximated locally according to the length of the edge.

An extreme situation for t as solution from equation (4.18) is when $\theta = \pi/2$. In this situation we have the minimum bound for t, constraining the vertex placement location, as :

$$t_{min} = \frac{b^2 u + ab\sqrt{a^2 + b^2 - u^2}}{a^2 + b^2} \tag{4.21}$$

When θ is obtuse, triangles are unfolded and split into two acute angle triangles and then we can proceed with the updating scheme. For $\triangle ABC$ whose $\angle C$ is obtuse, we follow the triangle unfold procedure as detailed in [72] and split $\angle C$ into two acute angles, and then update the vertex by using the scheme described above.

In the following we show that changes of geodesic distances, according to the proposed VPS, can be embedded into the FMM method by ensuring a minimal change along the geodesic front. Let us assume that in the case of $\triangle ABC$ we have A and B fixed while C changes to C' following watermarking by VPS. Assuming that $\{A, B\} \in \mathcal{D}$, the watermark embedding is performed along the geodesic front vertices $C \in \mathcal{F}$. We associate the statistical variables g and $\hat{g}$, as derived according to the histogram mapping functions from Section 4.3.3, after dropping their location parameters i, j for the sake of convenience, to the geodesic distance $T(C) = g$ and to that of a new location C', which would result after watermark embedding. The problem addressed in the following is about how to move the vertex C to a new location C', such that its new geodesic distance satisfies $T(C') = \hat{g}$, while ensuring that the graphical object suffers a minimal distortion. The proposed VPS consists of the following sequence of steps for vertices from a segmented strip $\mathcal{B}_i$:

1. Calculate $\hat{g}$ from either (4.9) or (4.13).

2. Choose a vertex $C \in \mathcal{B}_i$ in the downwind direction of FMM, calculate $T(C)$.

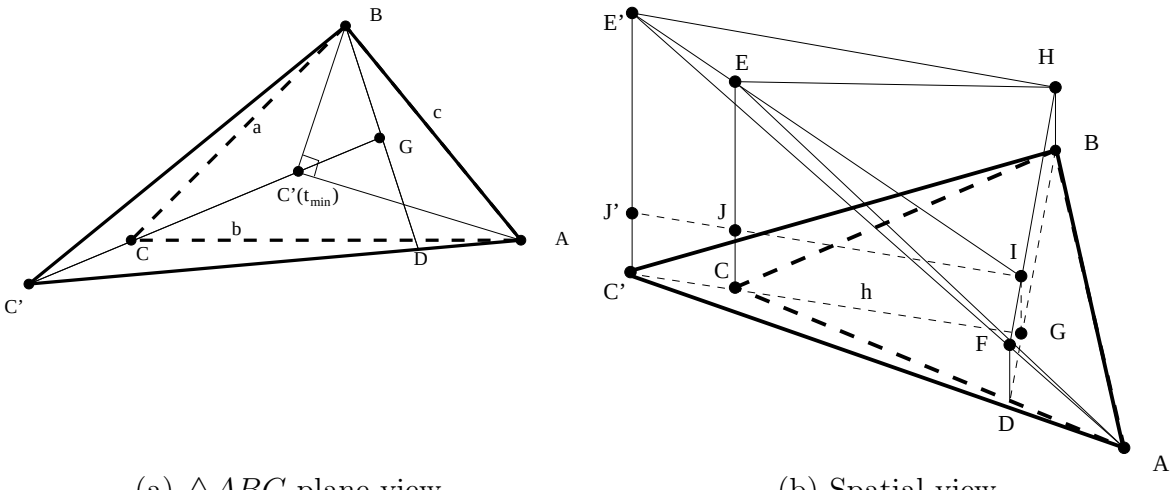
3. While $T(C) \neq \hat{g}$ and less than 50 iterations, repeat steps (4) - (6).

4. Locate $A, B \in \mathcal{N}(C)$ such that all three form a triangle which contributes to the minimum path calculation for $T(C)$.

5. Apply VPS in $\triangle ABC$ to move C to C' such that $T(C') = \hat{g}$.

6. Set $C = C'$ and update the geodesic distance $T(C)$ using only vertices from set $\mathcal{D}$ and go to step (3).

We have shown the details step (1) and (2) already. Now we demonstrate how to move the vertex C to a new position C' from a $\triangle ABC$ such that its new geodesic distance $T(C') = \hat{g}$. Similarly with Figure 5 from [72], depicting geodesic distance calculation, we illustrate the VPS procedure in Figure 4.3. In the case when the conditions from (4.19) are fulfilled, there is a point G inside $\triangle ABC$ such that $CG \perp BG$ and the Euclidean distance $\|CG\| = h$, with h defined in equation (4.17). Replacing h with $\|CG\|$ in equation (4.17) and expanding into geodesic distances we observe that $T(G) = T(B)$. BG is the approximation of the equal geodesic curve located at the distance $T(B)$ from the source location $\mathbf{s}$. In the following we describe the VPS procedure which transforms $\triangle ABC$ into $\triangle ABC'$ such that $T(C') = \hat{g}$, where $\hat{g}$ was calculated in Section 4.3.3, as shown in Figure 4.3.



(a) $\triangle ABC$ plane view (b) Spatial view

Figure 4.3: Illustration of the updating procedure from C to C'.

We define the following theorem for characterizing the vertex displacement for geodesic distance watermark embedding :

Theorem 1. *For $\triangle ABC$, we can find a new point C' on the line GC such that $\|CC'\| = |\hat{g} - T(C)|$, with $\hat{g} \geq t_{min} + T(A)$, where t_{min} is provided in (4.21), and assuming that the conditions from (4.19) are fulfilled for both C and C', then $T(C') = \hat{g}$.*

Proof. We can observe that for any point $C' \in CG$ we have $C'G \perp BG$, where BG is the geodesic front line calculated from the source location. We have the following geodesic distance for C' calculated from the source location :

$$T(C') = \|C'G\| + T(B) = \|CG\| \pm \|CC'\| + T(B) \tag{4.22}$$

where we use the fact that $\{G, C, C'\}$ are collinear and where the sign before $\|CC'\|$ is "+" if $T(C') \geq T(C)$ and "-" otherwise. This results into the following :

$$T(C') = T(C) - T(B) \pm |\hat{g} - T(C)| + T(B) \tag{4.23}$$

where we consider that $T(B) = T(G)$ and from the first relationship of (4.19) that $T(C) > T(B)$. Then we have :

$$T(C') = T(C) + \hat{g} - T(C) = \hat{g} \tag{4.24}$$

The watermark updating by means of VPS is visualized in Figure 4.3(a). In Figure 4.3(b) we show a perspective view of the geodesic distance propagation. In this figure we have $\{HB, IG, EC, FD\} \perp \triangle ABC$ and as in [72] we consider $\|HB\| = \|FD\| = \|IG\| = u$ while $\|EC\| = t$. We also consider $J \in EC$ such that $\|JC\| = u$ resulting in $\|JI\| = h$. From $\triangle EIJ$ we can observe that the slope of the plane containing $\triangle EHF$ and that of $\triangle ABC$ is equal to $\tan(\angle(EIJ)) = \frac{t-u}{h}$ which represents the geodesic propagation speed. In Figure 4.3(b), C' is located in the plane of $\triangle ABC$ while we consider E' such that $E'C' \perp \triangle ABC'$ and $\|C'E'\| = \hat{t} = T(C') - T(A)$. We consider $J' \in E'C'$ such that $\|J'C'\| = u$ resulting into $\|J'I\| = \hat{h}$, corresponding to the watermarked vertex following VPS. We can observe that E' is located in the plane of $\triangle EFH$ and by using the similarity of the triangles $\triangle IEJ$ and $\triangle IE'J'$ we

obtain :

$$\frac{\hat{t}-u}{\hat{h}} = \frac{t-u}{h} \tag{4.25}$$

which proves that C' is characterized by the same geodesic propagation speed as C. □

Note that the theorem is valid only when the geodesic distance $T(C')$ is still calculated from within the $\triangle ABC'$ after updating C to C'. For a single vertex C' there may be other neighbours $U, V \in \mathcal{N}(C')$ such that $U, V \in \mathcal{D}$ which can be used for calculating the geodesic distance. According to the definition from equation (4.1), the geodesic distance represents the shortest length on the manifold, calculated on the object surface. The final geodesic distance of a vertex is calculated by using at most two neighbours (*i.e.* forming a triangle such as $\triangle UVC'$). If $T(C')$ calculated from the $\triangle UVC'$ is less than the value calculated from $\triangle ABC'$, then $T(C')$ is not equal to $\hat{g}$. In this case, the vertex C' need to be updated again in order to satisfy the actual geodesic distance $T(C') = \hat{g}$ *i.e.* go back to step (3).

Theorem 1 indicates that within each iteration in a triangle $\triangle ABC$, a vertex is moved with the Euclidean distance of $|\hat{g} - T(C)|$ in order to ensure the watermark embedding, where $T(C)$ means the actual geodesic distance calculated within that iteration. $T(C)$ is initially equal to g in the first iteration but not the latter ones. Thus, the theorem states that if the vertex satisfies the watermark condition by just one move, then the Euclidean movement is $|\hat{g} - g|$. The vertex is moved with Euclidean distance $|\hat{g} - T(C)|$ in every iteration but the total movement may not be equal to the $|\hat{g} - g|$. But the movement within each iteration is guaranteed to be minimum as proved by Theorem 1. In some extreme cases, the vertex may be in an infinite loop as it is moving back and forth. We avoid this case by stopping the loop when the iteration is beyond a certain limit *i.e.* 50 iterations in our implementation. Because our algorithm is a statistical method, the nonconvergent vertices will not affect the robustness of the algorithm as long as they are not so many.

If $\hat{g} < t_{min} + T(A)$, where t_{min} is provided in (4.21), then the condition required by Theorem 1 is not fulfilled and will result into an angle $\angle C'$ which is obtuse as can be observed from Figure 4.3(a). In this case we choose C' such that $\|GC'\| = t_{min} - u + \epsilon$ where ϵ is a small value and we reassess the updating statistics for the rest of

geodesic distances $\hat{g}$. The limit situation of watermark embedding corresponding to the condition $\hat{g} = t_{min} + T(A)$ is represented in Figure 4.3(a) for $\angle AC'B = \pi/2$.

In the case when the conditions from (4.19) are not fulfilled, it means that there is no point G inside $\triangle ABC'$ such that $T(B) = T(G)$ [72] and we use the second equation from (4.20) for calculating $T(C')$. In this case, the updating scheme consists of:

$$\begin{cases} \text{Find } C' \text{ along } BC \text{ so that } \|BC'\| = \hat{h} & \text{if } T(B) + a < T(A) + b \\ \text{Find } C' \text{ along } AC \text{ so that } \|AC'\| = \hat{h} + u & \text{else} \end{cases} \tag{4.26}$$

The proof of Theorem 1 from above is still valid in this case by substituting G with A or B respectively.

4.3.5 Assessing the mesh distortion caused by VPS

Let us consider a non-flat surface, while trying to embed a large geodesic distance. Such a situation is shown in Figure 4.4 when embedding a watermark by mapping C into C', while considering $\triangle ABC$ as the base for calculating the geodesic distances. We can see that for a certain watermark mapping, $T(C') - T(C) = |\hat{g} - g|$, the triangle $\triangle CDE$ which follows $\triangle ABC$ may be turned over. So, in order to avoid such effects we should limit the amount of change in the geodesic distance when embedding the watermark. In the following we imagine the vertex C is changed into C' along a direction which is perpendicular to the geodesic front line defined by GC. Let us consider the plane α such that $GC' \in \alpha$ and $DE \perp \alpha$. We assume that $\alpha \cap DE = \{H\}$ and that there is a location $I \in \alpha$ such that I is contained in the polygon following $\triangle CDE$ and adjacent to DE, with $I \neq H$. Following that $DE \perp \alpha$ we have $CH \perp DE$, $C'H \perp DE$ and $IH \perp DE$. Under these circumstances we can define the following angles: $\angle CHI = \theta$ and $\angle C'HI = \psi$ between the plane α and the one defined by $\{D, E, C'\}$ after watermarking, respectively, with the plane formed by $\{D, E, I\}$. Let us denote the distance $\|CH\| = d$ and the angle between GC and CH as $\angle GCH = \phi$. It can be observed that the angle θ becomes ψ after watermark embedding, representing the angle defining the turn over for $\triangle C'DE$.

From the law of sines in $\triangle CC'H$ we obtain the following relationship :

$$|\hat{g} - T(C)| = d\frac{\sin(\theta - \psi)}{\sin(\psi + \phi - \theta)} \tag{4.27}$$

A condition for avoiding or limiting the turn over is imposed on the value of $\psi_{min} = \min(\psi)$ and this would result into a maximal admissible geodesic distance following the watermark embedding such that $T(C') = \hat{g}_{max}$, imposing restrictions on the amount of displacement caused by the watermark. Finally, a consistency check condition which was used for mesh simplification algorithms [50] can be used in order to prevent large distortions. Such a condition requires that a displaced vertex C' should lie inside the convex hull determined by planes perpendicular on the object surface which contain the edges that form the first ring neighbourhood of C', such as AB, AE, ED, BD in Figure 4.4. A vertex breaking this limit, following the watermark embedding, will cause the normal flip artifact in the watermarked object which would appear as a black triangle on the object surface. The conditions of minimum and maximum admissible distortions following watermark embedding are given by the conditions $|\hat{g} - T(C)| \geq t_{min} + T(A) - T(C)$ from Theorem 1 and by equation (4.27), respectively.

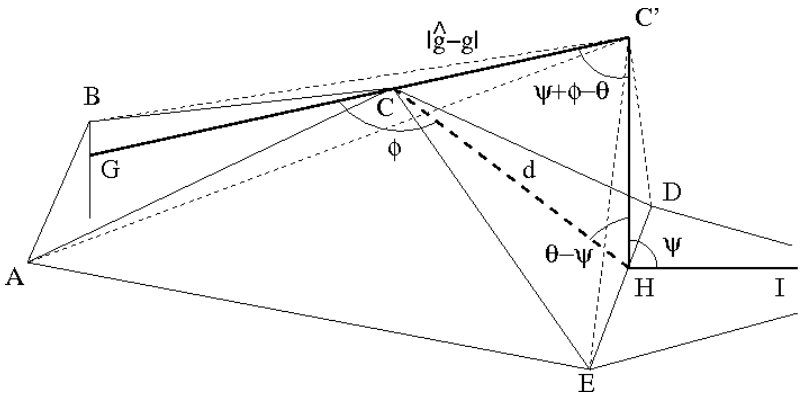
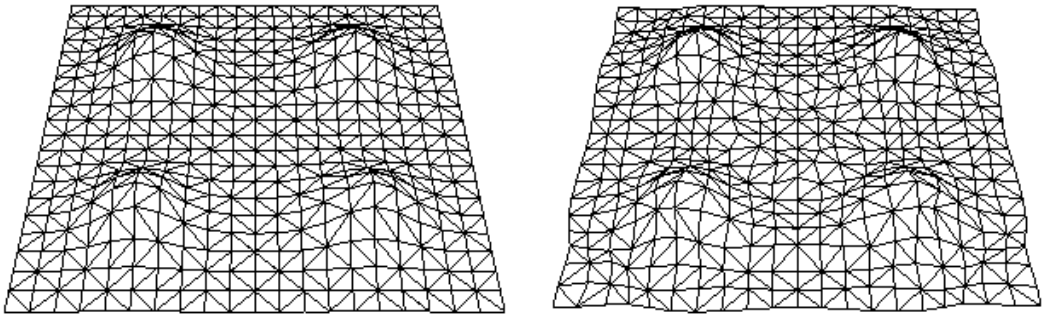


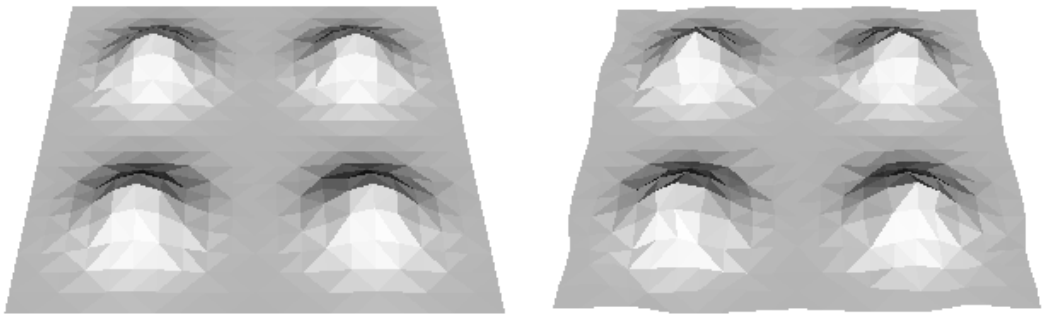
Figure 4.4: Assessing the triangle flip distortion when embedding the maximum vertex displacement.

In Figure 4.5(a) we show a simple mesh surface while in Figure 4.5(b) the same mesh is represented after watermark embedding using the vertex placement scheme when assuming a single source point in the center. We can observe that several vertices are changed on the mesh according to the embedding procedure described

above. It can be easily observed that the vertices are redistributed resulting in circular like effects on the mesh which are concentric in the centre of the object. However, after flat shading as shown in Figures 4.5(c) and 4.5(d), the distortions are no longer visible, excepting for the ripples at the boundary of the object. This shows that the mesh surface is very little affected when embedding watermarks by using the proposed VPS watermark embedding method.



(a) Original mesh (b) Watermarked mesh



(c) Shaded original mesh (d) Shaded watermarked mesh

Figure 4.5: Original and marked mesh surfaces using grid representations in (a), (b), and after flat shading in (c),(d).

4.3.6 Watermark extraction

The watermark extraction algorithm is blind, *i.e.* it does not need the cover object in the detection stage but only the knowledge of the number of embedded bits which can be predetermined. The detector needs to know the secret key, the watermarking algorithm and the number of bits that was embedded in order to recover the watermark message. As a statistical method, it is impossible to generate the bins without knowing the length of the message. For extracting the watermark we use the same procedures as for detecting the source location and generating the isogeodesic mesh strips as described in Sections 4.3.1 and 4.3.2. Histograms of local geodesic distances are formed for each strip. Statistical tests on the data from the histograms are used to detect the embedded information. For the first histogram mapping method, the average of the geodesic distances for vertices contained in the mesh strip $\mathcal{B}_i$, $i = 1, \ldots, M$ is calculated and compared with 1/2 :

$$\begin{cases} \text{if } \hat{\mu}_i > \frac{1}{2} & \text{then } B_i{=}1 \\ \text{if } \hat{\mu}_i < \frac{1}{2} & \text{then } B_i{=}0 \end{cases} \tag{4.28}$$

For the second histogram mapping method, the variance is calculated and compared with 1/3 :

$$\begin{cases} \text{if } \hat{\sigma}_i > \frac{1}{3} & \text{then } B_i{=}1 \\ \text{if } \hat{\sigma}_i < \frac{1}{3} & \text{then } B_i{=}0 \end{cases} \tag{4.29}$$

4.4 Experimental results

4.4.1 Experimental 3D Models

The proposed statistical watermarking methodology described in Section 4.3 was applied on several 3-D graphical objects represented as meshes. In the following we provide the results when watermarking a set of five objects: Bunny, Head, Statue, Dragon and Fandisk. These objects are displayed in Figure 4.6 and their mesh characteristics are provided in Table 4.1. It can be observed that the selected objects provide a diversity of shapes and of mesh characteristics. While Statue is an elongated object, Bunny has many round surfaces, Fandisk is a computer aided design

(CAD) object represented with few vertices and by using large flat surfaces, Dragon is a complex object, with many faces, displaying a high variation on its surface.

Object	No. Vertices	No. Faces
Bunny	34,835	69,666
Head	134,345	268,686
Statue	187,638	375,272
Dragon	422,335	844,886
Fandisk	6,475	12,946

Table 4.1: Characteristics of the graphical objects used in the study.

4.4.2 Distortion evaluation

We use the abbreviation ProMean for the method which embeds watermarks by changing the mean of the distribution of geodesic distances, according to equation (4.9) and ProVar when watermarking is performed by changing the variance of the geodesic distance distribution by using equation (4.13). The vertex placement scheme (VPS) method is used for replacing each vertex from the set $\mathcal{B}_i$, contained in an iso-geodesic strip, with the vertex corresponding to the watermarked distributions of geodesic distances. Before the segmentation into iso-geodesic strips the graphical objects are trimmed by considering $\varepsilon = 0.1$ in (4.4). The proposed methodology is compared with the graphics watermarking methods proposed in [31] which are called ChoMean and ChoVar for changing the mean or variance of distributions of distances from the object center to its vertices. Both methods use the same embedding and detection functions, as described in Section 4.3.3 and the same detection criteria have been considered for all methods.

We use the $E(\mathcal{O}, \hat{\mathcal{O}})$ to evaluate the objective surface distortion as introduced in Section 2.4.1. Table 4.2, provides the distortion results, measured by $E(\mathcal{O}, \hat{\mathcal{O}})$ from (2.16) for all four methods when watermarking the set of objects from Figure 4.6. The models are watermarked with 64 bits of message. As can be observed from Table 4.2, the graphical object distortion introduced by the proposed watermarking methodology is much lower than that produced by ChoMean and ChoVar methods [31]. Most existing 3-D watermarking methods introduce bump-like distortions on the surface of graphical objects. Figure 4.7 displays the visual effects of the proposed

watermarking methods using zoomed views of graphical object surfaces. As can be observed from these figures, the methods from [31] introduce visible staircase artifacts. In the methodology proposed in this chapter, vertices are moved in the plane of the triangle containing the current geodesic front line and perpendicularly on that front line. Thus, hardly any distortion can be observed in the watermarked objects when using either ProMean or ProVar methods. The error analysis from Section 4.3.5 leads to the derivation of a maximum vertex displacement distance when using the VPS algorithm, as provided by equation (4.27), in order to avoid the triangle flip error. If triangles would be flipped over by the VPS embedding, then the surface normal orientation would be reversed with respect to that of the most likely light source direction. This would cause those triangles to become back-facing from front-facing the light source leading to black triangles on the object surface due to the inappropriate interaction with the scene lighting. This problem was analyzed in Section 4.3.5 and such errors are avoided as can be observed in the watermarked graphical objects. However, the proposed methods do not consider preserving features which represent sharp changes on the object surface and some distortions are visible on the edges of the watermarked Fandisk object. As described in Section 4.3.4 and as can be observed from Figure 4.7, the proposed methodology preserves very well the original mesh after watermarking in most cases.

Object	ProMean	ProVar	ChoMean	ChoVar
Bunny	0.37	0.24	0.81	0.39
Head	0.22	0.10	0.37	0.19
Statue	0.35	0.22	0.94	0.40
Dragon	0.36	0.24	0.90	0.47
Fandisk	11.10	8.20	22.90	12.80

Table 4.2: Watermarked object distortion with respect to the original object, calculated as $E(\mathcal{O}, \hat{\mathcal{O}})$, where all results should be multiplied with 10^{-4}.

In order to visualize the error distribution, in Figures 4.8(a) and 4.8(b) we represent the local distortion measured by $E(\mathcal{O}, \hat{\mathcal{O}})$ between the surface of the watermarked and original Bunny objects when using ProMean and ProVar watermarking methods, respectively. In these figures we consider the same source location and number of watermark bits as in Figure 4.1. It can be observed that the watermarking errors produced by VPS are rather uniform and localized inside each iso-geodesic

strip as shown in the distortion study from Section 4.3.5. The propagation errors will only accumulate within each iso-geodesic strip but are not propagated to other strips.

4.4.3 Robustness evaluation

In the first experiment we analyze the embedding capacity when considering various ranges of bits, $M \in \{32, 64, 128, 256\}$ and $\alpha = 0.1$. The detection results for the proposed two methods are provided in Table 4.3. It can be observed that there are some problems with the ability to retrieve the watermark code when embedding $M = 256$ bits into a simple object such as the Fandisk. This is due to limitations for performing significant statistical changes in the geometry of small graphical objects. We have also tested the detection of false positives using the same keys in the same graphical objects. The tests gave zero detection rates in the case of both ProMean and ProVar when no watermark has been previously embedded.

In the following we consider embedding watermark codes of $M = 64$ bits with the strength $\alpha = 0.1$ for ProMean in (4.8) as well as for ProVar in (4.12) for the first four objects from Table 4.2 and we present the average detection results when embedding and detecting 100 different watermark codes by using 100 different random keys. The detection ratio is defined as the ratio between the number of correctly detected bits and the total number of embedded bits. The following results are obtained after embedding without attacks.

Object	Methods	32 bits	64 bits	128 bits	256 bits
Bunny	ProMean	1.00	1.00	1.00	0.99
	ProVar	1.00	1.00	1.00	0.99
Head	ProMean	1.00	1.00	1.00	1.00
	ProVar	1.00	1.00	1.00	1.00
Statue	ProMean	1.00	1.00	1.00	1.00
	ProVar	1.00	1.00	1.00	1.00
Dragon	ProMean	1.00	1.00	1.00	1.00
	ProVar	1.00	1.00	1.00	1.00
Fandisk	ProMean	1.00	0.99	0.99	0.93
	ProVar	1.00	0.99	0.97	0.88

Table 4.3: The bit detection ratio when varying the embedded information capacity.

