Topological Approaches to 3D Mesh Watermarking

Abdullah Omer A. Abbas

A Thesis

in

The Concordia Institute

for

Information Systems Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of Master of Applied Science (Information Systems Security) at

Concordia University

Montréal, Québec, Canada

March 2008

Your file Votre référence
ISBN: 978-0-494-40902-2
Our file Notre référence
ISBN: 978-0-494-40902-2

Abstract

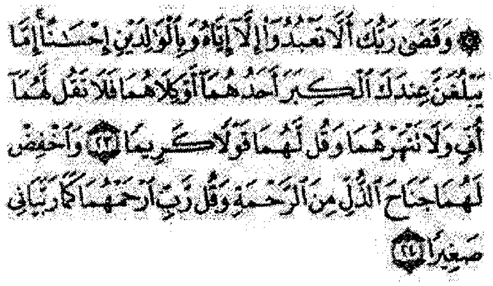
Topological Approaches to 3D Mesh Watermarking

Abdullah Omer A. Abbas

The recent rapid growth of digital media contents and the increase use of online services have triggered the need for multimedia protection. Watermarking plays an important role to solve the problem of unauthorized replication. Watermarking can be defined as the process of embedding data called "watermark" into a digital object without making changes to the quality of the host substantially. The digital object could be an image, video, or audio. The watermark is used as a signature to prove ownership and can only be detected or extracted by the owner.

This thesis is devoted to three robust watermarking techniques that we have developed for copyright protection. The first watermarking technique inserts a binary number into a set of critical points of a digital elevation map (DEM). Our method starts by extracting the critical points from a DEM depending on the important topographic features of the terrain. Then we embed the fingerprints into the coordinate values of all the critical points. The second watermarking technique partitions a 3D model into sub-meshes, then apply the eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh, followed by computing the hash value of each sub-mesh. The hash value is defined in terms of the entropy of each submesh. The last watermarking technique applies content-based hashing algorithm to the DEM using higher-order statistics and discrete wavelet transform to define the image fingerprint.

Acknowledgements



۞ وَقَضَىٰ رَبُّكَ أَلَّا تَعْبُدُوٓا إِلَّآ إِيَّاهُ وَبِٱلْوَٰلِدَيْنِ إِحْسَٰنًا ۚ إِمَّا
يَبْلُغَنَّ عِندَكَ ٱلْكِبَرَ أَحَدُهُمَآ أَوْ كِلَاهُمَا فَلَا تَقُل لَّهُمَآ
أُفٍّ وَلَا تَنْهَرْهُمَا وَقُل لَّهُمَا قَوْلًا كَرِيمًا ﴿٢٣﴾ وَٱخْفِضْ
لَهُمَا جَنَاحَ ٱلذُّلِّ مِنَ ٱلرَّحْمَةِ وَقُل رَّبِّ ٱرْحَمْهُمَا كَمَا رَبَّيَانِى
صَغِيرًا ﴿٢٤﴾

Figure 1: SURAH AL-ISRA (Aya 23 and 24)

23. Thy Lord hath decreed, that ye worship none save Him, and (that ye show) kindness to parents. If one of them or both of them attain old age with thee, say not "Fie" unto them nor repulse them, but speak unto them a gracious word. 24. And lower unto them the wing of submission through mercy, and say: My Lord! Have mercy on them both as they did care for me when I was little.

Table of Contents

List of Tables

List of Figures

CHAPTER 1

Introduction

1.1 Framework and motivation

Digital watermarking can be defined as the method of embedding data called a watermark into a digital content without affecting the quality of the digital content. The digital content could be an image, video, or an audio. Watermarking is used as a method to prove the ownership of the digital content. Therefore, only the owner of digital content can detect or extract the watermark using special key. Watermarking method should satisfy the watermarking requirement as an example imperceptibility, capacity, perceptibility, detectability and robustness against attacks. [1–3].

1.2 Watermarking requirements

Watermarking requirements are important in the creation and evaluation of a watermarking system. These requirements are linked to the resistance against attacks, common distortions and detectability [1, 2]. In the following section, we will explain some of the characteristics of these requirements.

Imperceptibility or invisibility

This property describes the ability of a third party to visually detect the existence of a watermark in the digital content. We consider the watermarking embedding system as imperceptibility if an innocent third party looking at the digital content is unaware of the watermark existence. To do that, the embedding system should not degrade the quality of the digital content.

Robustness

This property describes the surviving ability for the watermark throw the embedding and extraction algorithm even after intentional or unintentional attacks.

Unambiguousness

The extraction of the watermark should easily identify the owner of the digital content.

Capacity

This property refers to the amount of the data payload. This can be measured by counting the bits of the data payload embedded in each byte of the digital content.

Detectability

Detectability refers to the probability of an attacker to determine the existence of a watermark in a digital content. If the attacker can find the watermark in digital content with high probability we consider the water marking system as a low security system.

1.3 General watermarking scheme

All Digital watermarking systems consist of two parts, the embedding system and the watermark extraction or recovery system as illustrated in Figure 1.3.

Watermarking embedding system

To run watermarking embedding system we need to feed it with the cover media, the key and the watermark symbol. The output of the embedding system is the watermarked data. Figure 1.1 shows the model for watermark embedding system.