In [31] it was stated that ChoMean and ChoVar methods are not suitable for wa-

termarking Computer Aided Design (CAD) graphical objects which contain flat regions. In the following we exclude the Fandisk object from the robustness tests which are only carried out on the other four objects. Four methods, ProMean, ProVar, ChoMean and ChoVar are compared when considering additive noise, smoothing, mesh simplification, quantization and uniform remeshing. Figure 4.9 shows the watermarked Bunny after various attacks. In the experiments we vary the intensity of the attack up to the level where the resulting object becomes seriously degraded.

Figure 4.9(a) shows watermarked Bunny corrupted with additive noise when $\epsilon = 0.005$. The additive noise attack is defined in equation (3.24) in section 3.6.4. The plots from Figure 4.10 show the robustness against noise when varying ϵ for the four methods, ProMean, ProVar, ChoMean and ChoVar for all four graphical objects. From these plots it can be observed that ProMean and ChoMean methods provide better results than ProVar and ChoVar.

We use the Laplacian Smoothing method proposed in [119] for the smoothing attack. A watermarked and smoothed Bunny, when considering a smoothing parameter $\lambda = 0.3$ and 10 iterations according to the method from [119], is shown in Figure 4.9(b). The plots displaying the robustness of the watermarking methods are provided in Figure 4.11 for all four graphical objects. As can be observed from the plots, ProMean provides slightly better results for Bunny, Head and Statue graphical objects while ProVar is better for the Dragon. This is due to the fact that the Dragon is a very dense object and contains many variance on its surface. The Laplacian smoothing requires more iterations to smooth the object in this case.

The quadratic metric simplification software described in [50] was used for testing the robustness at mesh simplification. This approach for mesh simplification was chosen because it represents a harder to resist attack for the watermark than other mesh simplification algorithms. Figure 4.9(c) shows watermarked Bunny object after being 90% simplified. Figure 4.12 contains the plots showing the resistance to mesh simplification attack for the objects when using all four methods. The proposed methods are slightly less robust to mesh simplification than Cho's methods because the latter methods introduce distortions in the object surface which resist such an attack.

Figure 4.9(d) shows the Bunny object represented using 7 bits quantization. In

this attack, the difference between the maximum and minimum value of vertices along each axis is quantized by a specific number of bits. As shown in Figure 4.13 all four algorithms are fairly robust up to 8 bits quantization. ProMean provides better results for the Head, Statue and Dragon graphical objects, while ChoMean and ProMean provide better results for the Bunny when compared to ChoVar and ProVar methods.

We compare the robustness of all four watermarking methods against the resampling attack by using the algorithm proposed in [13]. This attack consists of uniformly sampling random vertices from the graphical object surface and connecting them in a way that is not related to the original mesh. The vertices are moved on the local triangle mesh plane which does not guarantee that they would lie on the initial surface. The number of sampled vertices represent $\{100\%, 80\%, 60\%, 40\%, 20\%\}$ from the total number of vertices in the original object. The results are shown in Figure 4.14. As can be observed from these plots the best results are provided by ProVar and afterwards ProMean methods for all four objects.

4.4.4 Parameter influence

The proposed graphics watermarking methodology depends only on the size of the watermark code length M. In the following we study the effect of the embedding capacity size when watermarking the Bunny object. Figure 4.15(a) shows the distortion measured using Metro, *i.e.* equation (2.16), when embedding $M \in \{32, 64, 96, 128\}$ bits. When the message length is increased, the width of the histogram becomes narrower. The change of the corresponding vertex norm is smaller. Thus the distortion is less. Figure 4.15(b) provides the error $E(\mathcal{O}, \hat{\mathcal{O}})$ when the watermark strength is increased such that $\alpha \in [0.05, 0.30]$ for all four methods. It can be observed that the distortion, as measured using equation (2.16), increases linearly with α for all four methods from Figure 4.15(b). From the plots from Figure 4.15 it is evident that the proposed methods whose names start with Pro show much lower distortions when increasing the strength factor α for embedding the watermark information then the methods which start with Cho, which use the same histogram based procedure depending on the α parameter for watermark embedding. It can be

observed from these figures that the watermarking methods based on changing the variance by using equation (4.12) provide lower levels of distortion than those based on changing the mean of histograms by using equation (4.8). The plots from Figure 4.16 show the robustness to additive noise on the watermarked Bunny by using (3.24) when varying ϵ and when increasing the embedded bit capacity for ProMean and ProVar methods. Evidently, from these plots, the graphical objects embedding a larger amount of information are performing worse under the noise attack when compared with those carrying fewer bits. The plots from Figures 4.17(a) and 4.17(b) show the robustness at noise, when increasing the strength factor α, for ProMean and ProVar, respectively. It can be observed that when increasing α we improve the watermark robustness up to a certain level. However, when $\alpha > 0.15$, the errors on the surface of graphical object become significant.

4.4.5 Source point location effect on the robustness

According to our experiments the robustness of the proposed methods can be improved if stable source locations can be found for the FMM method. Small changes in the orientation of the principal axis for the watermarked object after attacks can cause significant errors in the detection stage. In Figure 4.18 we compare the detection errors for the Bunny object after the noise and simplification attacks when considering various ways for deciding on the starting point $\mathbf{s}$ when calculating the geodesic distances. We compare the volume alignment described in Section 4.3.1 with the PCA alignment and the robust feature point detection described in [3] for deciding $\mathbf{s}$. In the plot from Figure 4.18(a) we consider additive noise according to equation (3.24), while in Figure 4.18(b) we consider the mesh simplification method described in [50]. As can be observed from these plots, all three source point localization methods provide similar robustness against additive noise because such an attack does not significantly change the object center or its principal axis. However, the mesh simplification may affect differently various regions of the graphical object leading to changes in the object center as well as to the orientation of its main axes. It can be observed from Figure 4.18(b) that by using PCA for finding the source location $\mathbf{s}$ is particularly sensitive to mesh simplification leading to reduced bit de-

tection rates. Defining robust feature points gives the highest robustness overall. The source localization using the significant feature position is shown in Figure 4.19 for the original Bunny object, watermarked and when attacked by using additive noise and mesh simplification. The robustness of the significant feature point is very good as it can be observed from these plots. However, the computational complexity requirement for finding the robust feature point is of $O(N^2 \log N)$. Moreover, locating the source at a specific feature point produces a reduced security since this can be easily guessed by an attacker. After considering both the computational complexity and the security to attacks we decide to use the volume moment alignment, as described in Section 4.3.1, for defining the starting point $\mathbf{s}$.

4.4.6 Computational complexity

The computational complexity of the proposed watermark embedding methodology is of order $O(N \log N + N|\mathcal{N}_{\mathcal{D}}|)$, where N is the number of vertices and $|\mathcal{N}_{\mathcal{D}}|$ represents the average number of vertex neighbours from the set $\mathcal{D}$ for the updating vertex. The first component of the computational complexity corresponds to FMM while the watermark embedding using VPS consists of changing at most a vertex per triangle. However, the updating may have to be repeated a number of times equal to the number of adjacent vertices contained in the set $\mathcal{D}$ in order to ensure that the updated geodesic distance is truly consistent with the watermark bit distribution. All the experiments are carried out on a computer with the CPU as AMD Athlon(tm) 64 X2 Dual Core Processor 4200+, 2.20GHz, 8GB RAM under 64 bits Linux operating system. The processing times in seconds required by various stages of the proposed methods are provided in Table 4.4. In this table, PT represents the time for calculating the geodesic distance on the object mesh using FMM, VPS represents the average computation time of the vertex placement scheme. The columns VolMom alignment and the PCA alignment provide the mesh alignment timing using the procedure described in Section 4.3.1 and the volumetric PCA, respectively, while Embedding Time gives the total required time for watermark embedding. It takes approximately 3 hours to find the significant feature point when using the method from [3] on the Bunny object which is the smallest object used in our experiments.

Object	PT	VPS	VolMom Alignment	PCA Alignment	Embedding Time
Bunny	0.26	0.40	0.16	0.06	0.99
Head	1.21	2.01	0.68	0.21	4.39
Statue	1.82	2.90	1.22	0.34	6.26
Dragon	3.86	6.60	3.38	0.70	14.28

Table 4.4: Table showing the watermark embedding times in seconds.

Table 4.5 demonstrates the convergence rate of the VPS algorithm. When one vertex is moved to the new position, its geodesic distance is required to be updated. Then it is possible that its new geodesic distance is calculated from another triangle rather than the one which was used for updating. So the vertex may need to be moved again to ensure that its geodesic distance satisfies the watermarking condition. In this case, the vertex may end up going back and forth infinitely. We examine the convergence rate of the VPS algorithm in Table 4.5. As observed from the table, the VPS algorithm shows very good convergence rate. In our implementation, if the vertex is iterated more than 50 times, we treat it as nonconvergent vertex and jump out loop. About 80% of vertices need only to be replaced once, which means most of the vertices are moved in a distortion minimum manner. There are only very few percentage of vertices that goes into the infinite loop. In this case, we detect the failure to converge and stop the iteration. As our watermarking algorithm is a statistical method, such a small percentage of nonconvergent vertices will not affect the robustness very much.

Object	1 Move	2 Moves	3 Moves	[4,. . . , 50]	Nonconvergent
Bunny ProMean	80.69%	13.80%	2.86%	2.20%	0.45%
Bunny ProVar	82.01%	14.26%	2.42%	0.80%	0.51%
Head ProMean	81.37%	9.42%	3.13%	5.79%	0.29%
Head ProVar	82.34%	8.05%	4.13%	5.25%	0.23%
Statue ProMean	78.58%	11.00%	3.56%	6.52%	0.34%
Statue ProVar	77.75%	11.17%	3.74%	7.06%	0.28%
Dragon ProMean	78.52%	4.93%	2.95%	12.97%	0.63%
Dragon ProVar	78.87%	5.64%	3.87%	11.02%	0.60%

Table 4.5: Table showing the convergence rate of the VPS.

4.4.7 Discussion

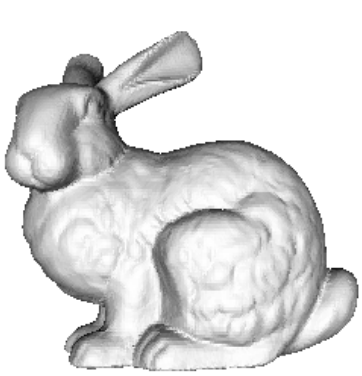
The proposed watermarking methodology depends on defining a source location $\mathbf{s}$ as a reference for calculating the geodesic distances as required by FMM. If the source point is selected as described in [3], the algorithm will obtain the highest robustness as shown in Figure 4.18. In this case the algorithm is partially robust against cropping as shown in [3]. However, the disadvantage is that the feature point finding algorithm requires a computational complexity which is a lot larger than that required by the other stages of the watermarking method. Moreover, significant graphical object features are very easy to guess providing an easier task to an attacker aiming for destroying or reading the watermark message and significantly reducing the watermark security. Volume moment alignment is the second robust scheme and has a much lower computational complexity requirement. As the starting point in this case is defined according to an intersection between a key generated vector direction and the object surface, malicious attackers will not be able to find the embedded message without knowing the secret key. The proposed scheme requires the knowledge of the watermark code length in the detection stage. Moreover, the cropping attack may significantly change the object center and result in the watermark detection failure. Another constraint is that the volume alignment will be undefined if the object is not closed or contains holes. PCA applied directly to the vertex coordinates can be used as an alignment scheme in order to define the source point. The PCA based watermarking scheme can be used on non-closed objects but suffers of other drawbacks such as an undefined positive principal axis [49], low robustness against mesh simplification, etc.

4.5 Conclusion

This chapter proposes a new 3-D watermarking methodology based on statistics of geodesic distances defined using the Fast Marching Method. The object is aligned using the volume moments and the starting point for FMM is defined as the intersection between a random direction generated according to a key and the object mesh. The surface of the graphical objects is segmented into strips, each containing

vertices located in a well defined band of geodesic distances. All strips have identical geodesic distance width and each of them is used for embedding a bit. Distributions of geodesic distances are formed for each strip. Two different statistical methods are proposed for watermark embedding, one by changing the mean of the distribution and the other by changing its variance. The vertices are changed along the graphical object surface, using the Vertex Placement Scheme such that the resulting object distortion is minimal. The study from this chapter shows theoretically as well as by means of numerical simulations that the proposed methodology ensures a minimal distortion in graphical objects following watermarking. The proposed methodology has low computational demands and results in watermarks which are robust to various mesh attacks excepting object cropping. The security of the proposed method is enforced in various stages including the source point location, segmentation and the construction of the statistical variable.

The method strongly relies on the robustness of the object centre and principal axis. Thus, the method is not robust against any attack that destroy the object principal axis such as cropping. The method is also not applicable on small mesh objects as there are not sufficient samples for constructing the statistical variable for watermarking. Furthermore, the VPS does not make sure that all the vertex placements converge although the nonconvergent rate is very small. In the future, it is possible to improve the VPS method so that all the vertices can converge within a small number of iterations.



(a) Bunny

(b) Head

(c) Statue

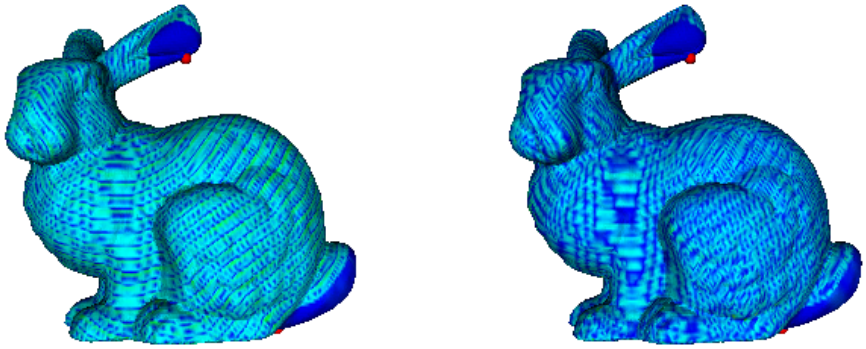
(d) Dragon

(e) Fandisk

Figure 4.6: Graphical objects used in the experiments.

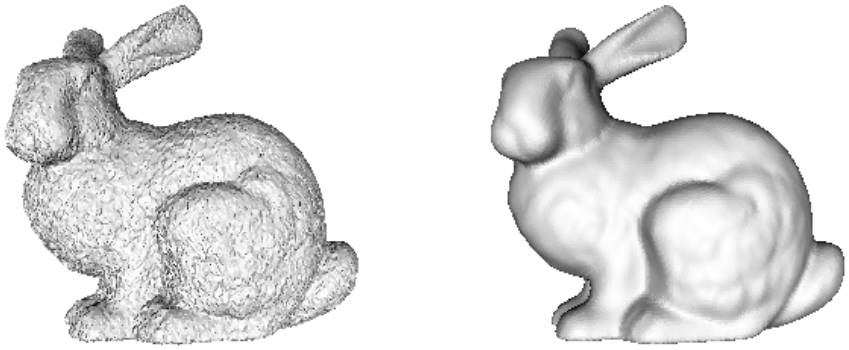
(a) (b) (c) (d) (e)

Figure 4.7: Comparison of visual distortion by displaying zoomed detail of original objects and the watermarked objects. (a) Original. (b) Watermarked using ProMean. (c) Watermarked using ProVar (d) Watermarked using ChoMean (e) Watermarked using ChoVar



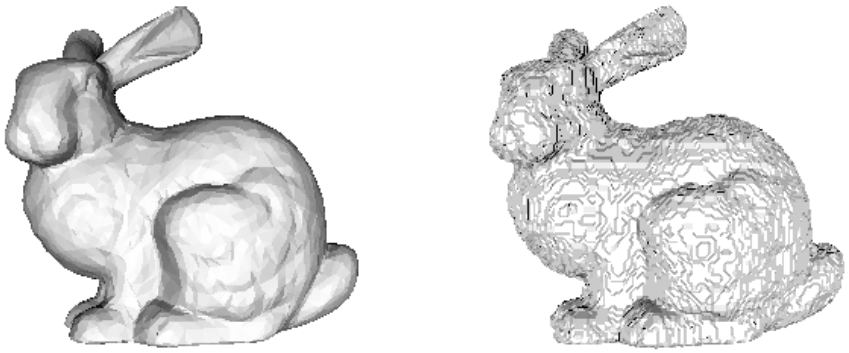
(a) Distortion by ProMean method. (b) Distortion by ProVar method

Figure 4.8: Distortion produced by VPS. Red point is the source. The lighter the color, the more the distortion.



(a) Additive random noise with $\epsilon = 0.005$.

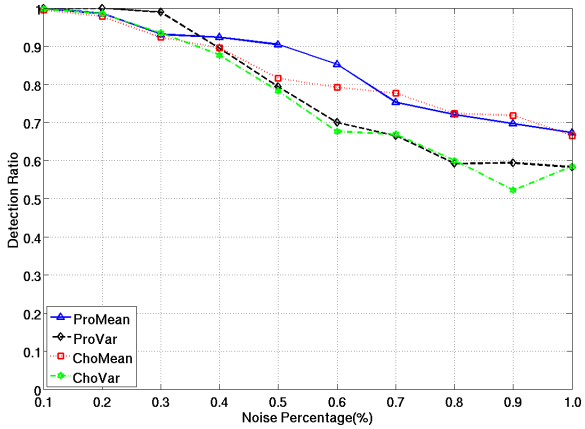
(b) Smoothing when considering $\lambda = 0.3$, 10 iterations



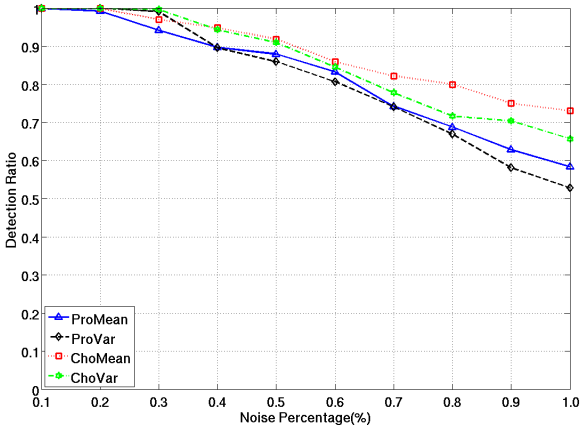
(c) 90% Mesh simplification.

(d) 7 bits quantization

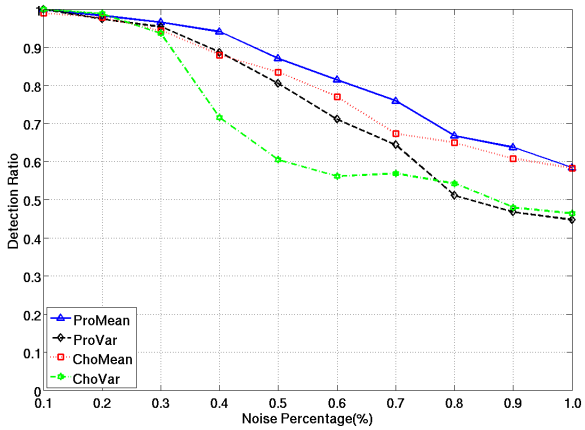
Figure 4.9: Watermarked Bunny object after various attacks.



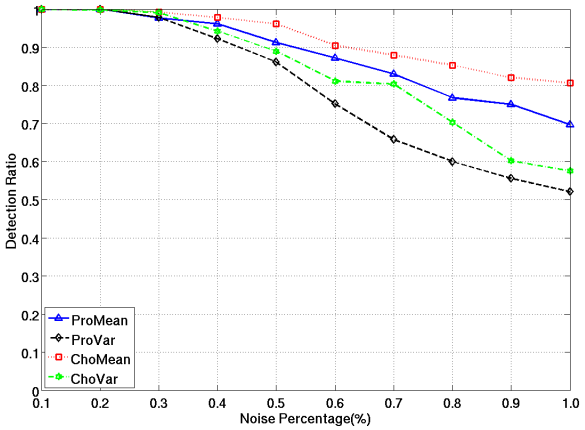
(a) Bunny



(b) Head



(c) Statue



(d) Dragon

Figure 4.10: Plots showing the robustness against additive noise.

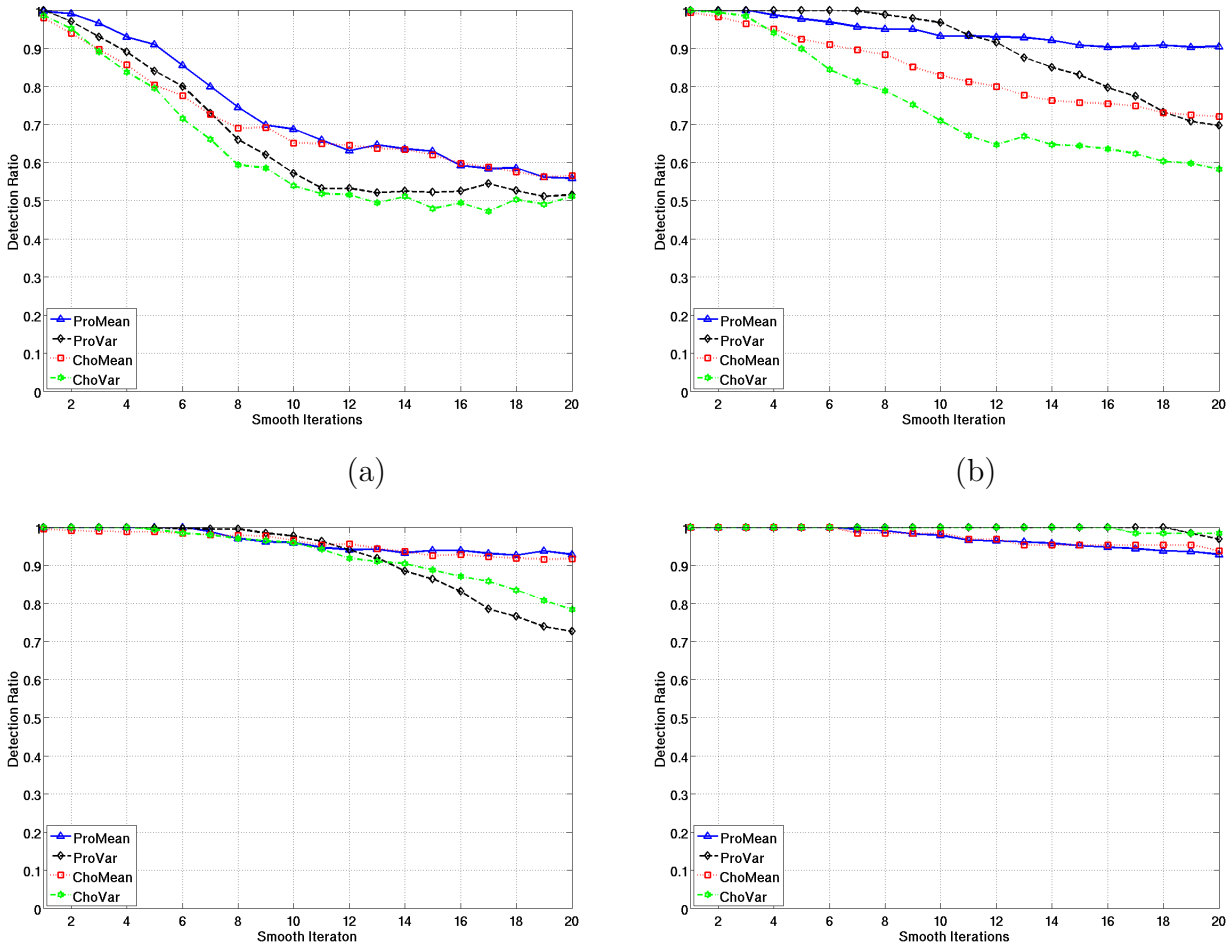


Figure 4.11: Plots showing the robustness at smoothing: (a) Bunny, (b) Head, (c) Statue, (d) Dragon