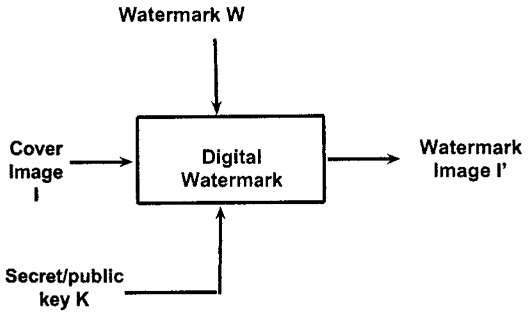


Figure 1.1: Watermark embedding model.

Watermarking extraction system

To run the watermark extraction system we need to input the watermarked data, the secret key, the original cover and/or the original watermark. The end result of running the system will be the suspect watermark or some kind of confidence measure. Figure 1.2 shows a watermark extraction system.

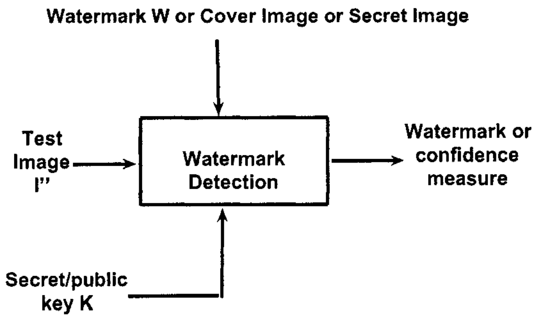

Watermark W or Cover Image or Secret Image
Test Image I"
Watermark Detection
Watermark or confidence measure
Secret/public key K

Figure 1.2: Watermark extraction model.

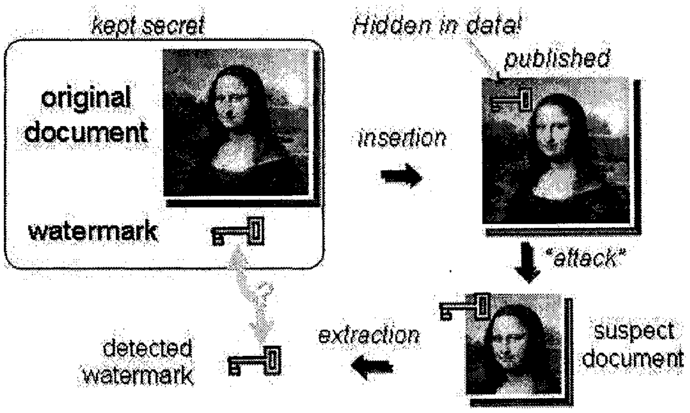


Figure 1.3: Watermark embedding and extraction model.

1.4 Types of watermarking systems

Watermarking system can be classified to different categories according to different criterion. [3,4].

Visible watermark

In the visible watermark, the water is embedded into the digital content in such a way that the watermark is purposely perceptible to the human observer [5]. This will help prevent unauthorized use of the digital content and encourage the observer to know the owner of the material. This watermark is commonly used in logos.

Invisible watermark

On the other hand, the invisible watermark is used to make an assertion about the digital content ownership [6,7]. To do that, the watermark is embedded into digital content where the watermark will not be perceptible to the observer. To extract the watermark from the digital content we need to use a computer program.

Private watermarking system

Private watermarking system also known as non-blind or non-oblivious watermarking system [8,9]. In the detection part of the system we need to provide the system with the digital content (host), secret key, and the watermarked digital content. This will allow only the authorized users to access the watermark.

Public watermarking system

It also called blind or oblivious watermarking system. In the detection part of the system we need to provide the system with only the secret key. The secret key used to generate a pseudo-random sequence watermark using the key as the seed.

Semi-private watermarking system

It also called semi-blind or semi-oblivious watermarking system. This system does not require the original cover for detection. Instead it requires a copy of the embedded watermark data and the secret keys.

Watermarking in spacial domain

It is among the earliest and simplest watermarking systems. In this system the watermark will be encoded by modifying pixels directly [4]. One of the advantage of this system is its lower computation cost. On the other hand, the system has low information hiding capacity; also the watermark can be easily erased using image compression.

Watermarking in transform domain

This system provides more capacity and better robustness against watermarking attacks. In this system the watermark information inserted into transforms coefficients of the digital content. Most of the presented idea has been originated from Cox et al [8]. Barni et al [10] improve the watermarking in transform domain using a blind detection system.

1.5 Application of watermarking

The application of watermark can be used in different areas [1–3] and can be categorized as follows:

Watermarking for copy protection

One of the important watermark applications is copy protection. This application is used to prevent unauthorized copying of the digital content. The water mark embedded into the digital content work as a copy-permission bit stream. This watermark will let the copying device know if the copying procedure legal or not. Watermarking for copy protection achieves bad result in open systems; in closed or proprietary systems [11].

Fingerprinting for transaction tracing

Fingerprinting application is used to convey information about the legal recipient of the digital content. This application embeds different watermarks for each copy to trace illegal copies of the digital content. Fingerprinting for transaction tracing requires a high robustness against standard data processing attacks.

Watermarking for copyright protection

Copyright protection is the most important application of watermarking used today. The goal is to protect the copyright of the digital content by embedding information about the owner into the digital content. Then, the watermark can be used as a proof of ownership [12].

Watermarking for image authentication

In this application we use the watermark to check the authenticity of the digital content. The watermark in this case is fragile and very sensitive to any kind of modifications. If after the extraction the watermark was corrupted, this shows that digital content has been altered.

Medical safety

The purpose of this application is to increase the confidentiality and security of medical information by embedding the date and patient's information into the medical image.

Broadcast monitoring

This application work by embedding the watermark into the digital content of the commercial advertisements. This will allow an automated monitoring system to verify if the advertisements broadcasted as contracted.

1.6 Overview of digital image watermarking techniques

Because of the large demands for image watermarking products most of the research and publications are focused on 2D images [13–16]. The idea for digital image watermarking is to embed the watermark into the digital content and use it later as a proof of ownership.

There are different watermark techniques have been proposed depending on the embedding domain. The watermark embedding procedural can be directly in the spatial domain or in some transform space using common transforms, such as Discrete Fourier Transform (DFT) [5, 17], Discrete Cosine Transform (DCT) [5, 13, 18], Discrete Wavelet Transform (DWT) [14, 15, 19–21], and Fast Hadamard Transform (FHT) [22–24]. In the case of transform-based schemes the image is transformed before the watermark embedding then the watermark is hidden in the transformed coefficients representing the image. To extract the watermark image we used an inverse transformation.

There are a greatly use of transform domains in the research of image compression and the result can be applied to digital watermarking. As an example mapping a typical image into the frequency domain concentrate the energy in low-index terms which are very large comparing to the high-index terms.

Transform domains have been extensively studied in image compression and many research results can be applied to digital watermarking. For example, when a typical image is mapped into the frequency domain, the energy is concentrated in low-index terms which are very large comparing to the high-index terms. Demonstrated a digital image in low frequency components represent the over all shape of the image, outline of features in the image, the luminance and the contrast characteristics. High frequencies components represent sharp edges. As an example 95% of the energy found in the lowest 5% of the special frequencies of the two dimensional DCT domain.

Because many of the common signal processing and geometric processes affect commonly insignificant components, the embedding of the watermark should not be located perceptually in insignificant regions of the image or its spectrum. As an example, the operation of lossy compression eliminates insignificant components of an image. To preserve the watermark algorithm from this operation the watermark should be placed in a significant region of the cover. Since the loss happen in the high frequency components the best solution is to place the watermark in low frequency components.

1.7 Spread spectrum watermarking

Cox et al. [8,25] propose a new invisible robust watermarking technique by inserting the watermark into the spectral components of the image using DCT domain. The idea of spread spectrum watermarking system is to spread a watermark in a narrow-band signal in wider and important frequency bands. These bands are obtained from the transformed cover image. As a result the watermark for each band will be smaller and undetectable. In the extraction of the watermark the knowledge of spreading function will be used to extract and sum up the watermark.

In [8] the watermark is embedded in the first n lowest frequency components or the first highest magnitude components $V = \{v_i\}_1^n$ of the full image DCT in order to provide high level of robustness to JPEG compression. The watermark consists of a sequence of real numbers $W = \{w_i\}_1^n$ is computed where each w_i is chosen according to $N(0,1)$ where $N(0,1)$ denotes a normal distribution with mean 0 and variance 1. It is embedded into an image using formula $\grave{v}_i = v_i(1 + \alpha w_i)$where α is the watermark strength factor=0.1. Watermark detection is performed using the following similarity measure:

$$sim(W, \acute{W}) = \frac{W, \acute{W}}{\sqrt{\acute{W}, \acute{W}}}$$

The W' is the extracted watermark, which calculated as:

$$\{\acute{w}_i\}_1^n = \{(\frac{\acute{v}_i}{v_i} - 1)/\alpha\}_1^n$$

Where $\acute{v}_i$ components are extracted from the received watermarked image, and v_i component extracted from the original cover image. The watermark is present if the extracted $sim(W, \acute{W})$ is greater than threshold.

Cox et al. spread the watermark across 1000 lowest frequency. Robustness tests showed that the watermark is robust to common attacks. Retrieval of the watermark unambiguously identifies the owner and the watermark. The watermarking technique has the disadvantage that it needs the original image for its extraction. It is also not clear whether the watermark is robust to photocopying. Figure 1.4 give the process of the insertion and extraction process.

1.8 Singular value decomposition techniques

The Singular Value Decomposition (SVD) is a widely used technique to decompose a matrix into several component matrices, exposing many of the useful and interesting properties of the original matrix. (SVD)is developed for a variety of applications. The main properties of (SVD) from the viewpoint of image processing applications are: The singular values (SVs) of an image have very good stability, i.e., when a small perturbation is added to an image its (SVs) do not change significantly; (SVs) represent intrinsic algebraic image properties.