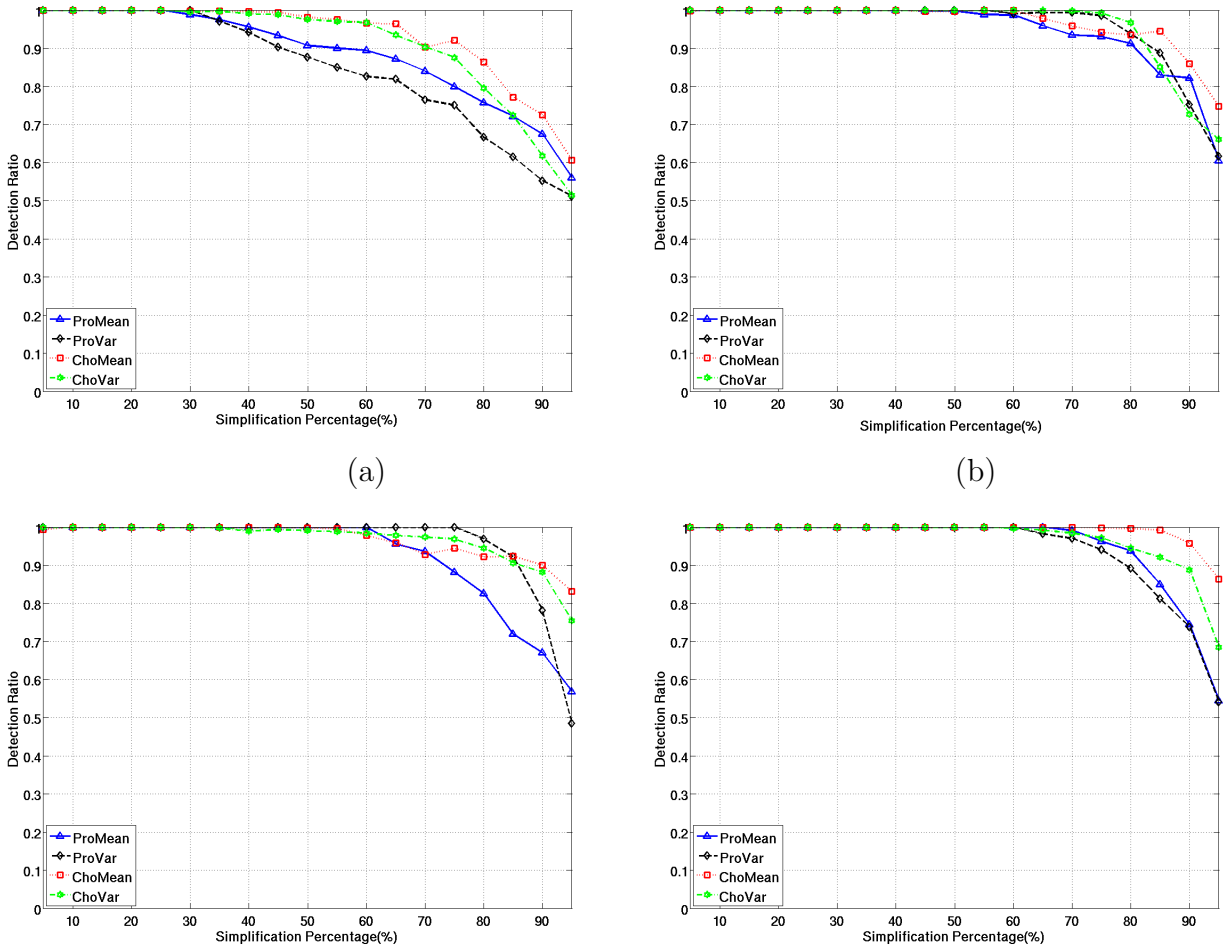


Figure 4.12: Plots showing robustness at mesh simplification: (a) Elephant, (b) Bunny, (c) Statue, (d) Dragon

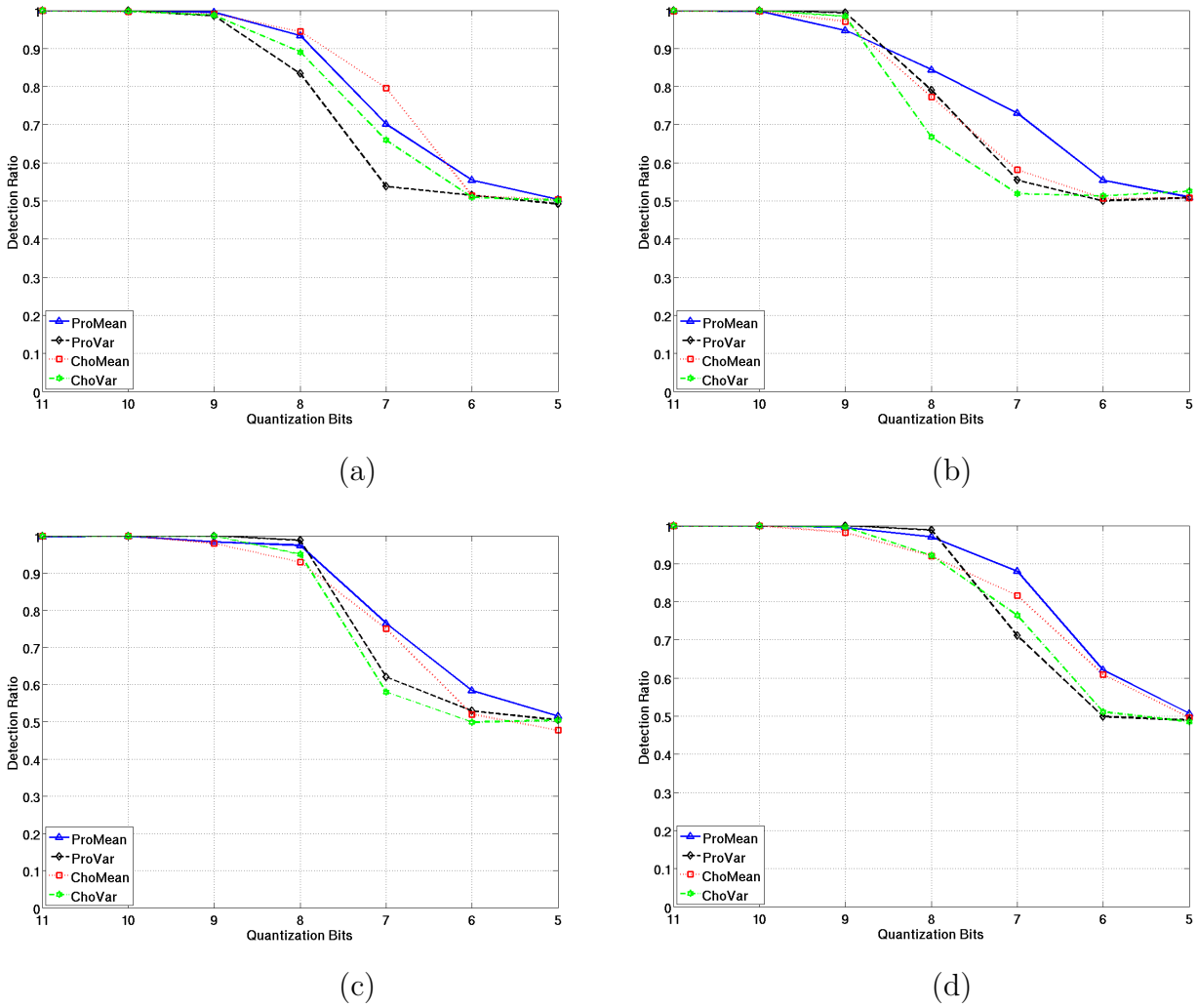


Figure 4.13: Plots showing robustness at bit quantization: (a) Bunny, (b) Head, (c) Statue, (d) Dragon

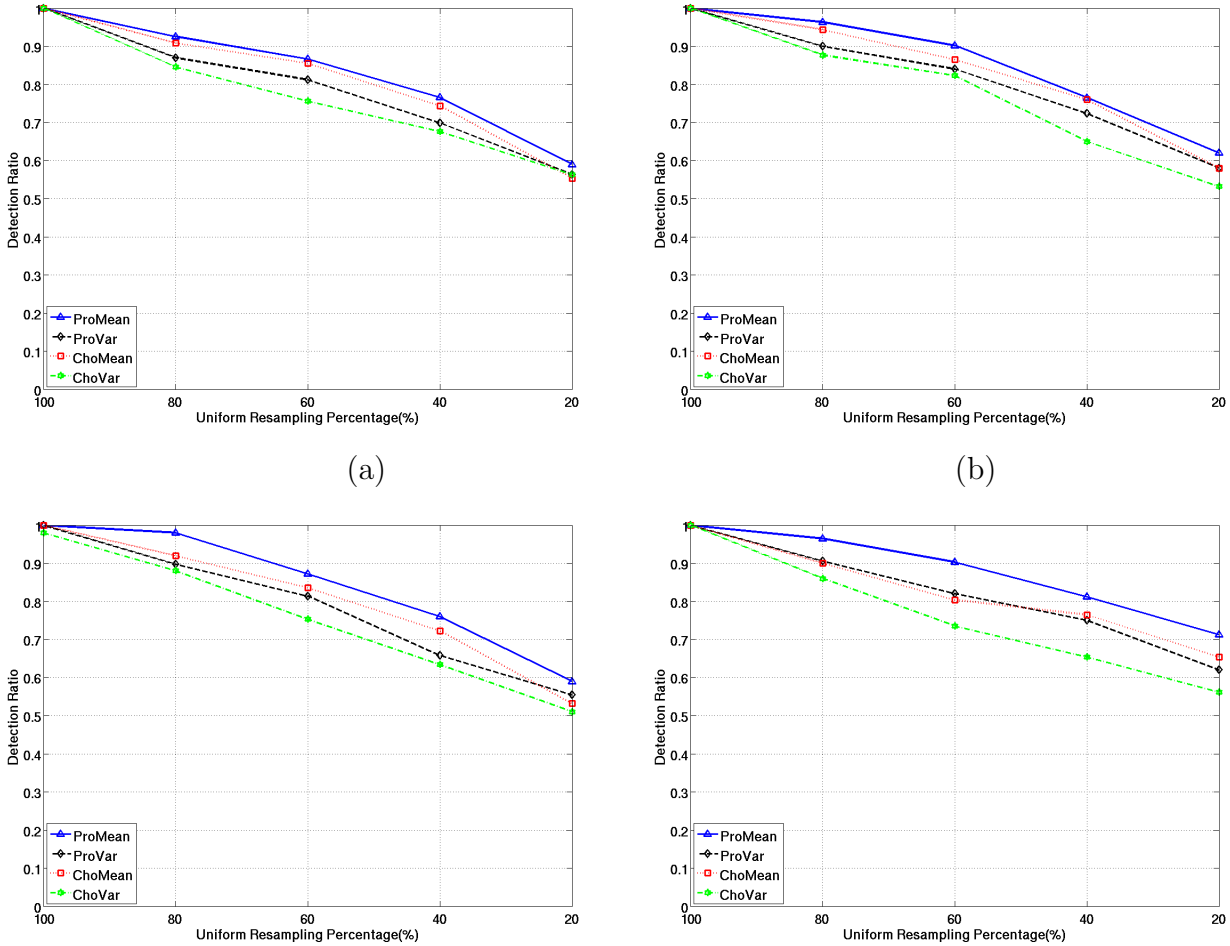
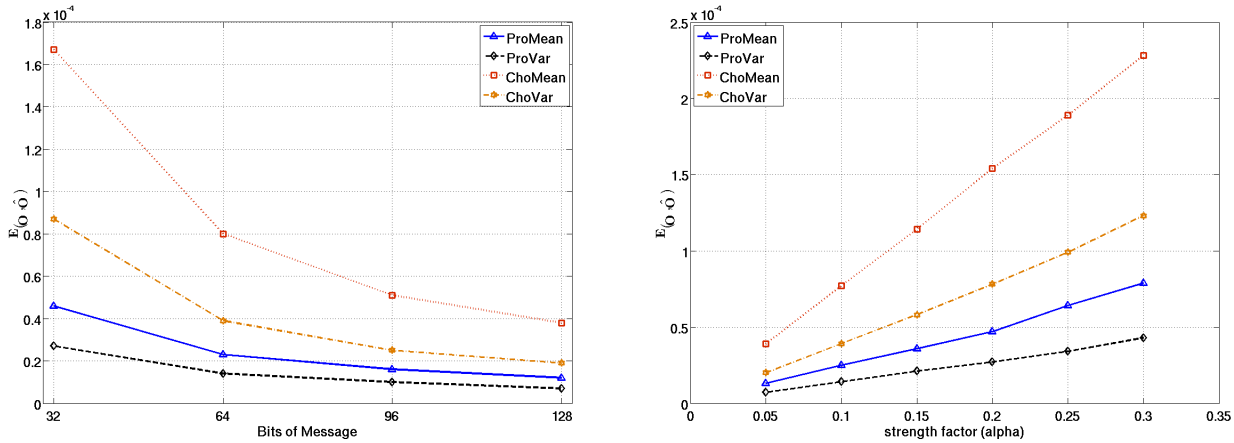
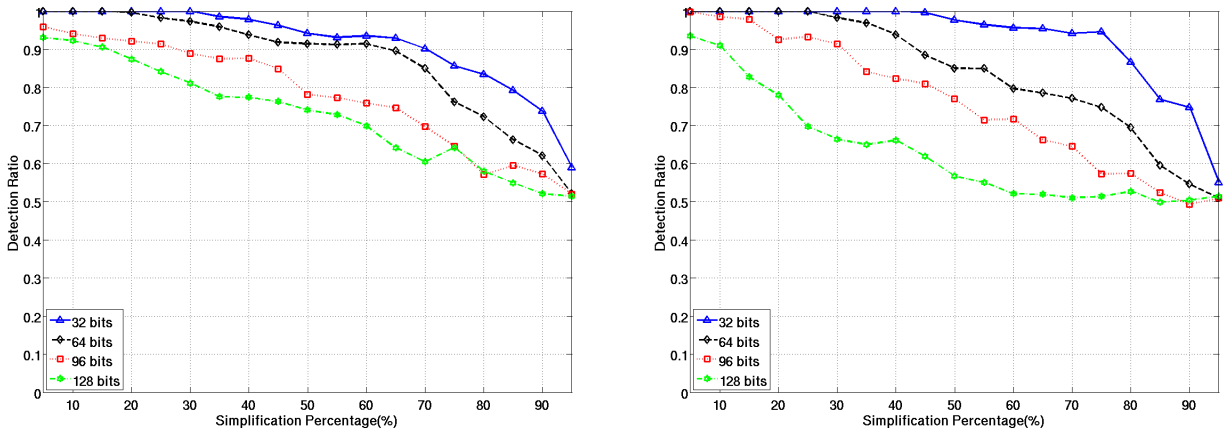


Figure 4.14: Plots showing robustness at resampling: (a) Bunny, (b) Head, (c) Statue, (d) Dragon



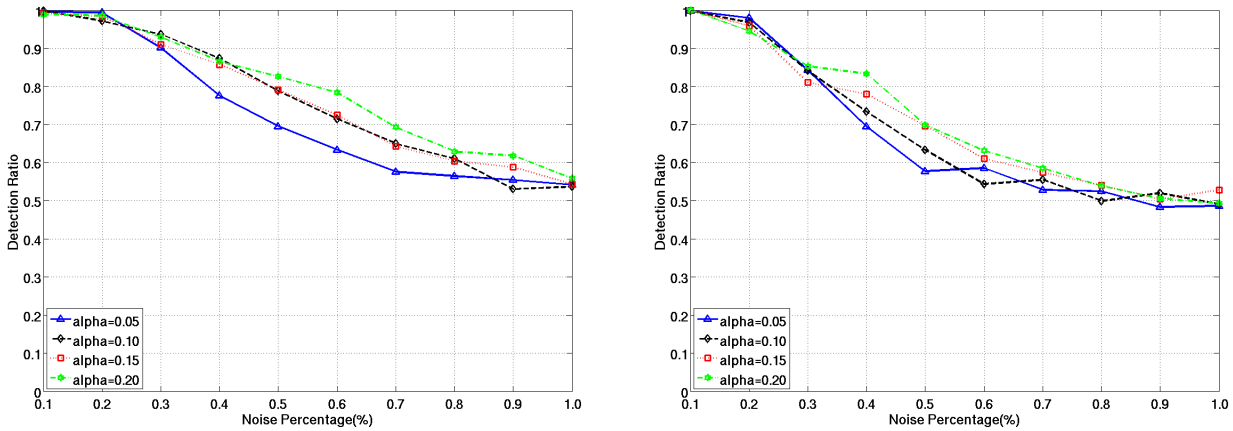
(a) (b)

Figure 4.15: Visibility distortion when varying various watermark parameters. (a) Relation between visual distortion and bit capacity. (b) Relation between visual distortion and strength factor α.



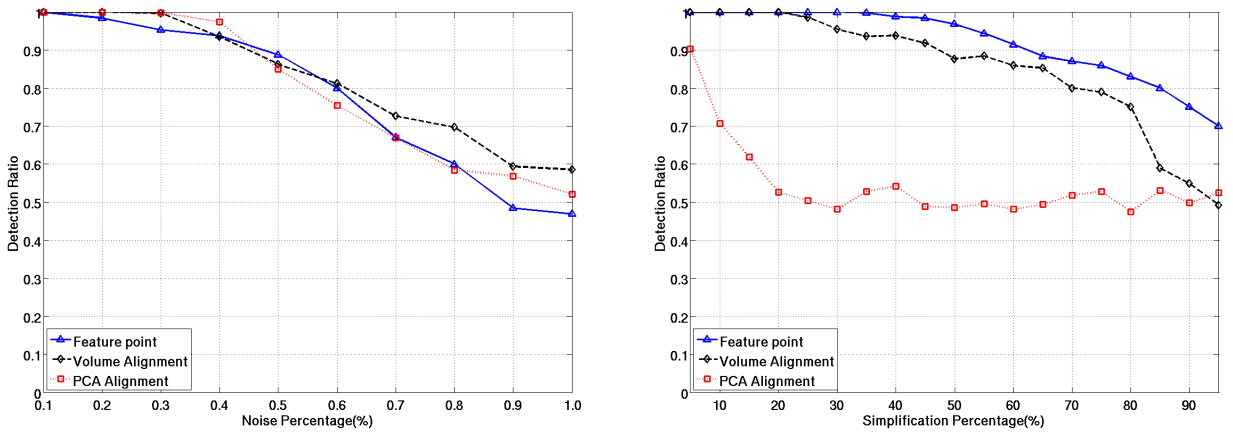
(a) (b)

Figure 4.16: Simplification robustness results when increasing the embedded bit capacity. (a) ProMean, (b) ProVar



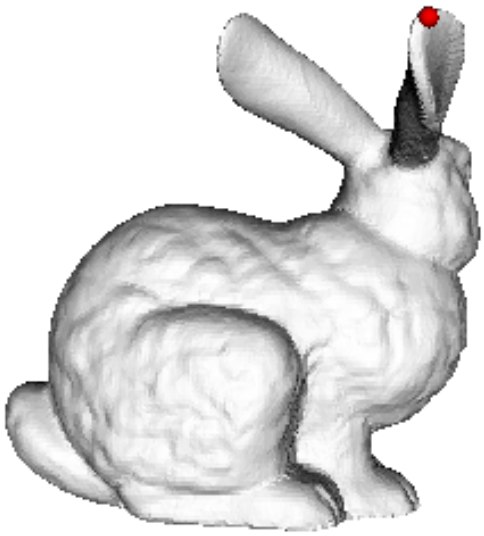
(a) (b)

Figure 4.17: Additive noise robustness when increasing the watermark strength factor α. (a) ProMean (b) ProVar.

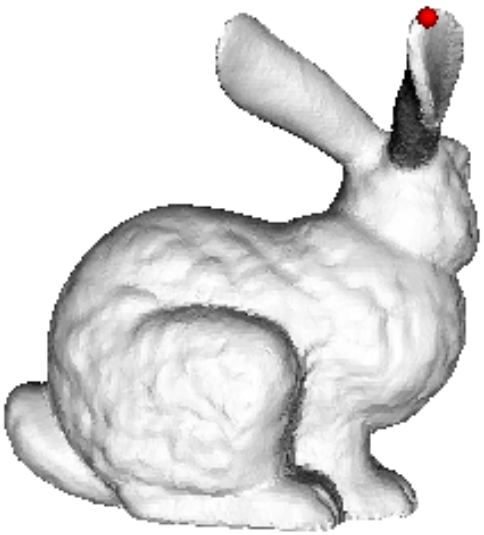


(a) (b)

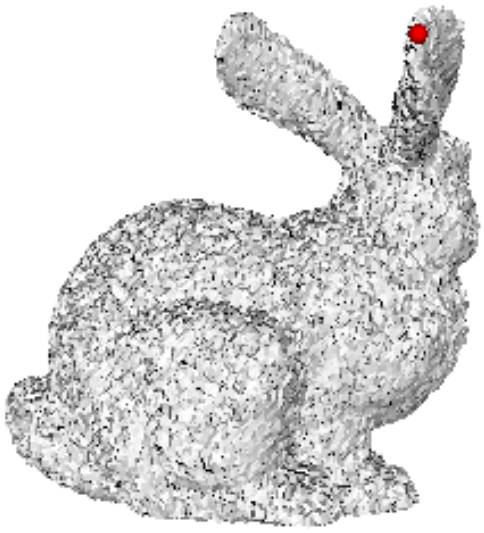
Figure 4.18: Comparison of different starting point schemes.



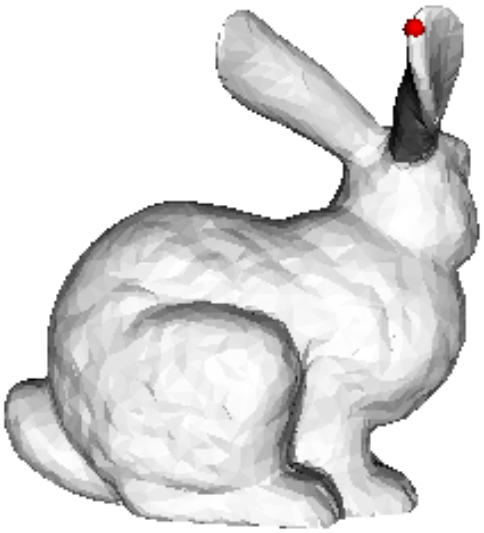
(a) Original



(b) Watermarked



(c) Additive noise with $\epsilon = 0.01$



(d) Mesh simplification of 90%

Figure 4.19: Starting point detected using the significant feature point described in on the Bunny graphical objects before and after attacks. The red circle shows the starting point.

Chapter 5

Optimization Watermarking

5.1 Introduction

In the previous chapter, we have examined the property of the distribution of the geodesic distance of the mesh. Watermarking the geodesic distance has shown a good balance between robustness and the surface distortion. In this chapter, we propose two novel spatial domain watermarking methods by changing the distance from the vertex to the object centre.

Watermarking methods can be categorized as deterministic method [19, 23] or statistical methods [31, 142] and all the methods in this thesis are statistical based methods. The methods from the first category employ a set of constraints for embedding messages while the second category extract the message by using a statistical test. Usually, deterministic methods allow a higher capacity of information embedding, making them suitable for steganography, but achieve lower robustness to attacks. On the other hand, the statistical methods are more robust but they achieve lower embedding capacity rates. Zafeiriou *et al* [142] proposed a robust watermarking method using distributions of distances from surface vertices to the local symmetry axis. Cho *et al* [31] proposed using vector norms representing distances from object surface to its centre as a statistical variable. In this approach sets of vertices are grouped into bins according to their distance to the object centre. The embeddings are performed by changing the vertex locations according to two histogram mapping functions. Two statistical algorithms are proposed in [31] by

changing the mean or the variance of the statistical variable. The statistical algorithms proposed by Cho *et al* show excellent robustness against most common mesh attacks. Moreover, these algorithms are blind and do not require the knowledge of additional information or specific object representation properties such as a regular connectivity. However, these algorithms produce visible artifacts such as ripples on the 3D object surface, which limits their usage in many application.

In this chapter, we propose two new methods employing the minimizing a surface distortion function. In the first approach, we employ the Quadric Error Metric (QEM). QEM was used as a measure of surface similarity for mesh simplification by Garland *et al* in [50]. The aim in mesh simplification is to produce similar graphical objects with fewer vertices. This measure was used for controlling the surface error for digital object watermarking [91] and was shown to have a similar robustness with the Cho *et al* methods [31]. In the second approach, we propose a new method for 3D watermarking using a novel error function. The surface error has been studied extensively in the area of mesh approximation [50,61]. The watermark distortion can be minimized by using a proper defined surface error function. This error function considers the following distances from the watermarked vertex location: the Hausdorff distance [7, 39] to the original object surface, to the watermarked surface as well as the Euclidean distance to the original vertex location. The vertex is placed, such that the error function is minimized, by using the Levenberg-Marquardt optimization method in spherical coordinates. Only a few iterations are necessary for the convergence of the proposed method. Levenberg-Marquardt is well-known optimization method which was used for surface fitting of graphical objects [29], shape processing [47], as well as for image watermarking [9]. The proposed methodology ensures a minimal distortion in the resulting watermarked 3-D mesh and constitutes a generalization of the methods from [31, 91]. The rest of the Chapter is organized as follows. Section 5.2 gives the statistical background of the watermarking methods. Section 5.3 describes the general framework of the vertex norm watermarking methods. Section 5.4 proposes the discretization watermarking method. Section 5.5 introduces the L-M method which minimizes the distortion with respect to a novel surface error function. Section 5.6 shows the experimental results and followed by conclusion in section 5.7.

5.2 Statistical watermarking of mesh-based representations of 3-D objects

In this section we describe the steps of the proposed 3D watermarking methodology which are adapted from the approach proposed by Cho *et al.* [31]. While the statistical approach proposed by Zafeiriou *et al.* [142] employs distributions of distances between vertices and the local symmetry axis, Cho *et al.* considers vertex norms, representing the distance from vertices located on the object surface to its center. The watermarking steps described in the following are: statistical variable representation, histogram mapping function and the watermark detection algorithm.

5.2.1 Statistical variable representation

Assume that we want to embed a code of M bits into a 3D object $\mathcal{O}$. Let us denote the object center by O and by V_j a vertex on its surface, while their coordinates are $\mathbf{o}$ and $\mathbf{v}_j$, respectively. The vertices of the mesh object $\mathcal{O}$ are clustered into M bins such that each bin is used to hold one bit of message $B_i,\ i = 1, \ldots, M$. In this chapter, we cluster the vertices according to their distance from a vertex $\mathbf{v}_j$ to the centre of the object $\mathbf{o}$, calculated as:

$$\mathbf{o} = \frac{1}{A(\mathcal{O})} \sum_{\mathbf{v}_j \in \mathcal{O}} A(\mathbf{v}_j)\mathbf{v}_j \tag{5.1}$$

where $A(\mathbf{v}_j)$ is the sum of the areas of the triangular faces incident to the vertex $\mathbf{v}_j$, and $A(\mathcal{O})$ represents the area of the entire surface of the graphical object $\mathcal{O}$. The object center defined in this way is more robust than by simply taking the average of all its vertices, as it was used in [31], particularly when considering the robustness of the watermarked object to various attacks such as remeshing, simplification, etc. Other ways of evaluating the object center could have been by using the object moments [123].