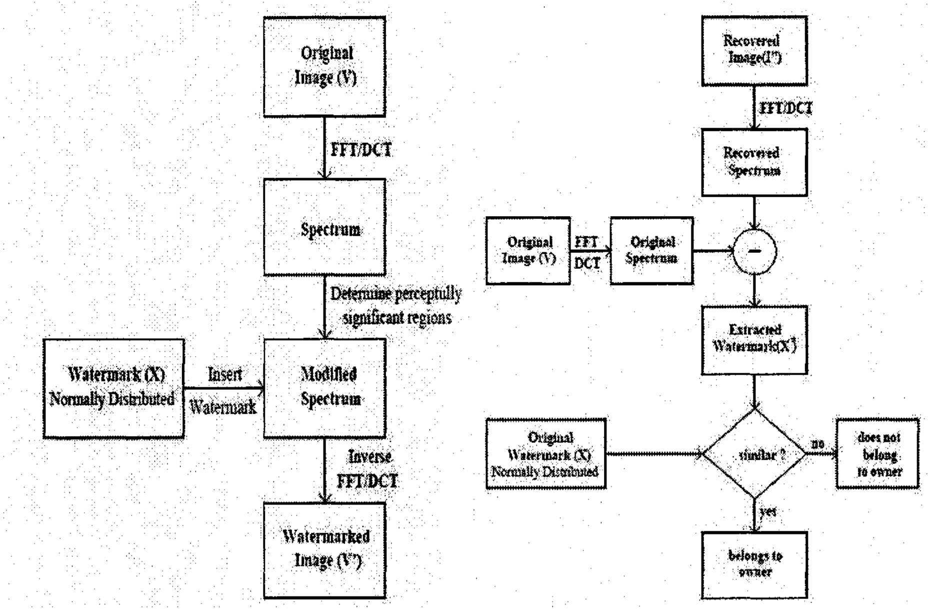


Figure 1.4: Spread spectrum watermarking embedding and extraction model.

The interest in the (SVD), from the point of view of watermarking is its ability to decompose the cover and the watermark images in to three matrices. Let A be an arbitrary real $m \times n$ matrix. There are two orthogonal matrices U and V, $U^TU = I$, $V^TV = I$ and a diagonal matrix Σ; such that:

$$A = U\Sigma V^T$$

In this case, U is $m \times m$ and V is $n \times n$, so that Σ is rectangular with the same dimensions as A. The diagonal entries of Σ, called the singular values (SVs) of A. The columns of U and V are called left and right singular vectors for A. Each SV specifies the luminance of the image layer and the corresponding pair of singular vectors specify the geometry of the image.

New invertible digital image watermarking method based on singular value decomposition was proposed [14]. This method performs well both in resolving rightful ownership and in resisting common attacks. The watermarking embedding and extraction algorithms can be summarized as follows: in watermark embedding process, the singular value decomposition of an $N \times N$ cover image A is computed to obtain two orthogonal matrices U and V and one diagonal matrix S, other non-square images can be processed in exactly the same way.

$$A \Rightarrow USV^T$$

The watermark W is added to the matrix S, followed by singular value decomposition to the new matrix

$$S + \alpha W \Rightarrow U_w S_w V_w^T$$

Where the positive constant α is the scale factor which controls the strength of the watermark to be inserted. The watermarked image A_w is obtained by:

$$A_w \Rightarrow U S_w V^T$$

In watermark detection algorithm, they simply reverse the above steps given U_w, S, and V_w matrices which are saved in the secret key during embedding process of the watermarked image and possibly distorted image $A_w^\star$.

$$A_w^\star \Rightarrow U^\star S_w^\star V^{\star T}$$

$$D^\star \Rightarrow U_w S_w^\star V_w^T$$

$$W^\star \Rightarrow 1/\alpha(D^\star - S)$$

To study the robustness of (SVD) watermarking method they compared the results with the Spread Spectrum Communication method proposed by Cox [1]. The results show that the (SVD) [26] method is much more robust by testing it against six different attacks: adding noise, low pass filtering, JPEG compression, scaling, image cropping and rotation.

1.9 Wavelet techniques

With the standardization of JPEG-2000 and the decision to use wavelet-based image compression instead of DCT-based compression. In several recent publications, wavelet technique has been applied to image watermarking. It prevents watermark removal by JPEG-2000 lossy compression. One dimensional DWT converts an input sequence into a low pass sub-band and high pass sub-band. A two dimensional DWT is constructed from single level decomposition first to the columns and then to the rows to give four sub-bands as shown in Figure 1.5 and Figure 1.6. In the first level decomposition, the lowest frequency band is found in the top-left corner LL. At the same resolution level, the block HL contains information about the highest horizontal and lowest vertical frequency band. Similarly, the block LH contains information about the lowest horizontal and the highest vertical frequency band, and block HH contains information about the highest horizontal and the highest vertical frequency band. The same process is repeated for higher levels.

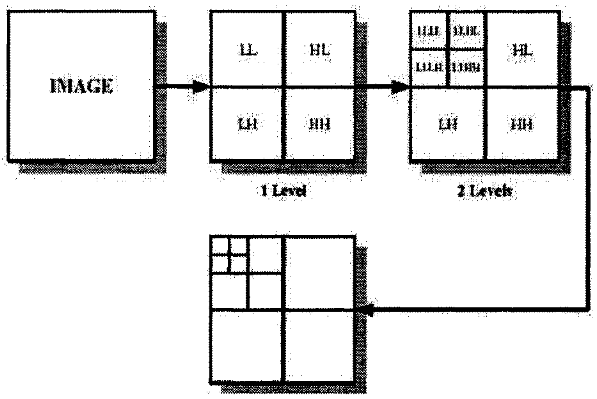


Figure 1.5: 3-Level of 2D Discrete Wavelet Decomposition model.

Recently many watermarking techniques use wavelet transform in watermarking. Some of the

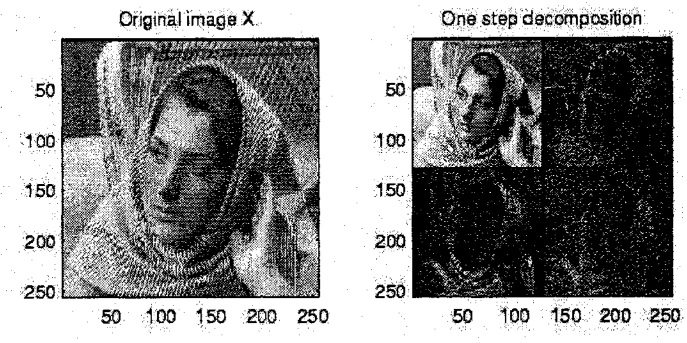


Figure 1.6: One-level of 2D Discrete Wavelet Decomposition for an image.

schemes that were reviewed will be discussed briefly. In [20], the authors used the idea that embedding the watermark in the low frequency area increase the robustness with respect with image distortion that have low pass characteristics like filtering, lossy compression, geometrical distortions. On the other hand, oblivious schemes with low-frequency watermarks are more sensitive to modifications of the histogram, such as contrast, brightness adjustment, gamma correction, histogram equalization, and cropping. Watermarks inserted into middle and high frequencies are typically less robust to low-pass filtering lossy compression and small geometric deformations of the image, but are extremely robust with respect to noise adding. It is understandable that the advantages and disadvantages of low and middle-to-high frequency watermarks are complementary. It appears that by embedding two watermarks into one image could achieve extremely high robustness properties with respect to image processing operations. The above reasoning leads to propose many techniques to embedding multiple watermarks into the low frequency and high frequency bands of Discrete Wavelet Transform.

In [20], two level decomposition is applied to the cover image, followed by embedding the watermarks into the second level LL and HH band respectively. The watermarked image is obtained using the following relationship. $\hat{V}_{ij} = V_{ij} + \beta W(i,j)$, where $\hat{V}_{ij} = (i,j)th$ watermark embedded DWT coefficient, $V_{ij} = (i,j)th$ DWT coefficient of value V, and β is a scaling factor which determines the strength of the watermark.

For the watermark extraction algorithm, two level of DWT decomposition is applied to the suspected and the original watermarked images to recover the LL and HH bands. Subtraction of the suspected and original bands is performed to recover the watermark bits in both LL and HH bands. The output is then divided by the watermark strength factor β. The operation can be summarized as $\hat{W}(i,j) = (\hat{V}(i,j) - V(i,j))/\beta$.

Another watermarking scheme proposed in [15] used SVD and DWT to embed the SVs of the watermark image in all frequencies of the discrete wavelet transformed cover image. This method consists of decomposing the cover image into four transformed sub-bands (LL, LH, HL, and HH), then the SVD is applied to each band, followed by modifying the singular values of the transformed sub-bands with the singular values of the visual watermark. This modification in all frequencies provides more robustness to different attacks. It is important to note that the wavelet coefficients with the highest magnitude are found in the LL sub-band, and those with the lowest coefficients are found in the HH sub-band. Correspondingly, the singular values with the highest magnitudes are in the LL sub-band, and the singular values with the lowest magnitudes are in the HH sub-band, Therefore, two scaling factor are used. The first scaling factor is used for the LL sub-band and the second scaling factor is used for all other sub-bands such that, the first scaling factor is greater than the second one. Experimental results show that the watermarks inserted in the lowest frequencies (LL sub-band) are resistant to one group of attacks, and the watermarks embedded in highest frequencies (HH sub-band) are resistant to another group of attacks. If the same watermark is embedded in 4 blocks, it would be extremely difficult to remove or destroy the watermark from all frequencies. In some cases, embedding in the HL and LH sub-bands is also resistant to certain attacks. Two examples of those attacks are histogram equalization and gamma correction.

1.10 Thesis overview and contributions

The organization of this thesis is as follows:

- ❏ In the first Chapter, we reviewed the essential concepts and definitions which will be used throughout the thesis. Also, we present a short summary of material relevant to watermarking systems, digital image watermarking and embedding and extraction models.
- ❏ In Chapter 2, we introduce a robust DEM watermarking scheme by inserting a binary number into set of critical points of a DEM. Our method starts by extracting the critical points from a DEM depending on the important topographic features of the terrain. Then we embed the fingerprints into x, y and z values for all the critical points. In the experimental results, we test the robustness of the proposed method against a number of challenging attacks.
- ❏ In Chapter 3, we present a new robust hashing technique for 3D models. The main idea is to partition a 3D model into sub-meshes, then apply the eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh, followed by computing the hash value of each sub-mesh. The hash value is defined in terms of the entropy value of each sub-mesh. The experimental results on a variety of 3D models demonstrate the effectiveness of the proposed technique in terms of robustness against the most common attacks including Gaussian noise, mesh smoothing, mesh compression, scaling, rotation as well as combinations of these attacks.
- ❏ In Chapter 4, we propose a digital elevation map (DEM) fingerprint scheme. The key idea

is to apply content-based hashing algorithm to the DEM using higher-order statistics and discrete wavelet transform to define the image fingerprint. We conducted several experiments to compare the performance of our proposed method with existing techniques. Experimental results show the great performance of the proposed method in terms of robustness against a number of challenging attacks.