For a given vertex $\mathbf{v}_j \in \mathcal{O}$, let us denote its distance to the centre of the object $\mathbf{o}$, by $\rho_j = \|\mathbf{v}_j - \mathbf{o}\|$ and consider this as a statistical variable. After ranking these distances we find $\rho_{min} = \min_j(\rho_j)$ and $\rho_{max} = \max_j(\rho_j)$, where $j = 1, \ldots, |\mathcal{O}|$ and

$|\mathcal{O}|$ represents all vertices from $\mathcal{O}$. Then all the vertices are grouped into M sets according to their distances to the object centre $\mathbf{o}$ as:

$$\mathcal{B}_i = \{\mathbf{v}_j \in \mathcal{O} \mid \rho_{min} + |\mathcal{O}| * \varepsilon + (i-1)\rho_b \leq \rho_j \ , \ \rho_j < \rho_{min} + |\mathcal{O}| * \varepsilon + i\rho_b\} \quad (5.2)$$

where each of these sets contains a number of vertices equal to:

$$\rho_b = \frac{(1-2\varepsilon)(\rho_{max} - \rho_{min})}{M} \quad (5.3)$$

where $\varepsilon \in [0, 0.15]$ represents a small percentage accounting for outliers which are likely to correspond to specific object features and are eliminated from the further consideration for watermarking in order to avoid visible distortions. The value of ε is generated as a random value within the range $[0, 0.15]$ seeded by the secret key. This adds to the watermark security because without the knowledge of the secret key it would be impossible to retrieve the embedded watermark.

5.2.2 The histogram mapping function

The histogram mapping function presented in this section is theoretically the same as the one which is presented in Section 4.3.3. However, the statistical variable is different from Chapter 4 where it was the distribution of geodesic distance. In this chapter, we consider the distribution of vertex norms *i.e.* the distance from the object centre to the vertex as the statistical variable. In order to avoid any confusion, here we detail all the necessary equations related to this chapter although they are theoretically equivalent to the previous one introduced in Chapter 4.3.3. Thus, the methods in this chapter share the same initialization as Cho's method [31].

In the following we consider two methods for statistically embedding a message bit into the i'th bin. The first bit embedding method changes the mean value of its corresponding histogram while the second method embeds the bit by changing the histogram variance, [31]. The minimum and maximum statistical variable values are calculated for each bin i as:

$$\rho_{i,min} = \rho_{min} + |\mathcal{O}| * \varepsilon + (i-1)\rho_b \quad (5.4)$$

$$\rho_{i,max} = \rho_{min} + |\mathcal{O}| * \varepsilon + i\rho_b \tag{5.5}$$

In the first embedding method, the statistical variables are firstly normalized to the range $[0, 1]$ by using:

$$\tilde{\rho}_{ij} = \frac{\rho_{ij} - \rho_{i,min}}{\rho_{i,max} - \rho_{i,min}} \tag{5.6}$$

As shown in [31], the distribution of the statistical variable $\tilde{\rho}_{ij}$ is close to a uniform distribution. Thus, the expected mean value of the statistical variable is $1/2$. In order to embed one bit we introduce a bias in the corresponding histogram by changing its mean value as:

$$\hat{\mu}_i = \begin{cases} \frac{1}{2} + \alpha & \text{if } B_i = 1 \\ \frac{1}{2} - \alpha & \text{if } B_i = 0 \end{cases} \tag{5.7}$$

where α is the watermark strength factor influencing the visual distortion as well as the robustness, while B_i, for $i = 1, \ldots, M$, is the bit to be embedded in the i'th bin. In order to change the normalized distances for fulfilling (5.7), the first histogram mapping function is defined as:

$$\tilde{\rho}'_{ij} = \tilde{\rho}^{\beta}_{ij} \quad \begin{cases} \beta \in (0, 1) & \text{if } B_i = 1 \\ \beta \in (1, \infty) & \text{if } B_i = 0 \end{cases} \tag{5.8}$$

where $\tilde{\rho}'_{ij}$ is the resulting watermarked normalized vertex norm. Finally, the watermarked vertex norms are obtained by mapping $\tilde{\rho}'_{ij}$ back to the original interval as:

$$\hat{\rho}_{ij} = \tilde{\rho}'_{ij}(\rho_{i,max} - \rho_{i,min}) + \rho_{i,min} \tag{5.9}$$

The second method embeds the message by changing the variance of the norms. In this case, ρ_{ij} is normalized to the range $[-1, 1]$ as:

$$\tilde{\rho}_{ij} = 2\frac{\rho_{ij} - \rho_{i,min}}{\rho_{i,max} - \rho_{i,min}} - 1 \tag{5.10}$$

The expected variance in this case is $1/3$ for a uniform distribution. We embed one

bit of message by modifying the variance of $\tilde{\rho}_{ij}$ according to :

$$\hat{\sigma}_i^2 = \begin{cases} \frac{1}{3} + \alpha & \text{if } B_i = 1 \\ \frac{1}{3} - \alpha & \text{if } B_i = 0 \end{cases} \tag{5.11}$$

The histogram mapping function for modifying each element from the set $\mathcal{B}_i$ is defined as:

$$\tilde{\rho}''_{ij} = sign(\tilde{\rho}_{ij})|\tilde{\rho}_{ij}|^{\beta} \quad \begin{cases} \beta \in (0,1) & \text{if } B_i = 1 \\ \beta \in (1,\infty) & \text{if } B_i = 0 \end{cases} \tag{5.12}$$

The watermarked vertex norms are obtained by the inverse normalization function:

$$\hat{\rho}_{ij} = \frac{1}{2}(\tilde{\rho}''_{ij} + 1)(\rho_{i,max} - \rho_{i,min}) + \rho_{i,min} \tag{5.13}$$

In the approach of Cho *et al.* [31], the vertex is moved along the direction from the object centre to the current vertex $\overrightarrow{OV}_i$, to a new location corresponding to the watermark in order to satisfy the watermarked vertex norm distributions. This means that the Euclidean distance from the vertex to the object center becomes $\hat{\rho}_i$, according to the bit B_i , and its corresponding distribution mean fulfills the condition from equation (5.7) for the mean-based watermarking, or its variance fulfills equation (5.11) for the variance-based watermarking. The change of vertex location may affect the object centre in principal. However, as shown in Figure 3.12, the volume moment alignment scheme is very robust. And as long as the watermarking process preserve the surface well enough, it will not affect the watermark detection. In Section 5.3 we provide a new mesh surface error criterion in order to achieve a lower 3D object modification visibility following watermarking.

5.2.3 Watermark Detection

The watermark extraction algorithm is blind, *i.e.* it does not need the original object in the detection stage. The detector needs to know the secret key, the watermarking algorithm and the number of bits that was embedded in order to recover the watermark message. As a statistical method, it is impossible to generate the bins without knowing the length of the message. In the detection stage we follow the same steps as presented in the previous section. After evaluating the object center

o, we segment the object vertices into bins assuming that the number of embedded bits M is known. Then, the statistical variable of distances from the vertices to the object center are normalized according to either equation (5.6) or (5.10), for the mean or variance methods, respectively. For the first watermarking method, the message is extracted by applying a test on the mean value of the bin histogram:

$$\begin{cases} \text{if } \hat{\mu}_i > \frac{1}{2} & \text{then } B_i{=}1 \\ \text{if } \hat{\mu}_i < \frac{1}{2} & \text{then } B_i{=}0 \end{cases} \tag{5.14}$$

For the second watermarking method, the variance is calculated, and the bit is extracted following the test :

$$\begin{cases} \text{if } \hat{\sigma}_i^2 > \frac{1}{3} & \text{then } B_i{=}1 \\ \text{if } \hat{\sigma}_i^2 < \frac{1}{3} & \text{then } B_i{=}0 \end{cases} \tag{5.15}$$

5.3 Watermarking while minimizing the surface distortion

In the following we use the same statistical embedding and detection approaches as those described in the above section. However, unlike in [31] where the vertex is moved along the direction of the line joining the vertex and the object center O, we propose to change the location of the vertex such that the surface of the object is not modified. Invisibility of the changes caused by watermarking is one of the major requirements for the practical application of watermarking. In this section we outline a surface error minimization approach for 3-D object watermarking.

We can observe that $\|O\hat{V}_i\| = \hat{\rho}_i$ corresponds to a mapping on a 3D sphere centered at O and of radius $\hat{\rho}$, where O represents the centre of the object and $\hat{V}_i$ is the new vertex location following watermarking. Any point on the sphere is eligible to satisfy the watermark condition, *i.e.* it can potentially be the location of the vertex following watermark embedding. In the approach from [31], the vertex is moved along the direction of $\overrightarrow{OV}_i$ in order to fulfill the watermark condition. Such a movement guarantees that the resulting Euclidean distortion with respect to the previous vertex location is minimal. However, this change does not guarantee that

the object surface distortion is minimal. This is because the similarity between two surfaces is better measured by the Hausdorff distance and not by the Euclidean distance [6, 40]. Thus, a Hausdorff-related metric is a more appropriate metric to evaluate the distortion introduced by the vertex displacement by watermarking.

We propose solving the following system when watermarking 3D graphical objects under the constraint of minimizing the shape distortion:

$$\begin{cases} \|O\hat{V}_i\| = \hat{\rho}_i \\ \hat{\mathbf{v}}_i = \operatorname{argmin} E(\hat{\mathbf{v}}_i) \end{cases} \tag{5.16}$$

where E is a function measuring the distortion produced by watermark embedding when changing the vertex V_i to $\hat{V}_i$. The above system means that we are aiming to find a location on the sphere centered in O, fulfilling the watermark condition $\|O\hat{V}_i\| = \hat{\rho}_i$, such that the displaced vertex introduces a minimal distortion to a given error function E. Thus, the watermarking is now modeled as a non-linear optimization method subject to constraints. We propose two solutions for solving the system from (5.16).

The first method consists of the discretization of the 3-D object surface and fitting the surface of the sphere centered in O, of radius $\hat{\rho}_i$, by using a number of candidate points among the surface vertices. We use the Quadric Error Metric (QEM) which is used in mesh simplification [50] as the surface error metric E. The location which introduces the minimum distortion according to the QEM error function will be selected as the watermarked vertex position. The advantages of this method are that a reasonable high quality watermarked mesh can be produced with a low computational complexity requirement [91]. However, limited by the approximation candidates, the distortion is not guaranteed to be the true minimum over all possibilities.

In the second method proposed in this chapter, we introduce a novel surface error function which generalizes over both the Euclidean and the Hausdorff metrics [6,40]. The error function in this case contains three parts measuring: the Hausdorff distance to the original surface, the Hausdorff distance to the updated surface as well as the Euclidean distance between the watermarked vertex and the original vertex,

respectively. The weights of each component of the error metric can be adjusted by changing user-defined parameters resulting into a method which represents the generalization of the approaches from [31] and [91]. The watermarked vertex is firstly placed on the sphere, centered in O and then the best position minimizing the error function is found iteratively by the Levenberg-Marquardt method.

The schemes that implement the optimization of (5.16) are described in detail in the following two sections.

5.4 Discretization method

The first proposed vertex placement method using the quadric error metric is described in this section.

5.4.1 Quadric Error Metric

Quadric Error Metric (QEM), proposed by Garland and Heckbert [50], is used in surface quality assessment for mesh simplification methods. QEM assesses the similarity between the simplified and the original local surface of 3D graphical objects by using an edge collapsing procedure. It evaluates the distances from vertices from one mesh surface to corresponding planes in the other object mesh.

The standard representation of a plane consists of the set of all points for which $\mathbf{n}^T\mathbf{x} + d = 0$ where $\mathbf{n} = [n_x \ n_y \ n_z]^T$ is a plane normal vector which is of unit magnitude. (*i.e.*, $n_x^2 + n_y^2 + n_z^2 = 1$), $\mathbf{x} = [x \ y \ z]^T$ is a point in the 3D space and d is a constant. The quadric $Q_{\mathbf{F}}(\mathbf{x})$ is defined as the squared distance from a point X, of location $\mathbf{x}$, to a plane $\mathbf{F}$ in the 3D space, as:

$$Q_{\mathbf{F}}(\mathbf{x}) = \mathbf{x}^T\mathbf{A}\mathbf{x} + 2\mathbf{b}^T\mathbf{x} + c \tag{5.17}$$

where the quadric is described by the triplet $Q_{\mathbf{F}} = (\mathbf{A}, \mathbf{b}, c)$, where $\mathbf{A} = \mathbf{n}\mathbf{n}^T$, $\mathbf{b} = d\mathbf{n}^T$ and $c = d^2$.

An important property of the quadric is that it can be summed up easily for entire 3D surface regions and the result will be a triplet whose components are the sums of each individual quadric triplet components. Let us assume N_{F_i} as the

number of neighbouring faces adjacent to the vertex V_i of a mesh $\mathcal{O}$ and that each V_j face, $j = 1, \ldots, N_{F_i}$ lies on a plane $\mathbf{F}_j$. Then, the quadric error with respect to the vertex $\mathbf{v}_i$ given the location of a point $\mathbf{x}$ and N_{F_i} can be defined as :

$$Q_{\mathbf{v}_i} = \sum_{j}^{N_{\mathbf{v}_i}} Q_{\mathbf{F}_j} = \left(\sum_{j}^{N_{\mathbf{v}_i}} \mathbf{A}_j, \sum_{j}^{N_{\mathbf{v}_i}} \mathbf{b}_j, \sum_{j}^{N_{\mathbf{v}_i}} c_j \right) \tag{5.18}$$

where $Q_{\mathbf{F}_j} = (\mathbf{A}_j, \mathbf{b}_j, c_j)$. This formula means that each quadric component is made up of a separate sum of individual quadric components, each corresponding to a single plane $\mathbf{F}_j$. The quadric error of X with respect to V_i is defined as the sum of the squared distance from a point X to all the planes adjacent to V_i.

The quadric error metric is proved to have close relation to the local shape characteristics including its fundamental form, curvature and local moment. It has also been shown that its metric represents a lower complexity implementation of the Hausdorff distance [59]. Moreover, QEM has low requirements for the mesh surface properties, *i.e.* it can be even applied on a non-manifold mesh. In this study, we employ QEM as a measure in order to evaluate the local distortion error when selecting the location of the watermarked vertex.

5.4.2 Quadric Selective Placement Scheme

We propose to employ the QEM as the distortion measure, according to (5.16), as follows:

$$\begin{cases} \|O\hat{V}_i\| = \hat{\rho}_i \\ \hat{\mathbf{v}}_i = \operatorname{argmin} Q_{\mathbf{v}_i}(\hat{\mathbf{v}}_i) \end{cases} \tag{5.19}$$

The proposed scheme consists in finding a candidate point on each neighbouring face $\mathbf{F}_j$ of the vertex V_i, $j \in 1, \ldots, N_{F_i}$. Let us consider a triangle $\triangle MNK$ on the surface of the object $\mathcal{O}$ as shown in Figure 5.1. We locate the closest and furthest away points from O, from all the points X inside $\triangle MNK$, denoted as P and Q, respectively, such that:

$$\begin{cases} ||OP|| = \min(||OX||) \\ ||OQ|| = \max(||OX||) \end{cases} \quad \forall X \in \triangle MNK. \tag{5.20}$$

We employ the following geometrically based selection scheme for choosing the watermarked vertex for each neighbouring face:

$$\mathbf{w}_j = \begin{cases} \mathbf{v}_i + \|V_iW_j\| \cdot \left(\dfrac{\overrightarrow{V_iP_j}}{\|V_iP_j\|} \right) & \text{if } \hat{\rho}_i \in [\|OP_j\|, \|OV_i\|] \\ \mathbf{v}_i + \|V_iW_j\| \cdot \left(\dfrac{\overrightarrow{V_iQ_j}}{\|V_iQ_j\|} \right) & \text{if } \hat{\rho}_i \in [\|OV_i\|, \|OQ_j\|] \\ \mathbf{o} + \hat{\rho}_i \left(\dfrac{\overrightarrow{OP_j}}{\|OP_j\|} \right) & \text{if } \hat{\rho}_i < \|OP_j\| \\ \mathbf{o} + \hat{\rho}_i \left(\dfrac{\overrightarrow{OQ_j}}{\|OQ_j\|} \right) & \text{if } \hat{\rho}_i > \|OQ_j\| \end{cases} \tag{5.21}$$

where W_j is the point of coordinates $\mathbf{w}_j$ that satisfies the watermark distance condition $||OW_j|| = \hat{\rho}_j$. The idea behind this configuration is to find a replacement of the vertex on the original manifold surface so that the watermarked vertex does not introduce too much error. The configuration proposed in equation (5.21) considers all the possible cases. In Figure 5.1 is shown an example of the first case from the system provided in (5.21), when W_j is located inside $\triangle MNK$ and $\hat{\rho}_i \in [\|OP_j\|, \|OV_i\|]$.

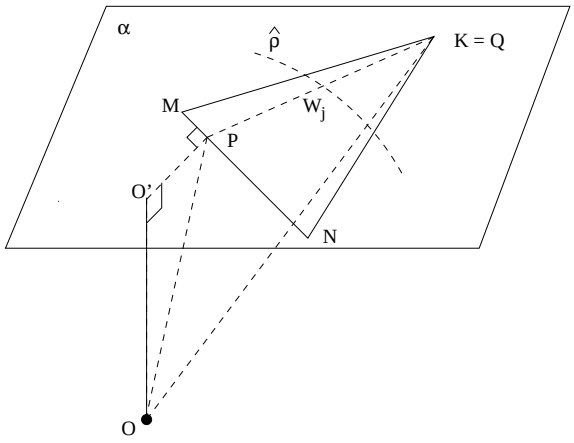


Figure 5.1: The watermarked vertex W_j is located at the intersection between the sphere centered in O, of radius $\hat{\rho}$, and $\triangle MNK$ from the surface of $\mathcal{O}$.

Thus, for each face $\mathbf{F}_j$ we have one candidate vertex $\mathbf{w}_j,\ j = 1, \ldots, N_{F_i}$ as the potential position of the watermarked vertex $\hat{\mathbf{v}}_i$. For defining the location $\hat{\mathbf{v}}_i$ of the watermarked vertex we choose the one that has the minimum quadric error $Q_{\mathbf{v}_i}(\mathbf{w}_j)$

from among all the candidates corresponding to the faces adjacent to $\mathbf{v}_i$:

$$\hat{\mathbf{v}}_i = \arg\min_{\mathbf{w}_j} Q_{\mathbf{v}_i}(\mathbf{w}_j), \quad j = 1, \ldots, N_{F_i} \tag{5.22}$$

The proposed selection scheme is a geometrically motivated approximation of the quadric error minimization problem as provided in equation (5.19), [91]. Although the algorithm makes sure that the newly selected vertex introduces a minimum distortion over all the candidates with respect to the original object surface, it does not guarantee that the new triangle will not flip-over after the vertex movement. A further simple consistency check on the surface normals is required in order to avoid such problems by ensuring that the watermarked vertex is contained in the convex hull formed by planes perpendicular on the surrounding surfaces. However, such a constraint may limit the robustness of the algorithm as in some cases the vertex will not be able to move to the position required by the embedding condition. Each neighbouring face is processed for a possible candidate $\mathbf{w}_j$ thus resulting in the complexity of $O(N_{F_i})$ for finding the location of a single watermarked vertex $\hat{\mathbf{v}}_i$.

5.5 Optimization method

In the first part of this section, we propose a novel surface distortion error function. Afterwards, we describe an iterative optimization method to find the optimal solution of vertex placement for watermarking with respect to the given surface error function.

5.5.1 Surface distortion metric

We propose an error function consisting of a set of constraints that enforce the minimum error after watermarking. The error function proposed not only measures the distortion between the watermarked surface and the original surface, but also considers the smoothness of the watermarked surface as well as the displacement error of the original vertex.

The proposed error function is defined as a product of a vector function:

$$E = \mathbf{f}^T \mathbf{f} \tag{5.23}$$

where $\mathbf{f}$ is a vector-function that contains three components :

$$\mathbf{f} = \begin{cases} \sqrt{k_1}\mathbf{f}_1 \\ \sqrt{k_2}\mathbf{f}_2 \\ \sqrt{k_3}\mathbf{f}_3 \end{cases} \tag{5.24}$$

where k_1, k_2, k_3 are the user-defined parameters which control the weights of the three error function components: $\mathbf{f}_1$, $\mathbf{f}_2$ and $\mathbf{f}_3$.

The first error metric $\mathbf{f}_1$ is used to measure the distortion of the watermarked vertex with respect to the original surface. This is defined as :

$$\mathbf{f}_1 = \begin{cases} \langle(\hat{\mathbf{v}}_i - \mathbf{v}_i),\ \mathbf{n}_1\rangle\mathbf{n}_1 \\ \vdots \\ \langle(\hat{\mathbf{v}}_i - \mathbf{v}_i),\ \mathbf{n}_j\rangle\mathbf{n}_j \end{cases} \tag{5.25}$$

where $< \cdot, \cdot >$ is the dot product and $\mathbf{n}_j,\ j = 1, \ldots, N_{F_i}$ is the normal vector of a neighbouring triangle to the vertex $\mathbf{v}_i$ of the original surface. The vector $< (\hat{\mathbf{v}}_i - \mathbf{v}_i),\ \mathbf{n}_j > \mathbf{n}_j$ is orthogonal from $\hat{\mathbf{v}}_i$ to the face $\mathbf{F}_j$. Let us define $D(\hat{\mathbf{v}}_i, \mathbf{F}_j)$ as the distance from vertex $\hat{\mathbf{v}}_i$ to face $\mathbf{F}_j$. We can observe that

$$\mathbf{f}_1^T \mathbf{f}_1 = \sum_j D^2(\hat{\mathbf{v}}_j, \mathbf{F}_j) = Q_{\mathbf{v}_i}(\hat{\mathbf{v}}_i) \tag{5.26}$$

Therefore, the first vector function is measuring the squared distance from the watermarked vertex to the original local surface and corresponds to the error from equation (5.19) as used in Section 5.4.

The second vector function $\mathbf{f}_2$ is defined to measure the distance of the water-

marked vertex to the updated surface as:

$$\mathbf{f}_2 = \begin{cases} \langle(\hat{\mathbf{v}}_i - \mathbf{v}_i),\ \hat{\mathbf{n}}_1\rangle\hat{\mathbf{n}}_1 \\ \vdots \\ \langle(\hat{\mathbf{v}}_i - \mathbf{v}_i),\ \hat{\mathbf{n}}_j\rangle\hat{\mathbf{n}}_j \end{cases} \tag{5.27}$$

where $\hat{\mathbf{n}}_j$, $j = 1, \ldots, N_{F_i}$ is the normal vector of the triangle of the modified surface neighbouring the vertex $\hat{\mathbf{v}}_i$, where we assume that the number of triangles does not change following watermarking. In case that no neighbouring vertex to $\hat{\mathbf{v}}_i$ has its location changed following watermarking, we have:

$$Q_{\mathbf{v}_i}(\hat{\mathbf{v}}_i) = \mathbf{f}_2^T\mathbf{f}_2 = \mathbf{f}_1^T\mathbf{f}_1. \tag{5.28}$$

The reason for using $\mathbf{f_2}$ is to ensure the quality of the resulting watermarked surface. When updating the vertex $\mathbf{v}_i$, some of its neighbours may have been watermarked in the previous stage. Then, the surrounding faces are no longer located on the same planes as the original ones. In this case, the smoothness of the watermarked surface can be affected if only $\mathbf{f}_1$ would have been considered for the distortion assessment.

The third error function component $\mathbf{f}_3$ corresponds to the Euclidean distance between the watermarked vertex and the original vertex locations :

$$\mathbf{f}_3 = \hat{\mathbf{v}}_i - \mathbf{v}_i \tag{5.29}$$

such that the squared Euclidean distance between the original vertex V_i and the watermarked vertex $\hat{V}_i$ is:

$$\mathbf{f}_3^T\mathbf{f}_3 = ||\hat{\mathbf{v}}_i - \mathbf{v}_i||^2 \tag{5.30}$$

The third constraint is added to compensate the error which can not be measured by the previous two functions $\mathbf{f}_1$ and $\mathbf{f}_2$. For example, any point on the plane containing the original triangle will produce no errors with respect to the first and second error functions if none of their neighbours locations are changed by watermarking. However, if the new vertex goes too far away from the original position on the plane, it will introduce a very large distortion, although $\mathbf{f}_1^T\mathbf{f}_1 = \mathbf{f}_2^T\mathbf{f}_2 = 0$ in this case. Therefore, we add the third error function as a dragging force in order to

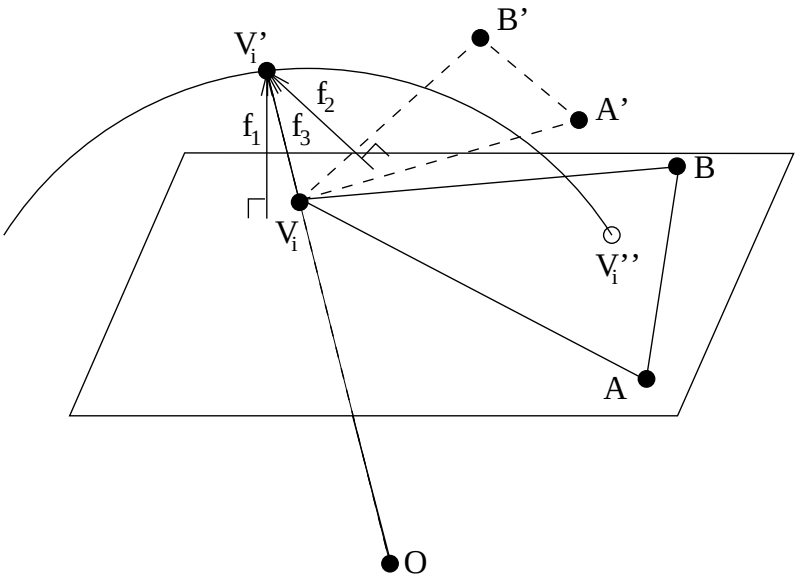


Figure 5.2: This diagram illustrates the three vector error functions when following watermarking a triangle is changed from $\triangle V_i AB$ to $\triangle V_i' A' B'$. The watermark enforces that the Euclidean distance from O to any location on the sphere is $||OV_i''|| = ||OV_i'|| = \hat{\rho}_i$. A' and B' are the watermarked neighbours of V_i' corresponding to the original A and B. $\mathbf{f}_1$ and $\mathbf{f}_2$ are the vector functions from V' to the original plane and the updated plane, respectively. $\mathbf{f}_3$ is the Euclidean distance from V_i to V_i'. In can be observed that if only $\mathbf{f}_1$ is considered, then V_i'' will introduce the smallest error. But obviously, this will introduce a big distortion to the new surface $\triangle V_i'' A' B'$.

make sure that the watermarked vertex will be displaced only as much as necessary from its original location.