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

CHAPTER 2

Watermarking of digital elevation maps

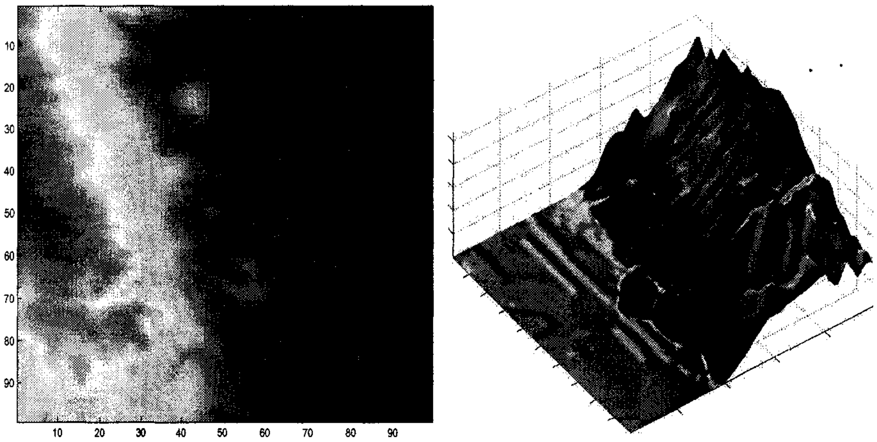
In this chapter, we propose a robust digital elevation map (DEM) watermarking scheme by inserting a binary number into a set of its critical points. Our method starts by extracting the critical pointes from a DEM depends on the important topographic features of the terrain. Then, we embed the fingerprints into x, y and z values for all the critical points. In the experimental results, we test the robustness of the proposed method against a number of challenging attacks.

2.1 Introduction

The recent rapid growth of digital media contents and the increase use of online services have triggered the need for multimedia protection. Watermarking plays an important role to solve the problem of unauthorized replication [27]. Watermarking can be defined as the process of embedding data called "watermark" into a digital object without making changes to the quality of the host substantially [1,28]. The digital object could be an image, video, or audio. The watermark is used as a signature to prove ownership and can only be detected or extracted by the owner.

A variety of watermarking techniques have been proposed. These techniques can be divided into two types according to the embedding domain of the cover image: spatial domain methods and transform domain methods [4, 8]. The spatial domain method directly modifies the intensities of selected pixels. Whereas, the transform domain method modifies the values of selected transformed coefficients. Based on a variety of studies, it is proved that the frequency domain method is superior to the spatial domain method in terms of robustness against watermarking attacks.

A DEM is a raster of elevation values, and consists of an array of points of elevations, sampled systematically at equally spaced intervals. We may represent a DEM as an image $I : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$ (see Fig 4.1(a)) where each image location denotes a height value. DEMs are usually constructed from aerial photographs and require at least two images of a scene [29].



(a) (b)

Figure 2.1: Illustration of a DEM in 2D and 3D

DEMs are used to provide a digital representation of surface terrains in three-dimensional space. Because of the large amount of effort in acquiring DEMs they have high commercial value. Also, DEMs can carry critical geospatial information when they are used in sensitive military applications. For these reasons we used watermarking to prevent unauthorized distribution and to trace back illegally produced copies.

Before starting distributing copies of a DEM, the authority has to embed into each copy of a DEM a unique ID. This ID known as a watermark and represents a recipient's identity. If some recipients leak their copies and the leaked copies acquired by the authority, we can find the source of the leak by examining the IDs in the suspicious DEMs. To succeed tracing the individual copies, it must be difficult to remove the embedded watermarks under variety of attacks.

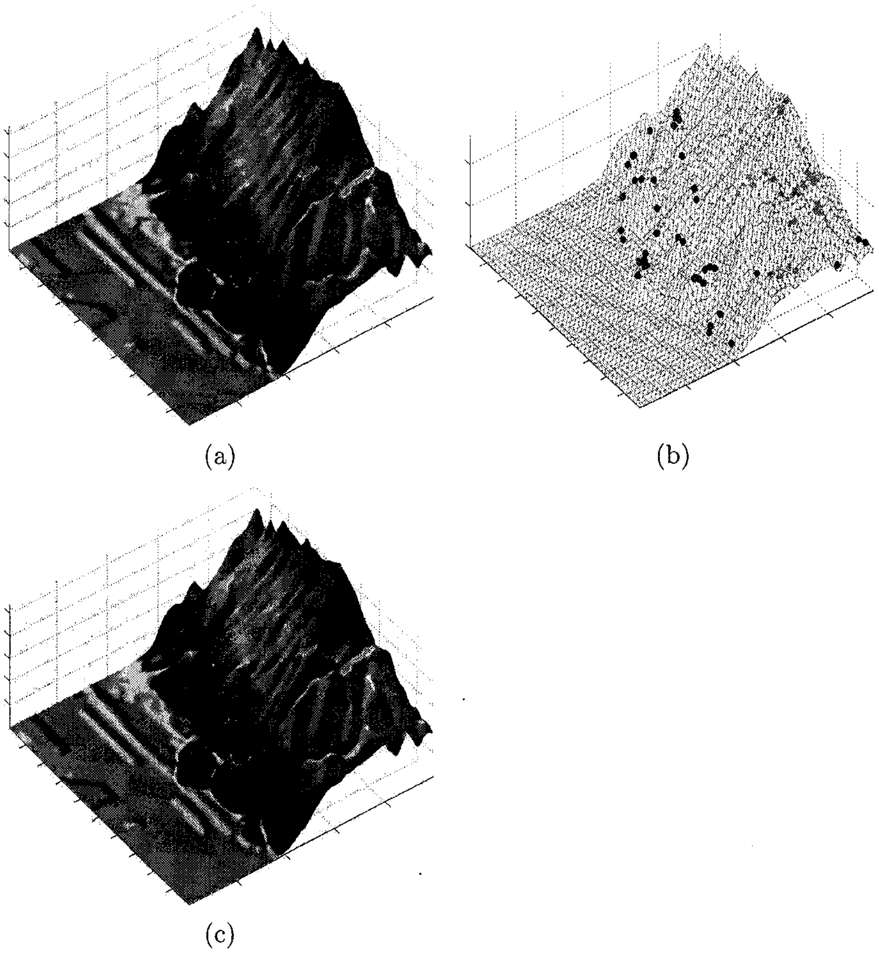
Recently, a digital watermark technique to protect DEM data from illegal re-distribution has been proposed [30]. The main idea of this technique is to extract a set of critical contours from a DEM depending on the important topographic features of the terrain. A watermark is then embedded into these critical contours by using parametric curve modeling and spread spectrum embedding. Finally, watermarked DEM is constructed to include the marked 2-D contours.

In this chapter we introduce a robust method for digital watermarking and secure copyright protection of DEMs. The proposed watermarking method will embed a different binary number into the critical points for each distributed copy of the DEM.

This chapter is organized as follows. Section 2.2 we introduce the proposed watermark embedding and extraction algorithms. In section 2.3, we provide experimental result to demonstrate the improve performance of the proposed method. Finally we conclude in section 2.4.

2.2 Proposed Watermarkink Method

In this section, we will explain the main steps of the proposed watermark embedding and extraction algorithms which are also illustrated in the block diagrams shown in Fig. 2.3 and in Fig. 2.5. The goal of our proposed approach is to embed the watermark binary number in all the critical points to provide a better robustness against attacks.



(a) (b) (c)

Figure 2.2: (a) Original DEM, (b) critical points, (c) watermark DEM

2.2.1 Watermarking Embedding Algorithm

To watermark a DEM through hiding data in its critical points we do the following steps:

1) The first step is to extract the critical points that will be used to hide the watermark. These critical points will represent the terrain features such as peaks, saddles and pit [31].

2) After we identify the critical points to carry out the hiding data, we watermark a DEM by

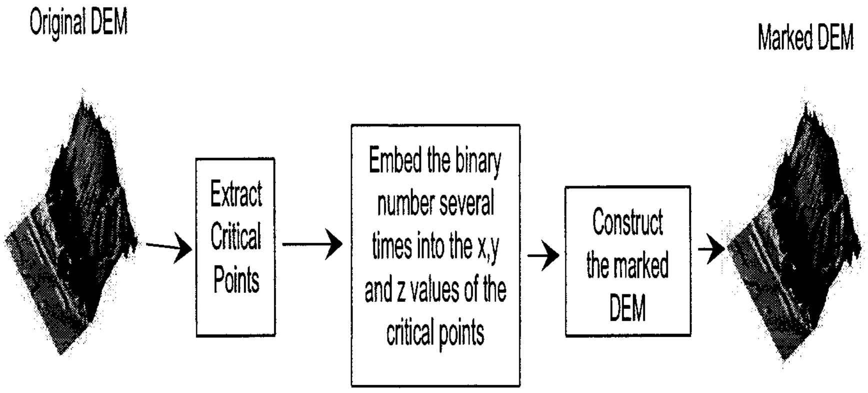

Original DEM
Extract Critical Points
Embed the binary number several times into the x,y and z values of the critical points
Construct the marked DEM
Marked DEM

Figure 2.3: Embedding procedure.

inserting a binary number into the x,y and z values for all the critical points (peaks, saddles and pits) by using the following formula:

$$\lambda_w^i = \lambda_o^i + \alpha\lambda_{wi}$$

Where λ_o^i denotes the transformed critical points, λ_{wi} denotes the watermark binary number, and α is a constant strength factor.

3) Construct the marked DEM.

2.2.2 Watermarking Extraction Algorithm

To extract the binary number, we do the following steps:

1) We use the following formula to get the binary number we inserted in x,y and z values of each

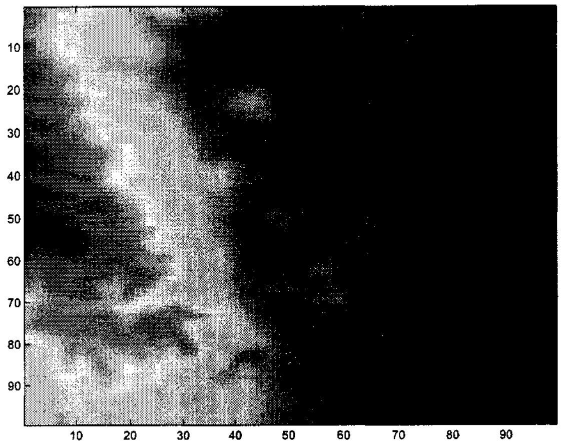


Figure 2.4: The 2D view of a DEM after inserting the binary number.