Figure 5.2 displays an example illustrating the three error functions for a single triangle when changing the vertex V_i location into V_i', following watermarking. If only the original surface is considered, *i.e.* the plane containing $\triangle V_i AB$, then V_i'' introduces the smallest distortion. But obviously, the new triangle $\triangle V_i'' A' B'$ is twisted and can result into unpleasant visual results of the watermarked surface. Such a movement will be prevented by employing the second error vector function $\mathbf{f}_2$. The watermarked position of the new vertex must consider the original surface and the updated surface simultaneously in order to ensure an appropriate surface smoothness after watermarking. The user defined parameters k_1, k_2 and k_3 weigh the error with respect to the original surface, watermarked object surface and the Euclidean distance between the original and updated vertices, respectively. The user can adjust the configuration in order to focus on a specific error according to different

requirements. A comprehensive empirical study about choosing these parameters is provided in the experimental results section. In the following section, we describe the optimization watermarking algorithm using the Levenberg-Marquardt method.

5.5.2 Optimization methodology for optimal vertex placement for watermarking

In this section we describe how we minimize the error function defined in equation (5.23) in order to find the best vertex position minimizing the error E. The constraint that we have to enforce is $||O\hat{V}|| = \hat{\rho}$, *i.e.* the watermarked vertex should be on the sphere centered in the object center O. Let us consider the spherical coordinate representation for the vertex $\hat{\mathbf{v}}$ which has an explicit term of the vertex norm $\hat{\rho}$:

$$\hat{\mathbf{v}} = \begin{pmatrix} \hat{\rho}\cos\hat{\phi}\sin\hat{\theta} \\ \hat{\rho}\sin\hat{\phi}\sin\hat{\theta} \\ \hat{\rho}\cos\hat{\theta} \end{pmatrix} \tag{5.31}$$

Thus, we can modify the vector $\psi = \begin{pmatrix} \hat{\phi} \\ \hat{\theta} \end{pmatrix}$ while the vertex norm $\hat{\rho}$ is constant, *i.e.* the corresponding angles of $\mathbf{v}$ are changed while the vertex is still located on the same sphere. Therefore, we only need to find the best vector ψ such that E is minimized :

$$\psi = \underset{\psi}{\operatorname{argmin}}\ E \tag{5.32}$$

Initially, we can select any point on the sphere as an initial value. We choose to move the vertex $\mathbf{v}$ to $\hat{\mathbf{v}}$ along the direction of $\overrightarrow{OV}$ such that $||O\hat{V}|| = \hat{\rho}$, as in [31], as a suitable initialization. Then, we use the Levenberg-Marquardt method to iteratively find the optimal vertex location minimizing the surface error defined according to (5.23). Levenberg-Marquardt [92] is a damped version of the Gauss-Newton method and represents an iterative gradient-descend method which solves nonlinear least square problems subject to constraints. This method has been used as an optimization method in computer graphics [28, 47], computer vision as well as for image watermarking [8].

Levenberg-Marquardt firstly linearizes the given nonlinear problem by using the

Taylor expansion around the vector ψ:

$$\mathbf{f}(\psi + \mathbf{h}) = \mathbf{f}(\psi) + \mathbf{J}\mathbf{h} \tag{5.33}$$

where $\mathbf{f}$ is defined in Equation (5.24) and $\mathbf{h} = \begin{pmatrix} \Delta\phi \\ \Delta\theta \end{pmatrix}$ is the step size. $\mathbf{J}$ is the Jacobian Matrix of the vector function $\mathbf{f}$ and is calculated as:

$$\mathbf{J} = \begin{pmatrix} \frac{\partial \mathbf{f}^T}{\partial \hat{\phi}} \\ \frac{\partial \mathbf{f}^T}{\partial \hat{\theta}} \end{pmatrix} \tag{5.34}$$

An optimal step $\mathbf{h}$ is evaluated at each iteration k, in order to update the vector ψ_k :

$$\psi_{k+1} = \psi_k + \mathbf{h} \tag{5.35}$$

Levenberg-Marquardt method uses the following equation to calculate an optimal $\mathbf{h}$ at the given iteration [92]:

$$(\mathbf{J}^T\mathbf{J} + \mu\mathbf{I})\mathbf{h} = \mathbf{J}^T\mathbf{f} \tag{5.36}$$

while $\mu > 0$ is called damping factor. The initial value of μ is chosen to be 10^{-6} times the largest value on the diagonal component of $\mathbf{J}^T\mathbf{J}$, evaluated at ψ_0, as suggested in [47, 92]. The μ value is updated according to the schedule proposed by Nielsen in [95]. The algorithm is not sensitive to the initial value of μ as this is continually optimized by the updating procedure [92]. The damping factor serves two main purposes. Firstly, as long as $\mu > 0$, the coefficient matrix is positive definite, and this ensures that $\mathbf{h}$ is calculated in order to ensure minimizing in the descent error direction. Secondly, the damping parameter influences both the step size and the direction of the gradient descent. When ψ is close to the optimal position, the convergence rate of the Levenberg-Marquardt is almost quadratic. The details of the Levenberg-Marquardt method can be found in [92].

Optimization using Levenberg-Marquardt was shown to converge and the process is terminated in the following situations [47, 92]: when the step size $\mathbf{h}$ is too small, the error E is too small, or when the loop exceeds a pre-set number of iterations,

according to the following thresholds :

$$\begin{cases} ||\mathbf{h}_k|| \leq 10^{-8}(||\mathbf{x}_k|| + 10^{-8}) \\ ||\mathbf{f}_k^T \psi_k||_\infty \leq 10^{-8} \\ k \geq k_{max} \end{cases} \tag{5.37}$$

where the subscript k indicates the iteration step, k_{max} is the maximum allowed number of iterations and $||\mathbf{f}_k^T \psi_k||_\infty$ represents the infinity norm of the matrix $\mathbf{f}_k^T \psi_k$ (the largest absolute value of all the matrix entries).

5.6 Experimental results

In the following we provide experimental results when using the proposed methodology for watermarking graphical objects. The experiments address both the visibility of the watermarks as well as their robustness to various attacks. In the simulations we use both the discretization method described in Section 5.4 (denoted as QSP) as well as the error minimization approach by employing Levenberg-Marquardt as described in Section 5.5 (denoted as L-M). For each of these methods we employ two different statistical approaches, corresponding to modifying the mean or the variance of the histogram of distances from vertices to the object center, according to either equation (5.7) or (5.11), respectively. The discretization method proposed in [90] , described in Section 5.4, has been shown to produce less visible changes in 3D objects and to possess similar robustness when compared to the methods proposed by Cho *et al.* in [31]. Moreover, the placement of the vertex proposed in [31] is used as the initialization of the Levenberg-Marquardt method. The optimization methodology described in Section 5.5 clearly leads to a better solution for the vertex placement following watermarking, such that it minimizes the energy function as defined in Section 5.5.1, and consequently produces less distortion in the graphical object than the approach from [31].

Models	No. of vertices	No. of faces
Bunny	34,833	69,449
Fish	64,982	129,664
Gear	231,703	463,430
Dragon	422,335	844,886

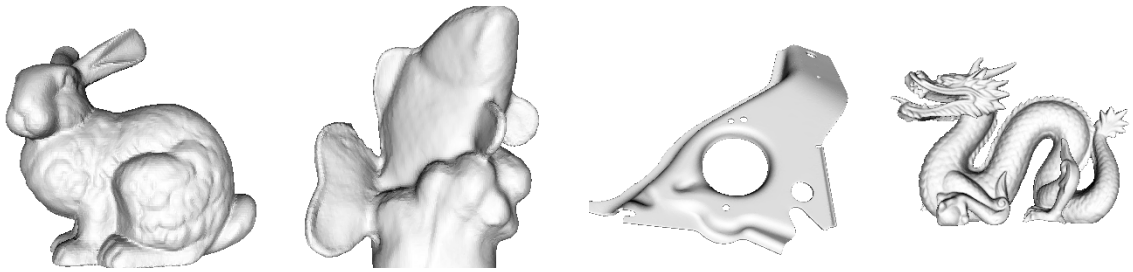
Table 5.1: Characteristics of the 3D models used in experiments.

Parameters			Bunny		Gear	
k_1	k_2	k_3	L-MMean	L-MVar	L-MMean	L-MVar
1.0	0.0	0.0	0.44	0.17	949.09	428.09
0.0	1.0	0.0	0.46	0.18	1025.21	355.11
0.0	0.0	1.0	1.18	0.62	1861.15	1024.07
0.5	0.5	0.0	0.41	0.18	654.57	303.15
0.5	0.0	0.5	1.10	0.59	1546.87	862.52
0.0	0.5	0.5	1.17	0.62	1817.69	998.94
0.45	0.45	0.1	0.69	0.36	621.97	313.10
0.4	0.4	0.2	0.88	0.46	1007.14	537.90
0.49	0.49	0.02	0.40	0.21	410.57	148.12

Table 5.2: Evaluating MRMS error, calculated according to equation (2.16), when changing k_1, k_2, k_3.

5.6.1 Experimental Models

In this chapter, we use the following four 3D graphical models : Bunny, Fish, Gear and Dragon. The details of the models are provided in Table 5.1. Figure 5.3 shows the rendered models. Bunny is a standard small mesh object which is widely used in computer graphics experiments. Gear is a CAD object which contains large flat regions and sharp corners. Dragon is considered as a large 3D model as number of vertices and faces.



(a) Bunny (b) Fish (c) Gear (d) Dragon

Figure 5.3: 3D Models used in the experiments.

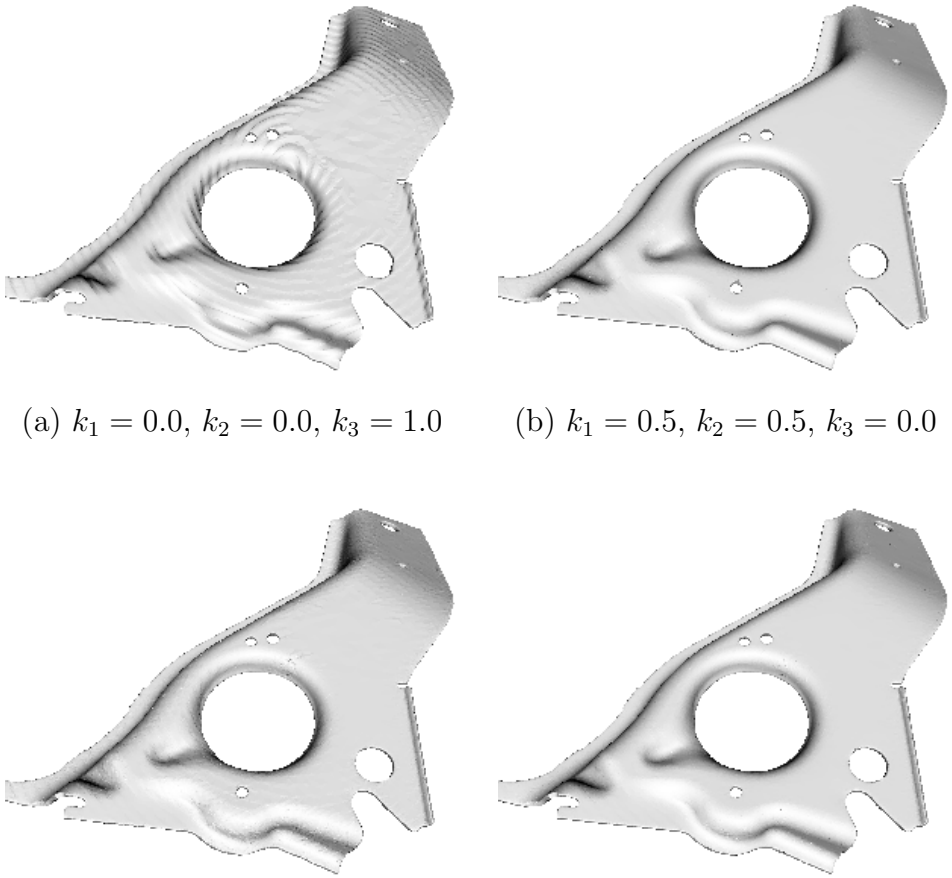
5.6.2 Evaluating the parameter setting

In this section we provide results about how to choose the parameters weighting the error components used in the optimization process as described in Section 5.5.1. There are 64 bits message embedded in the objects and the $\alpha = 0.1$ in these cases. The parameters k_1, k_2 and k_3 weigh the significance of the constraints characterizing the errors of the updated vertex with respect to the original surface, watermarked surface and the Euclidean distance to the original vertex location, respectively. A number of significant parameter combinations (k_1, k_2, k_3), emphasizing one or another of the error components, are listed in the first column of Table 5.2. L-MMean and L-MVar denote the watermarking methods using the error optimization by Levenberg-Marquardt, as described in Section 5.5, when employing either equation (5.7) for mean change, or equation (5.11) for variance change. The results are evaluated according to the distortion caused to the 3D object surface, according to MRMS error, provided in equation (2.16). Table 5.2 provides the MRMS results for two of the graphical objects: Bunny and Gear. The first three sets of parameters consider each a single error component while equating the other two to zero. In these cases, when only the distortion with respect to the original surface or to the updated surface is considered, the resulting surface distortion results are similar. However, when considering only the Euclidean distance with respect to the original vertex, *i.e.* when $k_3 = 1$ and $k_1 = k_2 = 0$, the resulting surface distortion is much larger. For this parameter combination it can be observed that the watermarking method becomes identical with the method of Cho *et al.* described in [31]. From the fourth case to the sixth case, we test the effects considering only two parameters while equating the third one to zero. It shows that if the errors with respect to the original surface and the watermarked surface are considered simultaneously and equally weighted, the resulting distortion is much smaller than the other two cases. This indicates that the significance of the first two error function components $\mathbf{f}_1$ and $\mathbf{f}_2$ should outweigh the third error component, respectively $\mathbf{f}_3$. As it is described in Section 5.5.1, the errors with respect to both the original surface and the updated surface must be considered. So the fourth case provides less surface distortion than the first and the second ones according to these experiments. The final three pa-

rameter sets evaluate the distortion when we take all three factors into account with various weights. We found that the last case, where $k_1 = k_2 = 0.49$ and $k_3 = 0.02$, provides the minimum surface distortion according to the MRMS criterion. This result verifies our error function construction, as described in Section 5.5.1, with the main emphasize on minimizing the error with respect to the original object surface as well as for enforcing the smoothness of the resulting watermarked object surface, but without neglecting the error with respect to the original vertex location. Figure 5.4 illustrates various visual effects on the Gear object when using different configurations of k_1, k_2 and k_3. According to the parameter settings, both Cho *et al.* and the QSP methods are special cases of the proposed L-M based methodology. In the rest of the chapter, we consider the configuration of (k_1, k_2, k_3)=$(0.49, 0.49, 0.02)$ for the weighting parameters because this provides the best watermarked object surface quality. Unless, specifically stated otherwise we embed $M = 64$ bits in all the watermarking experiments. The robustness parameter is set as $\alpha = 0.1$ in both equations (5.7) and (5.11), while the maximum number of iterations for the Levenberg-Marquardt method is set at $k_{max} = 500$ in the convergence criteria system from (5.37).

5.6.3 Evaluation of surface distortion

One of the main requirements of digital watermarking is that it does not produce visible changes on the surface of graphical objects. The watermarking methods developed by Cho *et al.* [31] are known to produce ripples like effects on the surface of graphical objects. The methodology presented in this chapter aims to specifically address the distortions produced by digital watermarking in the meshes of graphical objects. Table 5.3 compares the distortions introduced by the two methods proposed in this chapter and Cho's methods under the same parameter settings. For each method we use both statistical approaches corresponding to equation (5.7) for mean change and equation (5.11) for variance change. The names of the methods are obtained by appending either "Mean" or "Var", respectively to the name of the watermarking method. We use MRMS, proposed in [34] and provided in equation (2.16), as the numerical distortion measure for comparing various watermarking



(a) $k_1 = 0.0$, $k_2 = 0.0$, $k_3 = 1.0$ (b) $k_1 = 0.5$, $k_2 = 0.5$, $k_3 = 0.0$

(c) $k_1 = 0.45$, $k_2 = 0.45$, $k_3 = 0.1$ (d) $k_1 = 0.49$, $k_2 = 0.49$, $k_3 = 0.02$

Figure 5.4: The visual effects for various k_1, k_2 and k_3 settings.

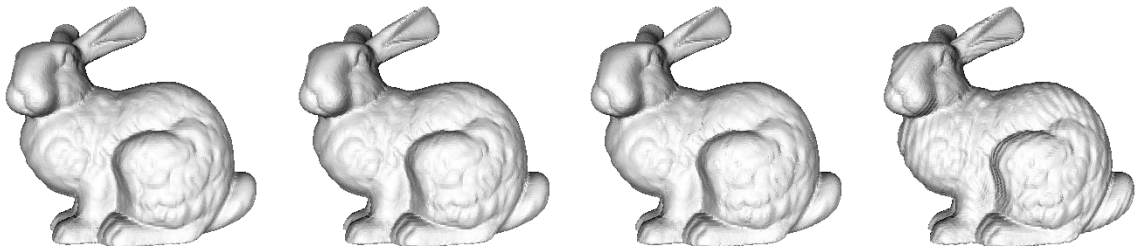
methods. Clearly, both methods proposed in this chapter outperform Cho's methods by producing a lower level of distortion in the surfaces of the watermarked graphical objects. The L-M methods and the QSP methods give similar distortion to the objects Bunny, Fish and Dragon. But L-MMean and L-MVar are clearly better than their corresponding QSP methods when applied on the Gear object which is a CAD object containing many flat regions and sharp angles.

Figure 5.5 shows the visual differences among the results provided by the three methods: L-M, QSP and Cho. The two methods proposed in this chapter produce much smaller distortion than Cho's method. Figure 5.6 compares close details of the distortions produced by the two proposed methods. In general, the visual quality of the watermarked objects by the two proposed methods is very close to each other but the M-L optimization based methods produce slightly better results than the

Object	L-MMean	L-MVar	QSPMean	QSPVar	ChoMean	ChoVar
Bunny	0.40	0.21	0.43	0.25	1.18	0.62
Fish	0.12	0.06	0.15	0.07	0.48	0.24
Gear	409.59	148.16	679.46	212.11	1860.67	1023.67
Dragon	0.29	0.13	0.36	0.15	1.09	0.57

Table 5.3: Watermarked object distortion measured by MRMS, where all the figures should be multiplied with 10^{-4}.

QSP based methods.



(a) Original object (b) L-MMean (c) QSPMean (d) ChoMean

Figure 5.5: Distortions produced by watermarking on the Bunny object.

5.6.4 Watermark robustness assessment

The results are obtained under 64 bits message embedded and $\alpha = 0.1$ in this section. The Levenberg-Marquardt method described in Section 5.5 is used for the vertex placement at a watermark defined distance from the object center while employing the statistical approach of Cho's method [31] as initialization. The optimization method by L-M ensures that the vertex is placed optimally according to the given error measure as described in Section 5.5. In [91] it is shown that the robustness of the QSP methods is very similar with that of Cho's methods. Given these aspects of the proposed methodology, in this study we only compare the robustness of the L-M method with that of the QSP methods. All these methods are perfectly robust against attacks that do not distort the graphical object surface including the affine transformations, vertex reordering etc. In the following, we evaluate the robustness against additive noise, Laplacian smoothing, mesh simplification, quantization and uniform resampling. Figure 5.7 shows the effects on the Bunny model after certain attacks. All those models are embedded with 64 bits of message while $\alpha = 0.1$.

Figure 5.7(a) shows the watermarked Bunny graphical object after noise addition with $\varepsilon = 0.5\%$. And the noise attack is defined in equation 3.24 on page 59. The plots from Figure 5.8 show the robustness against noise when varying $\varepsilon = [0.1\%, 1\%]$ for all four methods and for all four graphical objects under consideration. The twisting curves indicate that the robustness of the L-M methods and those of the QSP methods are similar to each other for each of the statistical cases of either "Mean" or "Var". The results on the Gear object are not that robust as those achieved when attacking the other three objects because this is a CAD object containing large flat areas and sharp angles. Additive noise can easily destroy completely such distinctive features.

For the smoothing attack test we use the Laplacian algorithm proposed in [119]. A watermarked and smoothed Bunny when considering a smoothing parameter $\lambda = 0.5$ and 10 iterations is shown in Figure 5.7(b). The robustness of the watermarking methods against the Laplacian smoothing when applied for 1 to 20 iterations with $\lambda = 0.5$ are provided in Figure 5.9 for the four objects. Again, the proposed methods show similar results to each other. It can be observed that as the size of the object, measured by the number of vertices, is increasing, the robustness is increased accordingly. The watermarked Bunny, which has the fewest number of vertices, is the least robust to the Laplacian smoothing attack among the four objects.

The quadric metric simplification software described in [50] was used for testing the robustness at mesh simplification. Figure 5.7(c) shows the watermarked Bunny object after 90% simplification. Figure 5.10 shows the robustness to the simplification attack for the four methods and for the given four graphical objects. In these experiments, we test the robustness of the watermarking methods when varying the mesh simplification ratio from 5% to 95%. The proposed methodology performs excellently in the case of the mesh simplification attack. According to the results from Figure 5.10, L-M methods provide better results than QSP methods. This is because the L-M methods approximate the watermarked object surface better than the QSP methods while the mesh simplification attack always tends to preserve the surface features.

Figure 5.7(d) shows the bunny object attacked after 7 bits quantization. As

shown in Figure 5.11 all four algorithms are fairly robust up to applying 8 bits quantization attacks and the histogram mean change methods when using (5.7) perform better than histogram variance change methods when using (5.11), similarly to the noise attack tests.

We compare the robustness of all four methods against the remeshing attack after uniform sampling a percentage of 100%, 80%, 60%, 40% and 20% from all vertices of the original object using the method proposed in [13]. The new points are sampled from the tangent plane to the object surface. From the robustness plots shown in Figure 5.12 we can see that while for Bunny L-M and QSP methods provide similar results, L-MMean gives better robustness results than the other methods when used for watermarking the Fish, Gear and Dragon.

5.6.5 Computational complexity

Table 5.4 shows the comparison of the timing required for embedding 64 bits when using the three methods L-M, QSP and Cho from [31] and their Mean and Var variants. The first two columns of Table 5.4 provide the average number of iterations required by the Levenberg-Marquardt method for each of the four models. In Figure 5.13(a) we provide the convergence of the step function $\|h\|$ as well as that of the minimization of the infinity norm $||\mathbf{f}_k^T \psi_k||_\infty$, which represents the component of the matrix $\mathbf{f}_k^T \psi_k$ with the largest absolute value from all its entries, evaluated for a typical vertex which is watermarked by using L-M. From these results it can be observed that L-M method converges in just a few iterations. Figure 5.13(b) shows the variation of the error function E. The Levenberg-Marquardt method automatically adjusts the step size and the step direction $\mathbf{h}$ while testing whether the step size is appropriate. In this example, although there are nine iterations required in total for convergence, the vertex is actually only moved once at step 5 where both the $||\mathbf{f}_k^T \psi_k||_\infty$ and the error E are decreased. Then, the process terminates as the method eventually reaches the stopping condition at step 9, *i.e.* when it reaches the local minimum. The less distortion is introduced by the watermarking method, the larger is the required execution time.

Object	Mean. Iter.(No)	Var. Iter.(No)	L-M Mean(s)	L-M Var(s)	QSP Mean(s)	QSP Var(s)	Cho Mean(s)	Cho Var(s)
Bunny	6.20	5.74	5.75	5.3	1.7	1.79	0.39	0.4
Fish	5.69	5.03	10.28	9.36	3.29	3.25	0.85	0.88
Gear	5.87	5.24	42.55	38.52	12.96	12.69	3.98	3.87
Dragon	5.88	5.35	77.55	69.78	22.82	22.4	6.09	6.67

Table 5.4: Comparison of the watermark computational requirements. The second and third columns provide the average number of iterations required by the Levenberg-Marquardt method until convergence for the Mean and Var methods. The columns fourth to ninth represent processing times in seconds.

5.6.6 Evaluation of the bit-capacity and of the embedding strength factor α

Figure 5.14(a) illustrates the relations between the 3-D object surface distortion and the watermark strength factor α. Figure 5.14(b) shows the distortion variation with respect to the watermarking capacity. As it was expected the distortion is increased when either α or the embedding capacity is increased. Figure 5.16 (a) and (b) provide the robustness plots for L-MMean and QSPMean methods, respectively, when increasing the bit capacity to be embedded in the graphical object. Figures 5.15 (a) and (b) provide the robustness plots for L-MMean and QSPMean methods, respectively, when increasing the watermark strength factor α as well as the relationship between the robustness and the embedding capacity. The robustness is higher when embedding fewer bits of information. However, the robustness is very similar when watermarking any of the given graphical objects for the watermark strength corresponding to $\alpha = \{0.1, 0.15, 0.2\}$.

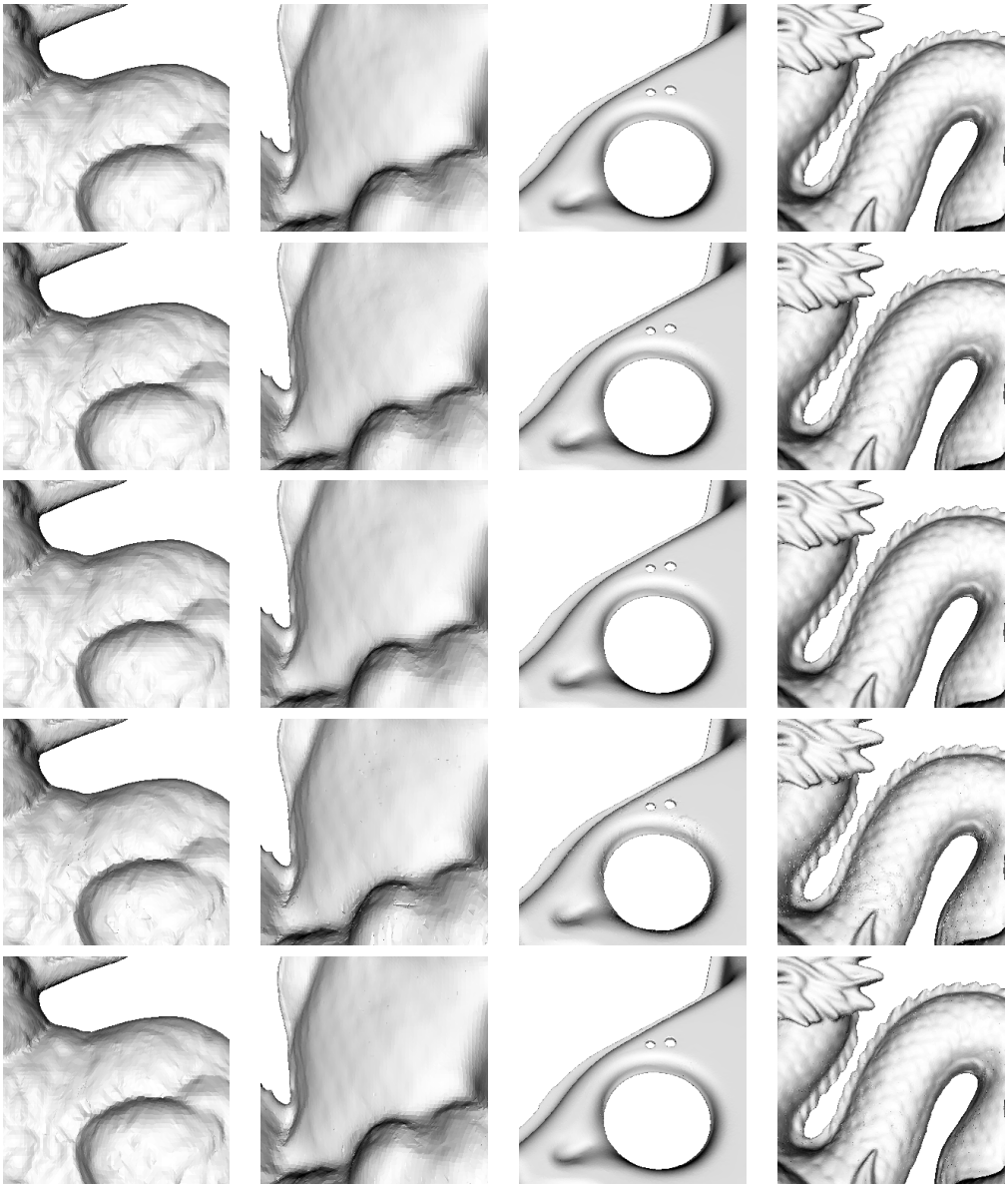
5.7 Conclusion

In this chapter, we propose two watermarking methods motivated by Cho's method. One is called QSP method based on the surface discretization. The method discretizes the surface into a number of candidate points and select the one minimizing the QEM error as the watermarked vertex. The second method uses a novel error function measuring the distortion caused by the vertex movement. The error function consists of three parts which consider the distortion with respect to the

original surface, watermarked surface and the Euclidean constraint, respectively. The surface distortion produce by the watermark is minimized by means of the Levenberg-Marquardt optimisation method. The optimization method can be considered as a generalization of the Cho's method and the QSP method. As shown in the experiments, the robustness of our methods matches Cho's method. However, the surface distortion is significantly reduced by the proposed methodology. We claim that the error function when introduced in the chapter can be considered as a standard metric in measuring the surface distortion.

To the best of our knowledge, the L-M method proposed in this chapter introduces the smallest visual and numerical distortion with respect to the original surface over all the robust 3D watermarking algorithms. The L-M method is the best method over the four methods proposed in this thesis, in the perspective of robustness and distortion. The security is also enforced in various stages including the mesh segmentation and statistical variable construction. When compared to other spatial domain methods in literature, the L-M method first time employs a mechanism to minimize the surface error with respect to a well defined surface error function.

However, the methods proposed in this chapter still suffer the same problem as the other statistical methods. The methods can not be applied on the small objects. Also, the methods are not robust against any attack modifying the object centre.



(a) Bunny (b) Fish (c) Gear (d) Dragon

Figure 5.6: Visual comparison of graphical object details following watermarking. From the first row to the bottom, the following are represented on each row: original objects and watermarked by L-MMean, L-MVar, QSPMean and QSPVar, respectively.

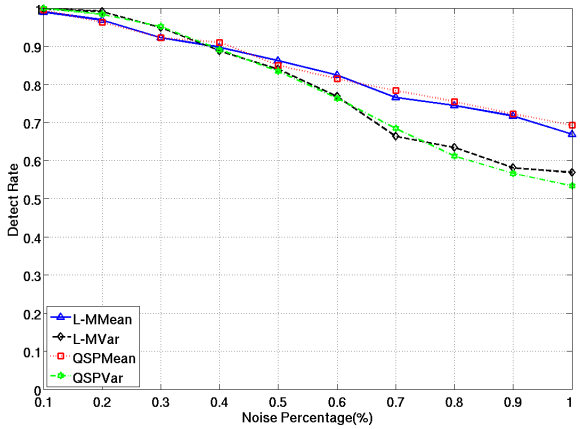
(a) Noise $\varepsilon = 0.5\%$

(b) Laplacian smoothing $\lambda = 0.5$, 10 iterations

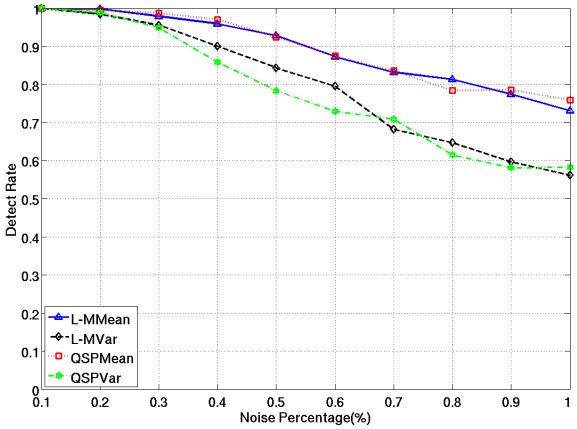
(c) 90% mesh simplification

(d) 7 bits quantization

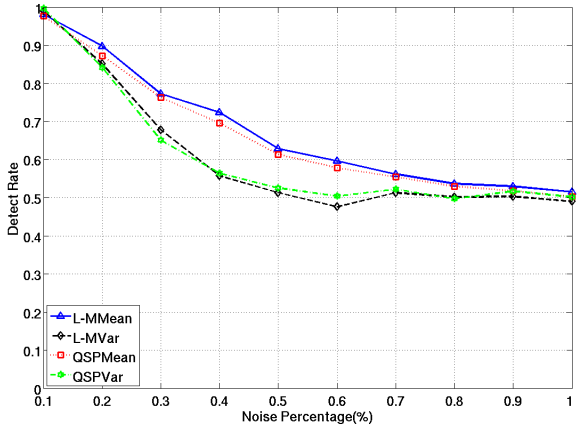
Figure 5.7: Watermarked Bunny model after various attacks.



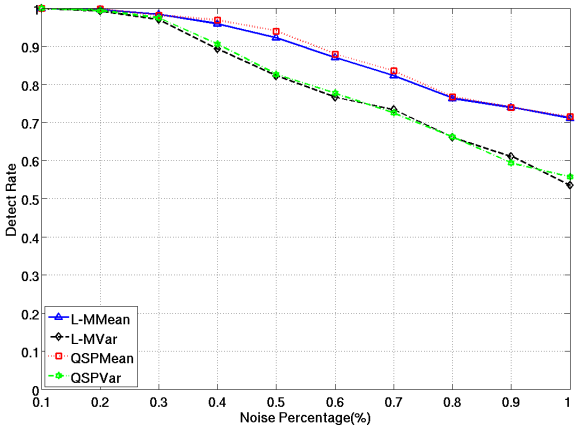
(a) Bunny



(b) Fish

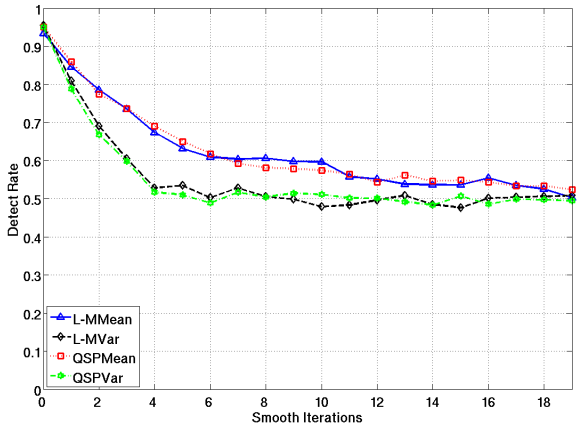


(c) Gear

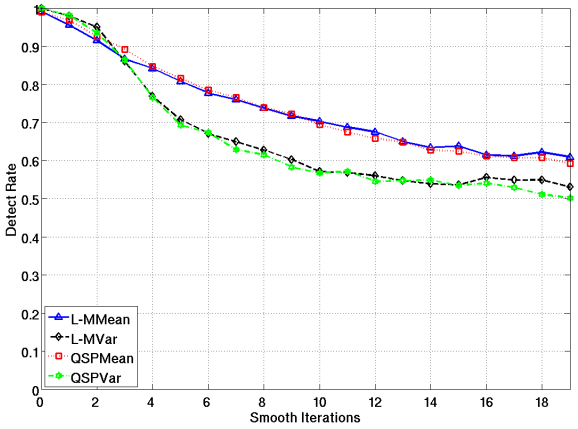


(d) Dragon

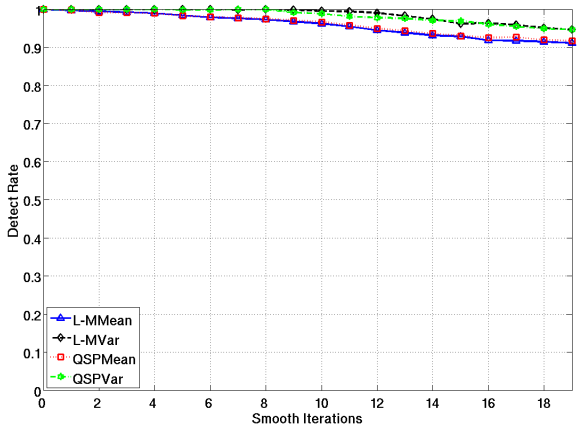
Figure 5.8: Robustness against additive noise.



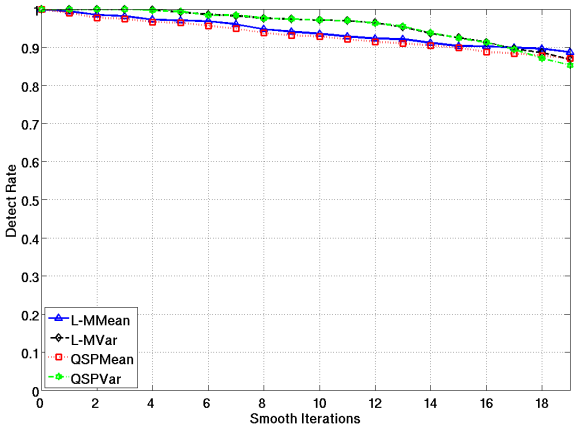
(a) Bunny



(b) Fish

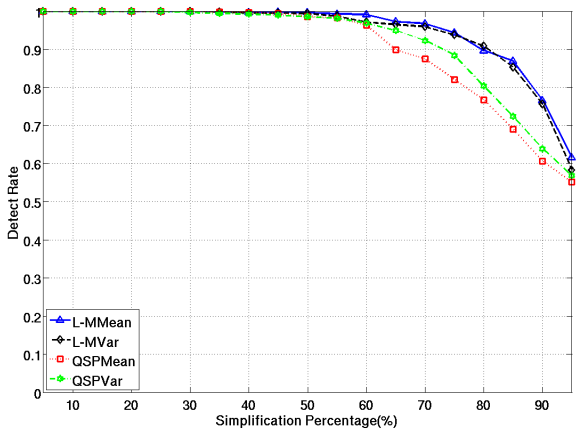


(c) Gear

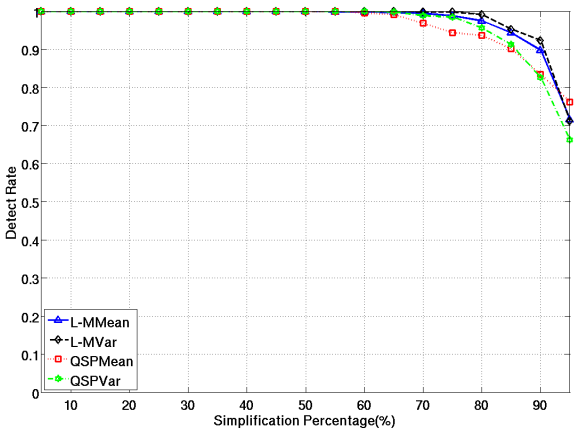


(d) Dragon

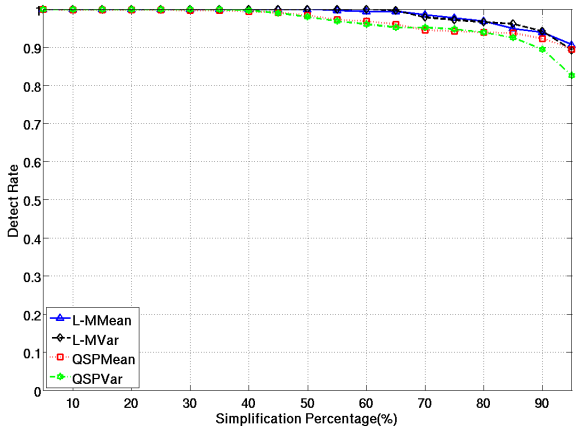
Figure 5.9: Robustness against Laplacian smoothing.



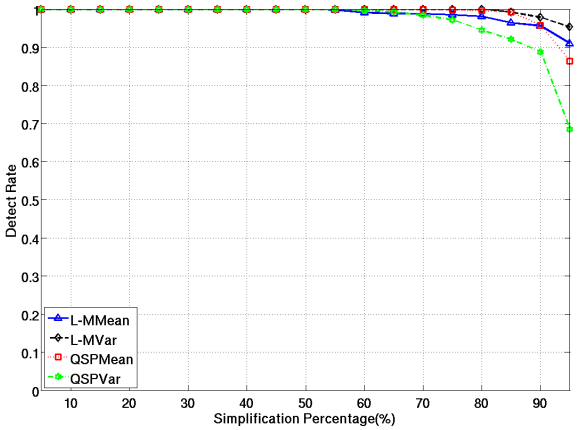
(a) Bunny



(b) Fish

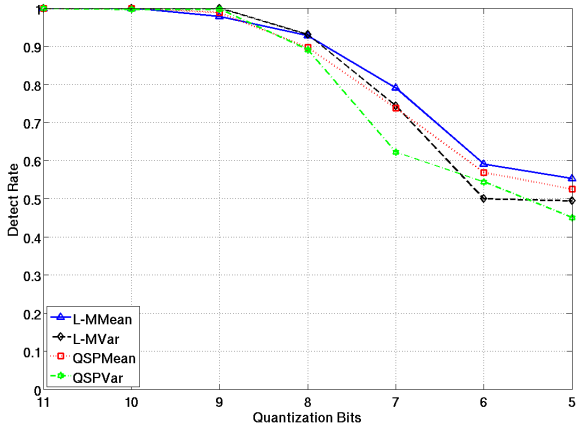


(c) Gear

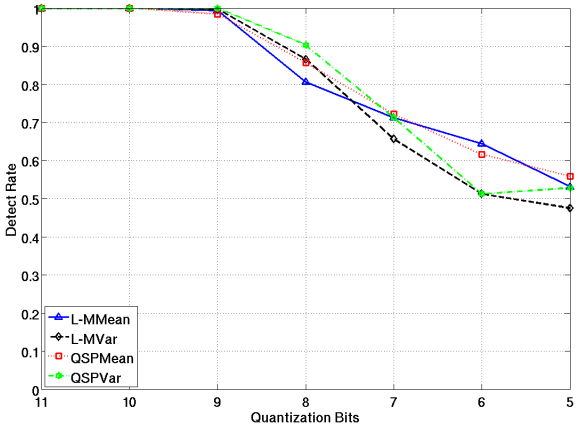


(d) Dragon

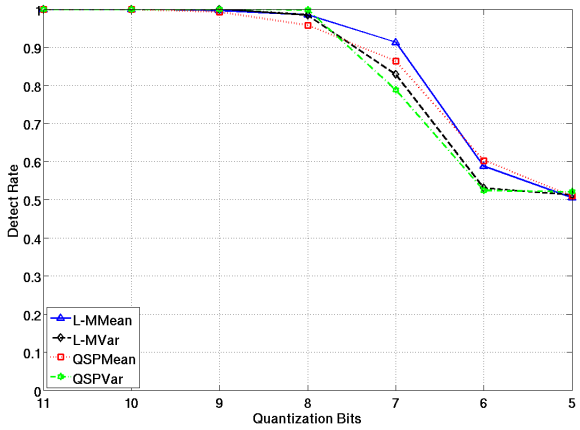
Figure 5.10: Robustness against mesh simplification.



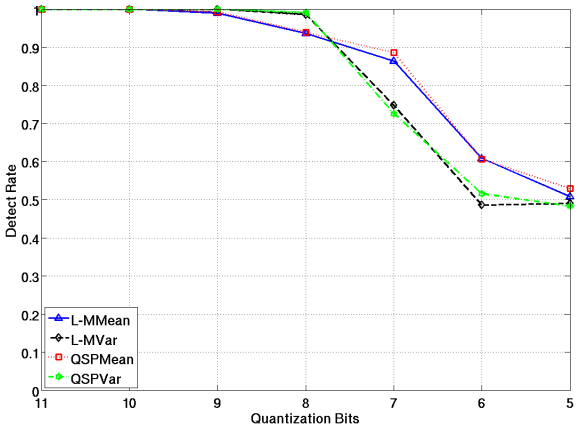
(a) Bunny



(b) Fish

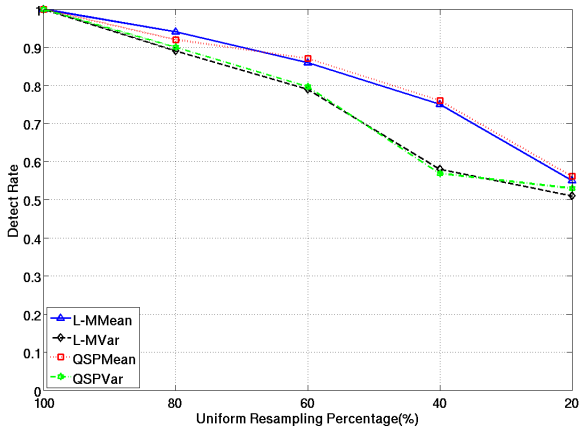


(c) Gear

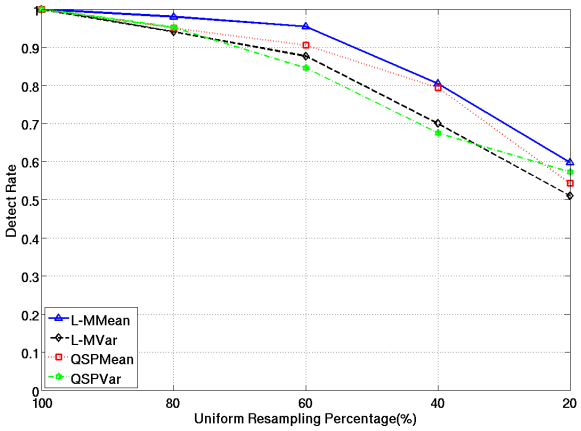


(d) Dragon

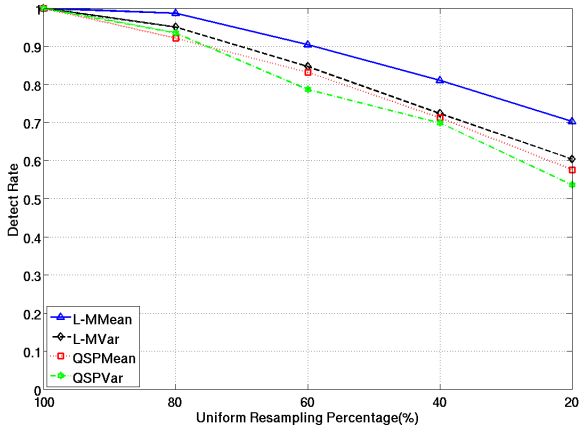
Figure 5.11: Robustness against quantization.



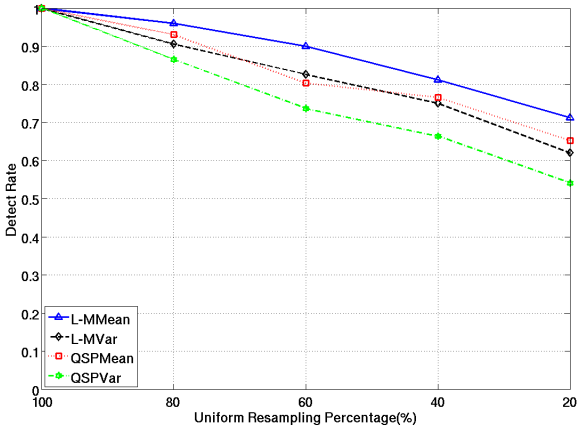
(a) Bunny



(b) Fish

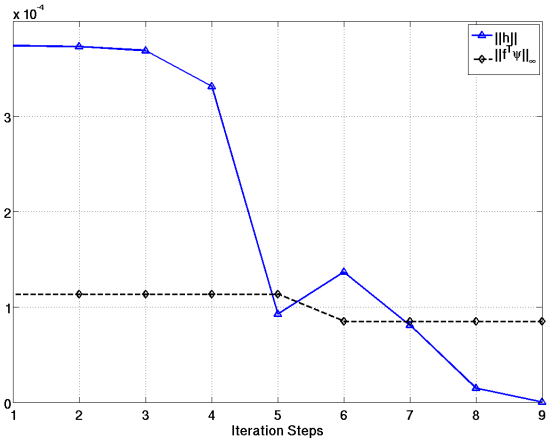


(c) Gear

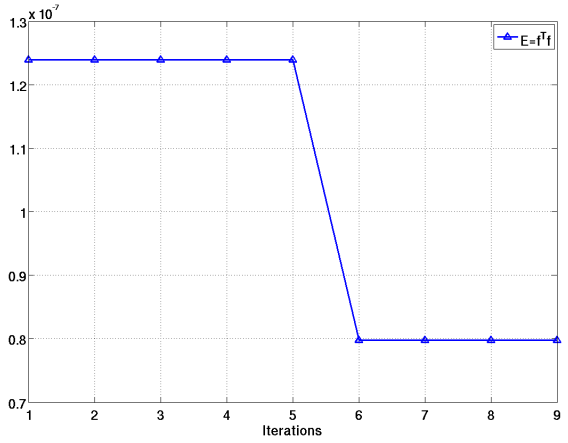


(d) Dragon

Figure 5.12: Robustness against uniform resampling and remeshing.

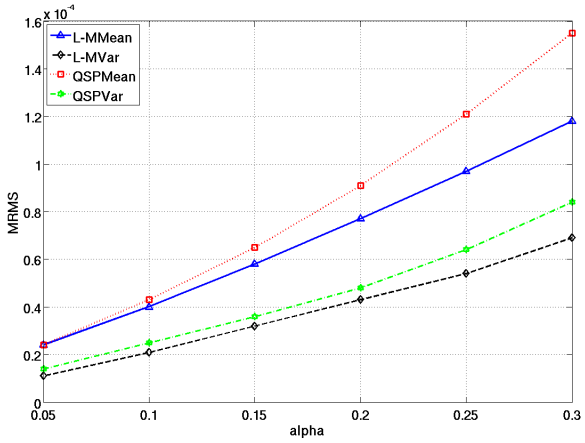


(a) The variation of the step size $||\mathbf{h}||$ and $||\mathbf{f}(\psi)||_\infty$.

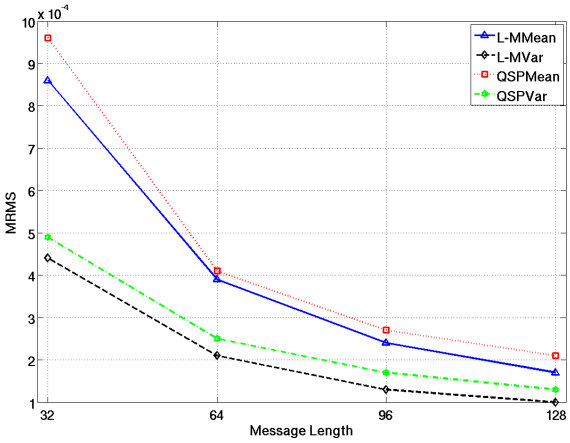


(d) The error function.

Figure 5.13: Typical convergence of Levenberg-Marquardt for a vertex.

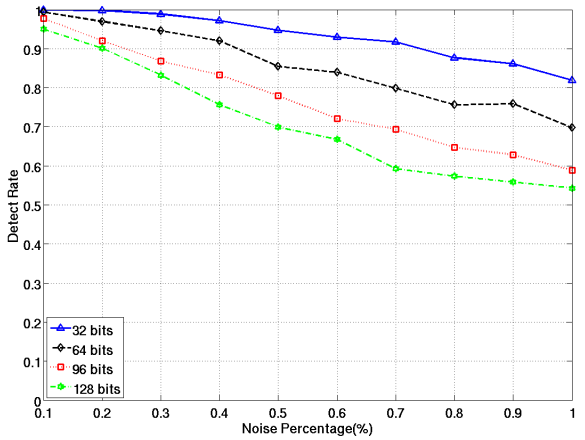


(a) Distortion when varying α

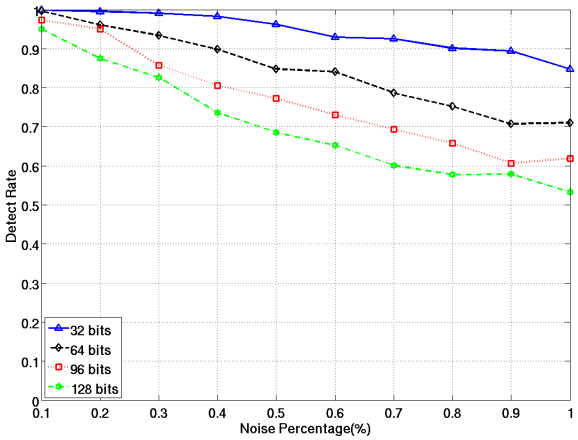


(b) Distortion when increasing capacity

Figure 5.14: Distortion with respect to the watermark strength factor α and bit-capacity.

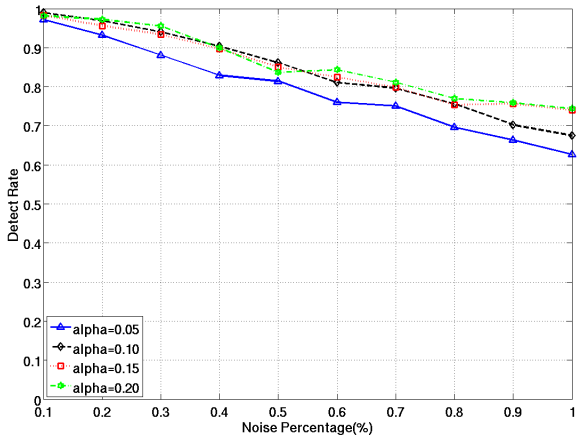


(a) Robustness of L-MMean when varying capacity

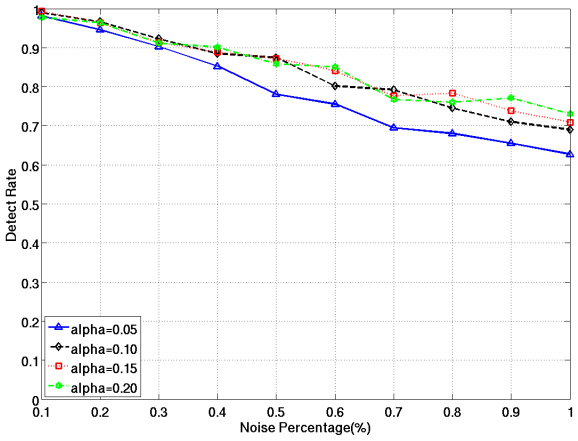


(b)Robustness of QSPMean when varying capacity

Figure 5.15: Robustness when increasing the bit capacity.



(a) Robustness of L-MMean when varying α



(b) Robustness of QSPMean when varying α

Figure 5.16: Robustness when increasing the watermark strength factor α.

Chapter 6

Conclusion and future work

This chapter commences with a summary of the contributions from Chapter 3 to Chapter 5 of this thesis. This is followed by a discussion of the strengths and weakness of the methods presented. Finally, we give the directions of the future work.

6.1 Summary of contributions

This thesis introduced four novel robust and blind watermarking methods from different directions. The first one presented in Chapter 3 is embedding the watermark message in the spectral domain. The method employs the Combinatorial Laplacian matrix to model the second derivative of the mesh based on the spectral graph theory. A robust 3D object alignment scheme using the Volumetric Moment is also introduced to ensure that the watermarked object can be oriented to the same pose as the original one. The method is the first spectral domain watermarking method that uses the statistical feature of the spectral coefficients to embed the message. The experimental results are compared with the state of the art [31] and show that our method is introducing less distortion.

In Chapter 4, we presented a novel spatial domain method using the statistical feature of the geodesic distances. The method is based on the observation that within a strip, the distribution of the geodesic distance is close to uniform. Thus, we embed the message by modifying the mean value or the variance value of the distribution. We firstly introduced a novel vertex placement scheme to move the

vertex with minimum distortion in order to change the geodesic distance. This algorithm preserves the surface feature very well and its robustness is comparable with the state of the art [31]. The geodesic method firstly shows that the flat region may be more suitable for watermarking purpose than the bumpy region [19].

In Chapter 5, we proposed two algorithms which share the same initialization as Cho's method [31] based on the well designed framework of constraint non-linear optimization problem. The first method is to discretize the configuration of the surface and generate a number of approximation candidates. The Quadric Error Metric used in mesh simplification is employed as the error function. The point introducing the smallest error will be selected as the watermarked position for a vertex. Although the point introduces the smallest error within the approximated candidates, it does not ensure the smallest error in all possibilities due to the limitation of discretization. In the second method, we minimize the surface error with respect to a novel surface error function using the Levenberg-Marquardt optimization algorithm. The error function involves three parts measuring the error introduced by the watermarked vertex with respect to the original object, watermarked object and the Euclidean distance to the original vertex, respectively. The Levenberg-Marquardt method is a gradient descend like optimization method and is used for finding the location minimizing the surface error function. The second method in Chapter 5 introduces the smallest distortion to the surface so far in the literature while its robustness matches the state of the art [31]. Another important contribution in Chapter 5 is that we bring an explicit mechanism to control the surface error while keeping the embedding strength factor unchanged.

The most important advantage over the majority of other 3D robust and blind watermarking algorithms is that the distortion introduced by our algorithms are clearly smaller. We also bridge the gap between the distortion and the robustness by introducing an explicit technique to control the distortion introduced.

On the other hand, there are certain weaknesses in each of the proposed algorithms. The spectral watermarking algorithm proposed in Chapter 3 has limited robustness against attacks. Also, the spectral algorithm involves decomposition of very large matrix which makes the algorithm unapplicable in the real world.

The geodesic distance watermarking method provides much better robustness

and visual quality in the resulting watermarked object when compared to the spectral methods. However, the robustness is still lower than that of Cho's methods. The method is computationally more expensive than Cho's method. Some of the vertices may not convergent following watermark embedding in the vertex placement.

The two methods in the Chapter 5 are probably the best 3D robust and blind watermarking algorithm so far in the literature. However, the second one involving a optimization process takes longer time to watermark an object.

In general, none of the algorithm proposed in this thesis is robust against the mesh cropping attack. Because all methods proposed rely on the object alignment in the preprocessing stage. Cropping will severely change the object centre and the principal axis. As a result, the methods proposed in this thesis can not deal with this attack.

6.2 Future work

In addressing the weaknesses detailed above, the proposed methods can be extended and improved in the following aspects. It is possible to reduce the complexity of the spectral decomposition method by employing some better matrix decomposition techniques such as proposed in [126]. It would significantly boost the usage of the mesh spectral decomposition technique if the relation is made clear, so does the watermarking technique.

The geodesic watermarking method still suffer the nonconvergent problem in some rare cases although this does not affect the robustness of the algorithm. This can be improved in the future. Once the convergence problem is solved, it is very promising to employ the VPS to design a steganography or reversible watermarking algorithm. The reversible method means that the original model can be recovered according the recovered message.

It is also worth to develop a better mesh registration method so that the watermarking algorithm can be robust against cropping attack for a certain level.

There are extensively applications that watermarking can be applied in the real world such as database management, copyright protection and authentication etc. It seems the second method proposed in Chapter 5 is the most promising method

to be applied in practice. Because it provides the best overall mesh quality and the robustness.

Bibliography

[1] Q. S. Ai, Q. Liu, Z. D. Zhou, L. Yang, and S. Q. Xie. A new digital watermarking scheme for 3d triangular mesh models. *Signal Process.*, 89(11):2159–2170, 2009.

[2] P. R. Alface and B. Macq. Blind watermarking of 3d meshes using robust feature points detection. In *Proceeding IEEE International Conference on Image Processing*, volume 1, pages 693–696, Nov. 2005.

[3] P. R. Alface, B. Macq, and F. Cayre. Blind and robust watermarking of 3D models: How to withstand the cropping attack? In *Proceeding IEEE International Conference on Image Processing*, pages 465–468, 2007.

[4] Patrice Rondao Alface, Mathieu De Craene, and Benoit B. Macq. Three-dimensional image quality measurement for the benchmarking of 3d watermarking schemes. volume 5681, pages 230–240. SPIE, 2005.

[5] P. Alliez, N. Laurent, H. Sanson, and F. Schmitt. Mesh approximation using a volume-based metric. In *Computer Graphics and Applications, 1999. Proceedings. Seventh Pacific Conference on*, pages 292–301, 1999.

[6] G. Alregib, Y. Altunbasak, and J. Rossignac. Error-resilient transmission of 3d models. *ACM Transactions on Graphics*, 24(2):182–208, 2006.

[7] Ghassan Alregib, Yucel Altunbasak, and Jarek Rossignac. Error-resilient transmission of 3d models. *ACM Trans. Graph.*, 24(2):182–208, 2005.

[8] M. Alvarez-Rodriguez and F. Perez-Gonzalez. Analysis of pilot-based synchronization algorithms for watermarking of still images. *Signal Processing - Image Communication*, 17(8):611–633, 2002.

[9] Manuel Álvarez Rodríguez and Fernando Pérez-González. Analysis of pilot-based synchronization algorithms for watermarking of still images. *Signal Processing: Image Communication*, 17(8):611 – 633, 2002.

[10] Y. Ng Andrew, I. Jordan Michael, and Yair Weiss. On spectral clustering: Analysis and an algorithm. In *Advances in Neural Information Processing Systems 14*, pages 849–856. MIT Press, 2001.

[11] D.N. Arnold, R.S. Falk, and R. Winther. Finite element exterior calculus, homological techniques, and applications. *Acta Numerica*, 15:1–155, 2006.

[12] Nicolas Aspert, Diego Santa-cruz, and Touradj Ebrahimi. Mesh: Measuring errors between surfaces using the hausdorff distance. pages 705–708, 2002.

[13] M. Attene and B. Falcidieno. Remesh: An interactive environment to edit and repair triangle meshes. In *Proceeding of the IEEE International Conference on Shape Modeling and Applications*, pages 271–276, 2006.

[14] M. Barni, F. Bartolini, V. Cappellini, and A. Piva. Copyright protection of digital images by embedded unperceivable marks. *Image and Vision Computing*, 16(12-13):897–906, 1998.

[15] M. Belkin and P. Niyogi. Laplacian eigenmaps and spectral techniques for embedding and clustering. In *Advances in Neural Information Processing Systems 14*, volume 14, pages 585–591, 2002.

[16] O. Benedens. Geometry-based watermarking of 3-D models. *IEEE Computer Graphics Applications*, 19(1):46–55, 1999.

[17] O. Benedens. Two high capacity methods for embedding public watermarks into 3D polygonal models. In *Proceeding of the Multimedia and Security Workshop at ACM Multimedia*, page 95, 1999.

[18] Zhe Bian, Shi-Min Hu, and Ralph R. Martin. Evaluation for small visual difference between conforming meshes on strain field. *J. Comput. Sci. Technol.*, 24(1):65–75, 2009.

[19] A. G. Bors. Watermarking mesh-based representations of 3-D objects using local moments. *IEEE Transaction Image Processing*, 15(3):687–701, 2006.

[20] G. Braudway, K. Magerlein, and F. Mintzer. Protecting publicly-available images with a visible image watermark. Technical Report TC-20336(89918), IBM Research Report, January 1996.

[21] T. Caelli and S. Kosinov. An eigenspace projection clustering method for inexact graph matching. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 26(4):515–519, April 2004.

[22] P. Cano, E. Batlle, T. Kalker, and J. Haisma. A review of audio fingerprinting. *Jounal of VLSI Signmal Processing Systems for Signal Image and Video Technology*, 41(3):271–284, 2005.

[23] F. Cayre and B. Macq. Data hiding on 3-D triangle meshes. *IEEE Transaction Signal Processing*, 51(4):939–949, 2003.

[24] Min-Wen Chao, Chao-hung Lin, Cheng-Wei Yu, and Tong-Yee Lee. A high capacity 3d steganography algorithm. *IEEE Transactions on Visualization and Computer Graphics*, 15(2):274–284, 2009.

[25] Mohamed Chaouch and Anne Verroust-Blondet. Alignment of 3d models. *Graph. Models*, 71(2):63–76, 2009.

[26] F.R.K. Cheng. *Spectral graph theory*. AMS, 1997.

[27] H. Cheng. A review of video registration methods for watermark detection in digital cinema applications. In *Proceedings of IEEE International Symposium on Circuits and Systems*, volume 5, pages 704–707, 2004.

[28] K. Cheng, Wang W., H. Qin, K Wong, H. Yang, and Y. Liu. Design and analysis of optimization methods for subdivision surface fitting. *IEEE Trans. on Visualization and Computer Graphics*, 13(5):878–890, 2007.

[29] K.-S.D. Cheng, Wenping Wang, Hong Qin, K.-Y.K. Wong, Huaiping Yang, and Yang Liu. Design and analysis of optimization methods for subdivision

surface fitting. *Visualization and Computer Graphics, IEEE Transactions on*, 13(5):878–890, Sept.-Oct. 2007.

[30] Yu-Ming Cheng and Chung-Ming Wang. An adaptive steganographic algorithm for 3D polygonal meshes. *VISUAL COMPUTER*, 23(9-11, Sp. Iss. SI):721–732, SEP 2007. 25th Computer Graphics International Conference (CGI), Petropolis, BRAZIL, MAY 30-JUN 02, 2007.

[31] J. W. Cho, R. Prost, and H. Y. Jung. An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms. *IEEE Transaction Signal Processing*, 55(1), 2007.

[32] W.H. Cho, M.E. Lee, H Lim, and Park S.Y. Watermarking technique for authentication of 3-d polygonal meshes. *Digital watermarking*, 55(1):259–270, 2005.

[33] C. M. Chou and D. C. Tseng. A public fragile watermarking scheme for 3D model authentication. *Computer-Aided Design*, 22(9-11), 2006.

[34] P. Cignoni, C. Rocchini, and R. Scopigno. Metro: Measuring error on simplified surfaces. *Computer Graphics Forum*, 17(2):167–174, 1998.

[35] Massimiliano Corsini, Elisa Drelie Gelasca, Touradj Ebrahimi, and Mauro Barni. Watermarked 3-d mesh quality assessment. *Multimedia, IEEE Transactions on*, 9(2):247–256, Feb. 2007.

[36] D. Cotting, T. Weyrich, M. Pauly, and M. Gross. Robust watermarking of point-sampled geometry. In *Proceedings of Shape Modeling Applications*, pages 233–242, Zurich, Switzerland, 2004.

[37] I.J. Cox, J. Kilian, T Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. Technical Report 95-10, NEC Research Institute, 1995.

[38] I.J. Cox, J. Kilian, T Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Transaction on Image Process*, 6(12):1673–1687, Dec. 1997.

[39] H. Desaulniers and N. F. Stewart. An extension of manifold boundary representations to the r-sets. *ACM Trans. Graph.*, 11(1):40–60, 1992.

[40] H. Desaulniers and N.F. Stewart. An extension of manifold boundary representations of the r-sets. *ACM Transactions on Graphics*, 11(1):40–60, 1992.

[41] M. Desbrun, M. Meyer, P. Schröder, and A.H. Barr. Implicit fairing of irregular meshes using diffusion and curvature flow. In *SIGGRAPH '99: Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 317–324, New York, NY, USA, 1999. ACM Press/Addison-Wesley Publishing Co.

[42] S. Dong, P.T. Bremer, M. Garland, V. Pascucci, and J.C. Hart. Spectral surface quadrangulation. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Papers*, pages 1057–1066, New York, NY, USA, 2006. ACM.

[43] J. L. Dugelay, A. Baskurt, M. Daoudi, and eds. *3D object processing: compression, indexing and watermarking.* J. Wiley & Sons, 2008.

[44] M Eck, T. DeRose, T. Duchamp, H Hoppe, M. Lounsbery, and W. Stuetzle. Multiresolution analsysi of arbitrary meshes. In *Proceedings of ACM SIGGRAPH 1995*, pages 173–182, 1995.

[45] Matthias Eck, Tony DeRose, Tom Duchamp, Hugues Hoppe, Michael Lounsbery, and Werner Stuetzle. Multiresolution analysis of arbitrary meshes. In *SIGGRAPH '95: Proceedings of the 22nd annual conference on Computer graphics and interactive techniques*, pages 173–182, New York, NY, USA, 1995. ACM.

[46] J.J. Eggers, R. Bauml, R. Tzschoppe, and B. Girod. Scalar costa scheme for information embedding. *Signal Processing, IEEE Transactions on*, 51(4):1003–1019, Apr 2003.

[47] M Eigensatz, R.W. Sumner, and M. Pauly. Curvature-domain shape processing. In *Proc. of Eurographics, Computer Graphics Forum, vol. 27, no. 2*, pages 241–250, 2008.

[48] A. Elad and R. Kimmel. On bending invariant signatures for surfaces. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(10):1285–1295, Oct. 2003.

[49] Thomas Funkhouser, Patrick Min, Misha Kazhdan, Joyce Chen, Alex Halderman, David Dobkin, and David Jacobs. A search engine for 3d models. *ACM Transactions on Graphics*, 2003.

[50] M. Garland and P. Heckbert. Surface simplification using quadric error metrics. In *Proceeding SIGGRAPH, Graphical Models 66(6)*, pages 370–397, 1997.

[51] E.D. Gelasca, T. Ebrahimi, M. Corsini, and M. Barni. Objective evaluation of the perceptual quality of 3d watermarking. In *Image Processing, 2005. ICIP 2005. IEEE International Conference on*, volume 1, pages I–241–4, Sept. 2005.

[52] G.H. Golub and C.F. Van Loan. *Matrix Computations*. Johns Hopkins University Press, third edition edition, 1996.

[53] Craig Gotsman, Xianfeng Gu, and Alla Sheffer. Fundamentals of spherical parameterization for 3d meshes. In *SIGGRAPH '03: ACM SIGGRAPH 2003 Papers*, pages 358–363, New York, NY, USA, 2003. ACM.

[54] E.W. Grafarend and A.A. Ardalan. World geodetic daum 2000. *Journal of Geodesy*, 73(11):611–623, 1999.

[55] Eitan Grinspun and Adrian Secord. Introduction to discrete differential geometry: the geometry of plane curves. In *SIGGRAPH '06: ACM SIGGRAPH 2006 Courses*, pages 1–4, New York, NY, USA, 2006. ACM.

[56] A. Guezlec. Meshsweeper: dynamic point-to-polygonal mesh distance and applications. *Visualization and Computer Graphics, IEEE Transactions on*, 7(1):47–61, Jan-Mar 2001.

[57] Thomas Harte and A.G. Bors. Watermarking 3d models. In *Proceeding of IEEE International Conference on Image Processing*, pages 661–664, 2002.

[58] F. Hartung and B. Girod. Copyright protection in video delivery networks by watermarking of pre-compressed video. In *Lecture Notes in Computer Science*, volume 1242, pages 423–436. Springer, 1997.

[59] P. Heckbert and M. Garland. Optimal triangulation and quadric-based surface simplification. *Journal of Computational Geometry*, 14(1-3):49–65, 1999.

[60] K. Hildebrandt, K. Polthier, and M. Wardetzky. On the convergence of metric and geometric properties of polyhedral surfaces. *Journal Geometriae Dedicata*, 123(1):89–112, 2006.

[61] Hugues Hoppe. Progressive meshes. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 99–108, New York, NY, USA, 1996. ACM.

[62] Roland Hu, Patrice Rondao-Alface, and Benoit Macq. Constrained optimisation of 3d polygonal mesh watermarking by quadratic programming. In *ICASSP '09: Proceedings of the 2009 IEEE International Conference on Acoustics, Speech and Signal Processing*, pages 1501–1504, Washington, DC, USA, 2009. IEEE Computer Society.

[63] DP HUTTENLOCHER, GA KLANDERMAN, and WJ RUCKLIDGE. COMPARING IMAGES USING THE HAUSDORFF DISTANCE. *IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE*, 15(9):850–863, SEP 1993.

[64] V. Jain and H. Zhang. A spectral approach to shape-based retrieval of articulated 3d models. *Comput. Aided Des.*, 39(5):398–407, 2007.

[65] J. Jeon, S.K. Lee, and Y.S. Ho. A three-dimensional watermarking algorithm using the dct transform of triangle strips. *Digital Watermarking*, pages 269–285, 2004.

[66] A.D. Kalvin and R.H. Taylor. Superfaces: polygonal mesh simplification with bounded error. *Computer Graphics and Applications, IEEE*, 16(3):64–77, May 1996.

[67] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3d polygons using multiresolution wavelet decomposotion. In *Proceeding of International Workshop on Geometric Modeling: fundamentals and applications*, pages 296–307, Tokyo, Japan, 1998.

[68] Z. Karni and C. Gotsman. Spectral compression of mesh geometry. *Computer Graphics (Proceeding SIGGRAPH)*, pages 279–286, 2000.

[69] Michael Kazhdan. *Shape Representations and Algorithms for 3D Model Retrieval*. PhD thesis, Princeton University, 2004.

[70] M.S. Kim, S. Valette, H.Y. Jung, and R. Prost. Watermarking of 3d irregular meshes based on wavelet multiresolution analysis. *Lecture Notes in Computer Science, Digital Watermarking*, pages 313–324, 2005.

[71] Sun-Jeong Kim, Soo-Kyun Kim, and Chang-Hun Kim. Discrete differential error metric for surface simplification. In *PG '02: Proceedings of the 10th Pacific Conference on Computer Graphics and Applications*, page 276, Washington, DC, USA, 2002. IEEE Computer Society.

[72] R. Kimmel and J. A. Sethian. Computing geodesic paths on manifolds. In *Proceeding National Academic Science USA*, volume 95, pages 8431–8435, 1998.

[73] D. Kirovski and H.S. Malvar. Spread-spectrum watermarking of audio signals. *IEEE Transactions on Signal Processing*, 51(4):1020–1033, 2003.

[74] Reinhard Klein, Gunther Liebich, and Wolfgang Strasser. Mesh reduction with error control. In *VIS '96: Proceedings of the 7th conference on Visualization '96*, pages 311–318, Los Alamitos, CA, USA, 1996. IEEE Computer Society Press.

[75] J. M. Konstantinides, A. Mademlis, P. Daras, P. A. Mitkas, and M. G. Strintzis. Blind robust 3-d mesh watermarking based on oblate spheroidal harmonics. *Multimedia, IEEE Transactions on*, 11(1):23–38, 2009.

[76] Yehuda Koren, L. Carmel, and D. Harel. Ace: a fast multiscale eigenvectors computation for drawing huge graphs. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 137–144, 2002.

[77] G. Lavoué, F. Denis, F. Dupont, and A. Baskurt. A watermarking framework for subdivision surfaces. In *Multimedia Content Representation, Classification and Security*, pages 223–231, 2006.

[78] Guillaume Lavoué. A local roughness measure for 3d meshes and its application to visual masking. *ACM Trans. Appl. Percept.*, 5(4):1–23, 2009.

[79] Guillaume Lavoué, Florence Denis, and Florent Dupont. Subdivision surface watermarking. *Computers and Graphics*, 31(3):480 – 492, 2007.

[80] Guillaume Lavoué, Elisa Drelie Gelasca, Florent Dupont, Atilla Baskurt, and Touradj Ebrahimi. Perceptually driven 3D distance metrics with application to watermarking. In *SPIE Applications of Digital Image Processing XXIX*, August 2006.

[81] M. Leordeanu and M. Hebert. A spectral technique for correspondence problems using pairwise constraints. In *ICCV '05: Proceedings of the Tenth IEEE International Conference on Computer Vision*, pages 1482–1489, Washington, DC, USA, 2005. IEEE Computer Society.

[82] B. Levy. Laplace-beltrami eigenfunctions: Towards an algorithm that understands geometry. In *IEEE International Conference on Shape Modeling and Applications, invited talk*, 2006.

[83] L. Li, D. Zhang, Z. Pan, J. Shi, K. Zhou, and K. Ye. Watermarking 3d mesh by spherical parameterization. *Computers and Graphics*, 28(6):981 – 989, 2004.

[84] Peter Lindstrom and Greg Turk. Image-driven simplification. *ACM Trans. Graph.*, 19(3):204–241, 2000.

[85] R. Liu and H. Zhang. Mesh segmentation via spectral embedding and contour analysis. *Computer Graphics Forum (Proceeding of EuroGraphics)*, 26(3):385–394, 2007.

[86] Yang Liu, Balakrishnan Prabhakaran, and Xiaohu Guo. A robust spectral approach for blind watermarking of manifold surfaces. In *MM&Sec '08: Proceedings of the 10th ACM workshop on Multimedia and security*, pages 43–52, New York, NY, USA, 2008. ACM.

[87] Michael Lounsbery, Tony D. DeRose, and Joe Warren. Multiresolution analysis for surfaces of arbitrary topological type. *ACM Trans. Graph.*, 16(1):34–73, 1997.

[88] Bin Luo, Richard C. Wilson, and Edwin R. Hancock. A spectral approach to learning structural variations in graphs. *Pattern Recogn.*, 39(6):1188–1198, 2006.

[89] M. Luo, K. Wang, A.G. Bors, and G. Lavoue. Local patch blind spectral watermarking method for 3d graphics. In *Proceeding International Workshop on Digital Watermarking*, Guildford, UK, Aug. 2009.

[90] Ming Luo and A.G. Bors. Principal component analysis of spectral coefficients for mesh watermarking. In *Image Processing, 2008. ICIP 2008. 15th IEEE International Conference on*, pages 441–444, Oct. 2008.

[91] Ming Luo and A.G. Bors. Shape watermarking based on minimizing the quadric error metric. In *Shape Modeling and Applications, 2009. SMI 2009. IEEE International Conference on*, pages 103–110, June 2009.

[92] K. Madsen, H. Nielsen, and O. Tingleff. Methods for non-linear least squares problems. Technical report, Informatics and Mathematical Modelling, Technical University of Denmark, 2004.

[93] M. Meyer, M. Desbrun, P. Schroder, and A.H. Barr. Discrete differential-geometry operators for triangulated 2-manifolds. Heidelberg, 2003. Springer-Verlag.

[94] K. Murotani and K. Sugihara. Watermarking 3d polygonal meshes using the singular spectral analysis. *Joho Shori Gakkai Kenkyu Hokoku*, (86):19–24, 2003.

[95] H. Nielsen. Damping parameter in Marquardt's method. Technical report, Technical University of Denmark, 1999.

[96] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models. In *Proceeding of the ACM International Conference on Multimedia*, pages 261–272, Seattle, USA, November 1997.

[97] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models through geometric and topological modifications. *IEEE Journal Sel. Areas Communication*, 16:551–560, 1998.

[98] R. Ohbuchi, A. Mukaiyama, and S. Takahashi. A frequency-domain approach to watermarking 3-D shapes. In *Proceeding EUROGRAPHICS, Computer Graphics Forum, vol. 21*, pages 373–382, 2002.

[99] R. Ohbuchi, A. Mukaiyama, and S. Takahashi. Watermarking a 3d shape model defined as a point set. In *Proceedings of International Conference on Cyberworlds table of contents*, pages 392–399, Washington, DC, USA, 2004.

[100] R. Ohbuchi, S. Takahashi, T. Miyazawa, and A. Mukaiyama. Watermarking 3D polygonal meshes in the mesh spectral domain. In *Proceeding Graphics Interface*, pages 9–17, 2001.

[101] J.J.K. O'Ruanaidh, W.J. Dowling, and F.M. Boland. Watermarking digital images for copyright protection. In *Proceeding of IEE Visual Image Signal Process*, volume 143, pages 250–256, August 1996.

[102] G. Peyre and L. D. Cohen. Geodesic remeshing using front propagation. *International Journal of Computer Vision*, 69(1):145–156, 2006.

[103] K. Polthier and M. Schmies. *Straightest Geodesics on Polyhedral Surfaces*. pp. 135-150, Mathematical Visualization, Eds: H.C. Hege, K. Polthier, Springer Verlag, 1998.

[104] E. Praun, H. Hoppe, and A. Frinkelstein. Robust mesh watermarking. In *Proceeding SIGGGRAPH*, pages 69–76, 1999.

[105] Huaijun Qiu and Edwin R. Hancock. Graph matching and clustering using spectral partitions. *Pattern Recogn.*, 39(1):22–34, 2006.

[106] Martin Reddy. Perceptually optimized 3d graphics. *IEEE Comput. Graph. Appl.*, 21(5):68–75, 2001.

[107] M. Reuter, F.E. Wolter, and N. Peinecke. Laplace-beltrami spectra as 'shape-dna' of surfaces and solids. *Computer-Aided Design*, 38(4):342 – 366, 2006. Symposium on Solid and Physical Modeling 2005.

[108] R.M. Rustamov. Laplace-beltrami eigenfunctions for deformation invariant shape representation. In *SGP '07: Proceedings of the fifth Eurographics symposium on Geometry processing*, pages 225–233, Aire-la-Ville, Switzerland, Switzerland, 2007. Eurographics Association.

[109] J. A. Sethian. A fast marching level set method for monotonically adavancing fronts. In *Proceeding National Academic Science USA*, volume 93, pages 1591–1595, 1996.

[110] J. A. Sethian. *Level Set Methods and Fast Marching Methods Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science*. Cambridge Press, 1999.

[111] J.B. Shi and J. Malik. Normalized cuts and image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 22(8):888–905, Aug 2000.

[112] J.R. Smith and B.O. Comiskey. Modulation and information hiding in images. In *Lecture Notes in Computer Science*, volume 1174, pages 207–296. Springer-Verlag, 1996.

[113] E.J. Stollnitz, T.D. DeRose, and D.H. Salesin. *Wavelets for Computer Graphics: Theory and Applications*. Morgan Kaufmann, 1996.

[114] H. Sundar, D. Silver, N. Gagvani, and S. Dickinson. Skeleton based shape matching and retrieval. In *SMI '03: Proceedings of the Shape Modeling In-*

ternational 2003, page 130, Washington, DC, USA, 2003. IEEE Computer Society.

[115] V. Surazhsky, T. Surazhsky, D. Kirsanov, S. J. Gortler, and H. Hoppe. Fast exact and approximate geodesics on meshes. In *Proceeding SIGGRAPH, ACM Transactions on Graphics 24(3)*, pages 553–560, 2005.

[116] L. N. T Refethen and D. Bau. *Numerical linear algebra*. SIAM, 1997.

[117] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding secret information into a dithered multilevel image. In *Proceeding of IEEE Military Communications*, pages 216–220, 1990.

[118] G. Taubin. A signal processing approach to fair surface design. In *SIGGRAPH 95: Proceedings of the 22nd annual conference on Computer Graphics and interative techniques*, pages 351–358, New York, NY, USA, 1995. ACM Press.

[119] G. Taubin. Geometric signal processing on polygonal meshes. In *EUROGRAPHICS, State of the art report*, 2000.

[120] N.C. Thong and E.W. Grafarend. A spheroidal harmonic model of the terrestrial gravitational field. *Manuscripta Geodaetica*, 14(5):285–304, 1989.

[121] A. Tremeau and D. Muselet. Recent trends in color image watermarking. *Journal of Imaging Science and Technology*, 53(1), 2009.

[122] Y.Y. Tsai, C.M. Wang, Y.M. Cheng, C.H. Chang, and P.C. Wang. Steganography on 3d models using a spatial subdivision technique. In *Advances in Computer Graphics*, pages 469–476, 2006.

[123] S. Tuzikov, A. Sheynin and P.V. Vasiliev. Computation of volume and surface body moments. *Pattern Recognition*, 36(11):2521–2529, 2003.

[124] F. Uccheddu, M. Corsini, and M. Barni. Wavelet-based blind watermarking of 3d models. In *MM&Sec '04: Proceedings of the 2004 workshop on Multimedia and security*, pages 143–154, New York, NY, USA, 2004. ACM.

[125] Sébastien Valette and Rémy Prost. Wavelet-based multiresolution analysis of irregular surface meshes. *IEEE Transactions on Visualization and Computer Graphics*, 10(2):113–122, 2004.

[126] B. Vallet and B. Levy. Spectral geometry processing with manifold harmonics. *Computer Graphics Forum (Proceeding EuroGraphics*, 27(2):251–260, 2008.

[127] Deepak Verma and Marina Meila. A comparison of spectral clustering algorithms. Technical report, University of Washington, 2003.

[128] Ulrike von Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, December 2007.

[129] D. V. Vranic and D. Saupe. In *Proceedings of the EURASIP Conference on Digital Signal Processing for Multimedia Communications and Services (ECMCS 2001) month = September, pages = 271–274, priority = 2, title = 3D Shape Descriptor Based on 3D Fourier Transform, year = 2001*, Budapest, Hungary.

[130] S. Walton. Image authentication for a slippery new age. *Dr. Dobb's Journal*, pages 18–26, April 1995.

[131] Kai Wang, G. Lavoue, F. Denis, and A. Baskurt. Hierarchical blind watermarking of 3d triangular meshes. In *Multimedia and Expo, 2007 IEEE International Conference on*, pages 1235–1238, July 2007.

[132] Kai Wang, G. Lavoue, F. Denis, and A. Baskurt. Hierarchical watermarking of semiregular meshes based on wavelet transform. *Information Forensics and Security, IEEE Transactions on*, 3(4):620–634, Dec. 2008.

[133] Kai Wang, Ming Luo, Adrian Bors, and Florence Denis. Blind and robust mesh watermarking using manifold harmonics. In *IEEE International Conference on Image Processing*. IEEE, nov 2009.

[134] Yu-Ping Wang and Shi-Min Hu. A new watermarking method for 3d models based on integral invariants. *Visualization and Computer Graphics, IEEE Transactions on*, 15(2):285–294, March-April 2009.

[135] M. Wardetzky, M. Bergou, D. Harmon, D. Zorin, and E. Grinspun. Discrete quadratic curvature energies. *Computer Aided Geometry Design*, 24(8-9):499–518, 2007.

[136] R.C. Wilson and P. Zhu. A study of graph spectra for comparing graphs and trees. *Pattern Recogn.*, 41(9):2833–2841, 2008.

[137] R.B. Wolfgang, C.I. Podilchuk, and E. Delp. Perceptual watermarks for digital images and video. In *Proceedings of the IEEE*, volume 87, pages 1108–1126, 1999.

[138] Hao-Tian Wu and Yiu-Ming Cheung. A fragile watermarking scheme for 3d meshes. In *MM&Sec '05: Proceedings of the 7th workshop on Multimedia and security*, pages 117–124, New York, NY, USA, 2005. ACM.

[139] J. Wu and L. Kobbelt. Efficient spectral watermarking of large meshes with orthogonal basis functions. *The visual computer*, 21(8):848–857, 2005.

[140] B.-L. Yeo and M. M. Yeung. Watermarking 3D objects for verification. *IEEE Computer Graphics and Applications*, 19(1):36–45, 1999.

[141] Zhiqiang Yu, Horace H. S. Ip, and L. F. Kwok. A robust watermarking scheme for 3d triangular mesh models. *Pattern Recognition*, 36(11):2603 – 2614, 2003.

[142] S. Zafeiriou, A. Tefas, and I. Pitas. Blind robust watermarking schemes for copyright protection of 3D mesh objects. *IEEE Transaction on Visualization and Computer Graphics*, 11(5):496–607, 2005.

[143] C. Zhang and T. Chen. Efficient feature extraction for 2d/3d objects in mesh representation. In *Proceeding of the IEEE International Conference on Image Processing*, pages 935–938, 2001.

[144] Fan Zhang and Edwin R. Hancock. Graph spectral image smoothing. In *Graph-Based Representations in Pattern Recognition*, pages 191–203, 2007.

[145] H. Zhang. Discrete combinatorial laplacian operators for digital geometry processing. In *Proceedings of SIAM Conference on Geometric Design and Computing*, pages 575–592, 2004.

[146] H. Zhang, O. Van Kaick, and R. Kyer. spectral methods for mesh processing and analysis. In *EUROGRAPHICS 2009, State of the art report*, pages 1–22, 2009.

[147] J. Zhao and E. Koch. Embedding robust labels into images for copyright protection. In *Proceeding of Interational Congress on Intellectual Property Rights for Specialized Information, Knowledge, and New Technologies*, Vienna, August 1995.

[148] G. Zigelman, R. Kimmel, and N. Kiryati. Texture mapping using surface flattening via multidimensional scaling. *IEEE Transactions on Visualization and Computer Graphics*, 8(2):198–207, 2002.

Appendix A

Comparisons of All Methods

In this appendix, we make the comparison among the methods proposed in this thesis and Cho's method proposed in [31].

A.1 Models and Parameter Settings

We use two models to do the comparison. One is the Bunny object which contains 34,835 vertices and 69,666 faces. All the experiments are taken with 64 bits of message embedded in the object. In the detection stage, we assume that the secret key and the length of the message are known. And the other is Buddha with 89,544 vertices and 179,222 faces. For the spectral method, we have the parameter setting as: $\alpha = 0.1$, $K = 15$ and $T = 2.25$ as used in equations (3.6), (3.20) and (3.23), respectively. And the object is split into $N = 70$ patches grouped into $\kappa = 2$ layers. For the geodesic method, the discretization method and the optimization method, the watermarking strength factor α are all set as $\alpha = 0.1$. For Cho's method, we set $\alpha = 0.05$ so that the watermarked object can have similar surface quality with those watermarked by geodesic, QSP and L-M Methods.

A.2 Distortion

We use the MRMS as introduced in Section 2.4.1 to evaluate the distortion introduced by the watermarking embedding as shown in Table A.1. The value obtained in the table are may vary from time to time if the watermark message is changed.

We use the same message to watermark the object using different algorithms to test the numerical distortion. The difference between the algorithms are small but L-M method provides the smallest distortion.

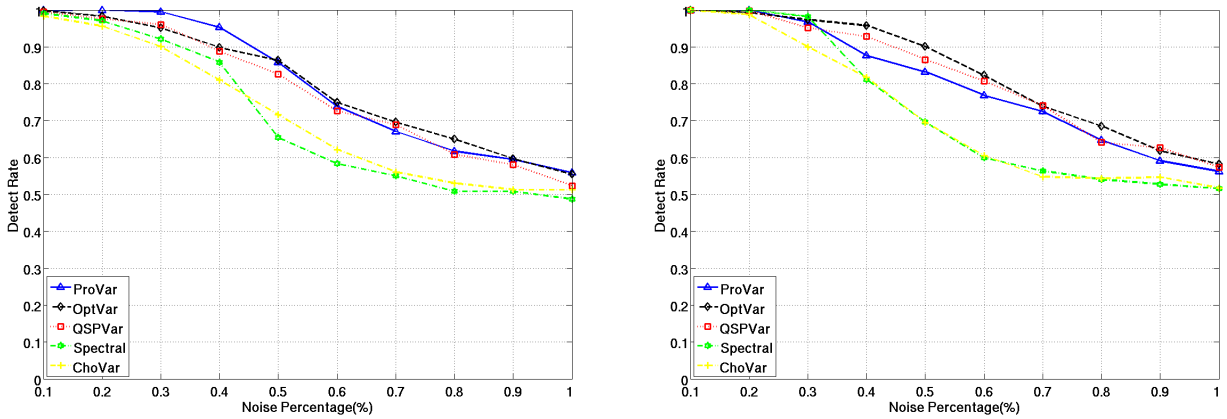
Note that the surface distortion of each method is affected by two reasons. Firstly, the embedding message. For example a message 0101 has different effect as 1111 on the surface. Secondly, different initialization gives different result. For example, in the method of geodesic distance, a small region at the end point (the smallest geodesic distance point and the largest) are not considered for watermarking. This is determined by $\varepsilon \in (0, 0.2)$. For example, if $\varepsilon = 0.05$, there are more vertices can be used for embedding and in this case the distortion is smaller. If $\varepsilon = 0.15$, then 10% of vertices can not be used and the difference is quite large. The epsilon is generated according to a secret key. Therefore, different key also gives different results. The results obtained in Table A.1 are seeded using the same key 35233.

	Bunny	Buddha
Spectral	0.25	0.26
ProVar	0.25	0.24
QSPVar	0.26	0.19
L-MVar	0.21	0.19
ChoVar	0.27	0.26

Table A.1: Watermarked object distortion with respect to the original object measured by MRMS, where all results should be multiplied with 10^{-4}.

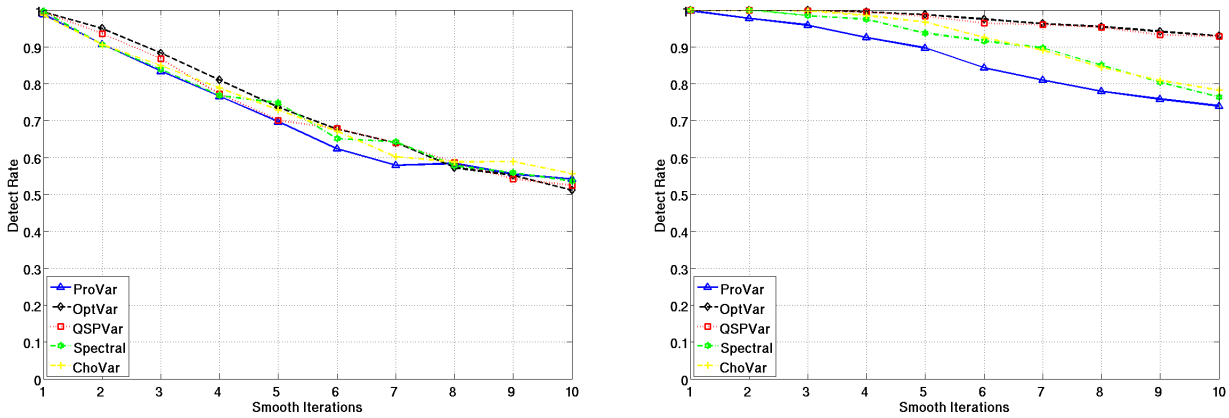
A.3 Robustness

The noise attack is defined in equation (3.24) in section 3.6.4. Figure A.1, figure A.2, figure A.3, figure A.4 and figure A.5 demonstrate the robustness comparison of five methods against the attack of additive noise, Laplacian Smoothing, mesh simplification, quantization and uniform resampling respectively. All the results are taken as an average value of 20 random cases. As shown in those results, L-M and QSP methods possess the highest overall robustness. These results are obtained user the same key 35233.



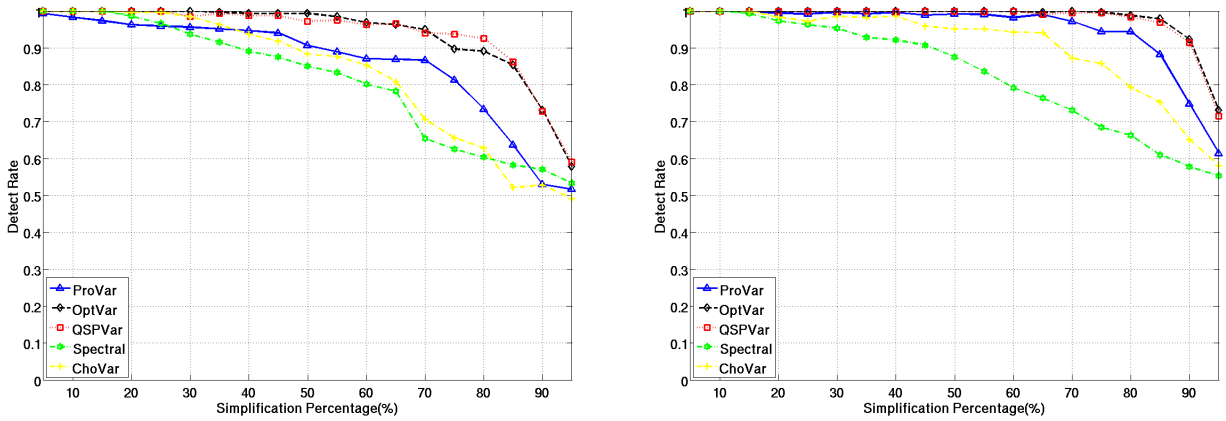
(a) (b)

Figure A.1: Plots showing the robustness at noise, (a) Bunny, (b) Budda



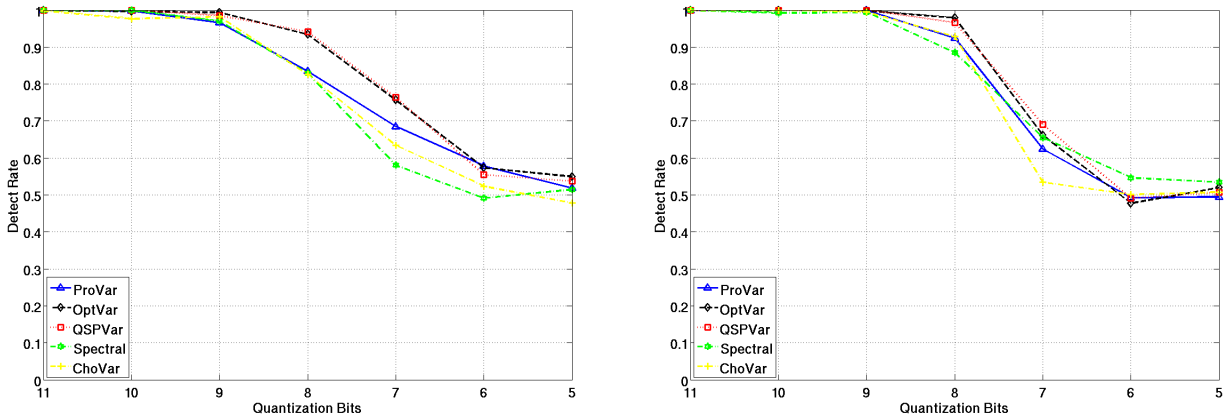
(a) (b)

Figure A.2: Plots showing the robustness at smooth, (a) Bunny, (b) Budda. $\lambda = 0.3$



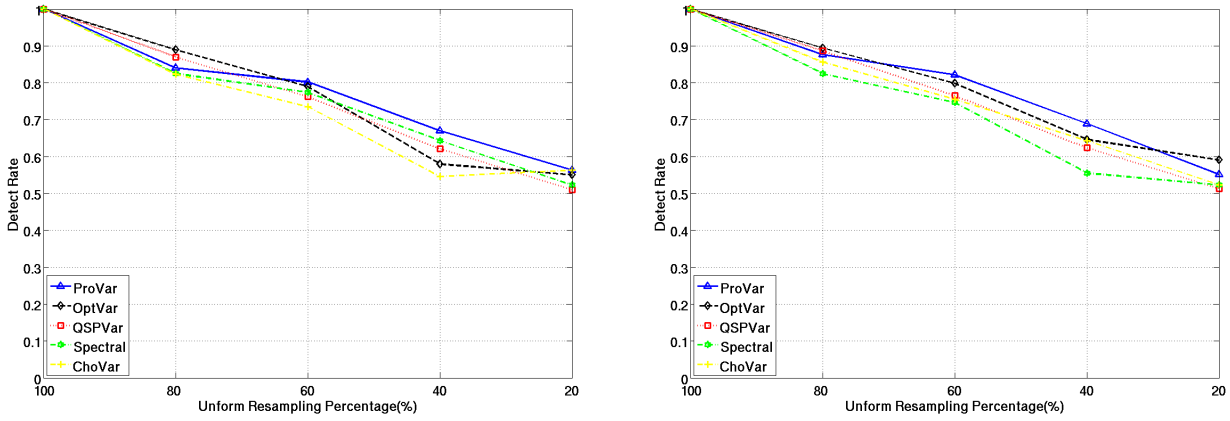
(a) (b)

Figure A.3: Plots showing the robustness at mesh simplification attack, (a) Bunny, (b) Budda.



(a) (b)

Figure A.4: Plots showing the robustness at quantization, (a) Bunny, (b) Budda.



(a) (b)

Figure A.5: Plots showing the robustness at resampling, (a) Bunny, (b) Budda.