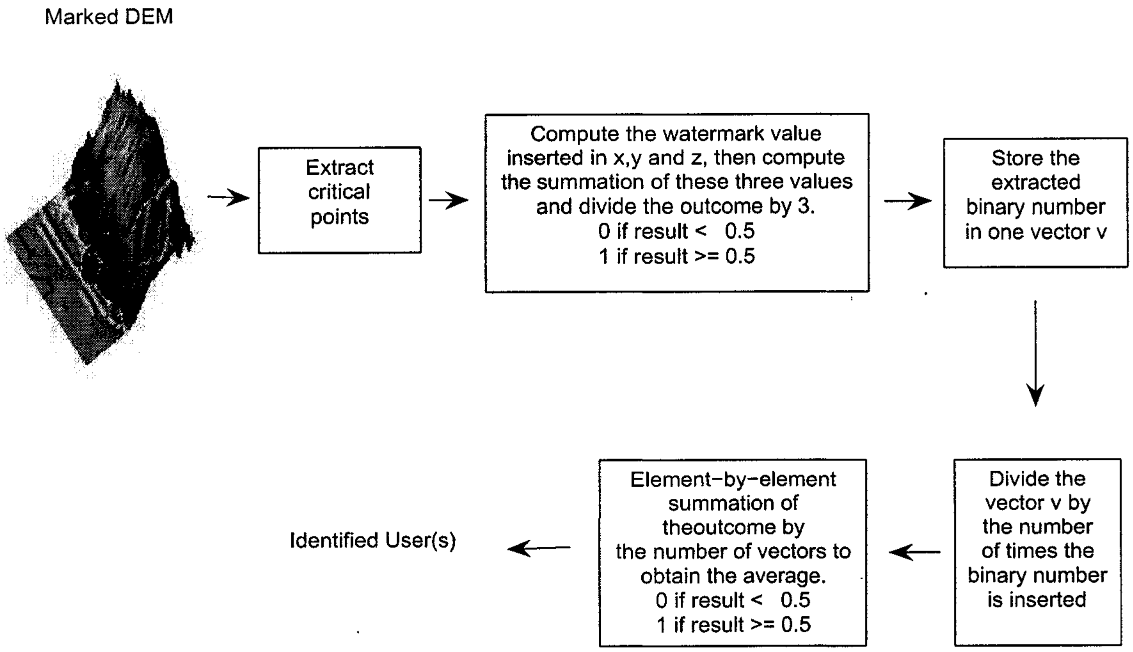
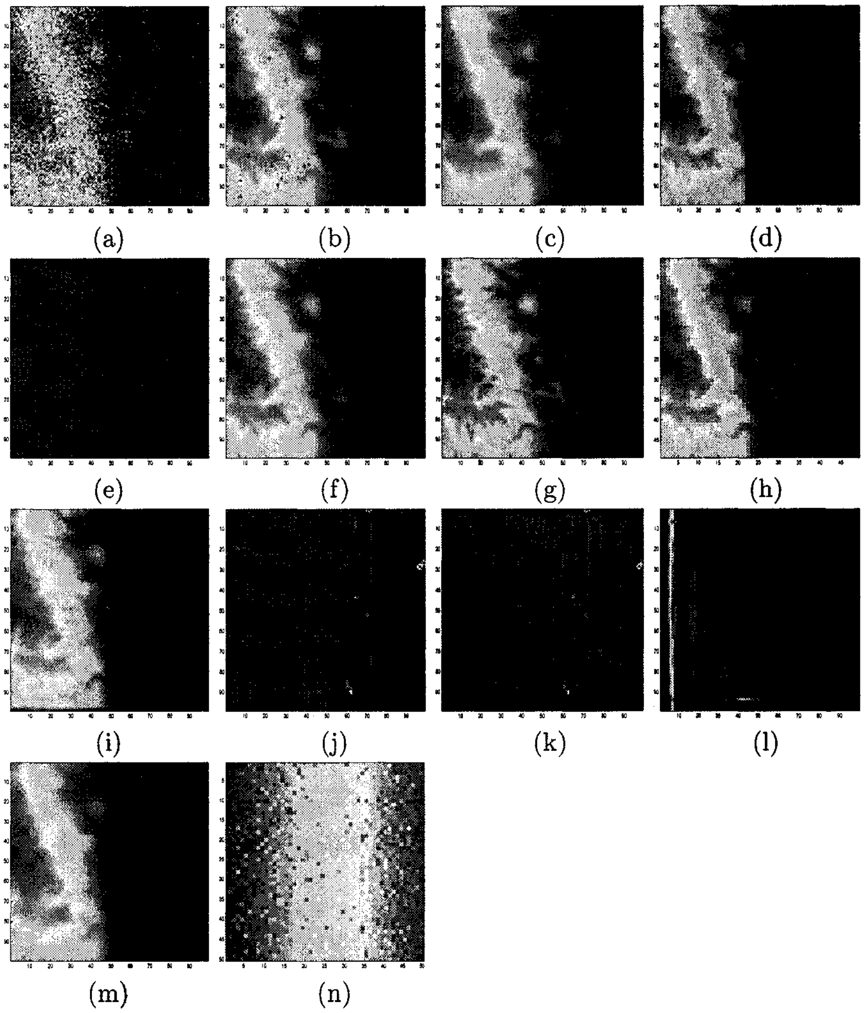

Marked DEM
Extract critical points
Compute the watermark value inserted in x,y and z, then compute the summation of these three values and divide the outcome by 3.
0 if result < 0.5
1 if result >= 0.5
Store the extracted binary number in one vector v
Divide the vector v by the number of times the binary number is inserted
Element-by-element summation of theoutcome by the number of vectors to obtain the average.
0 if result < 0.5
1 if result >= 0.5
Identified User(s)

Figure 2.5: Extraction procedure.



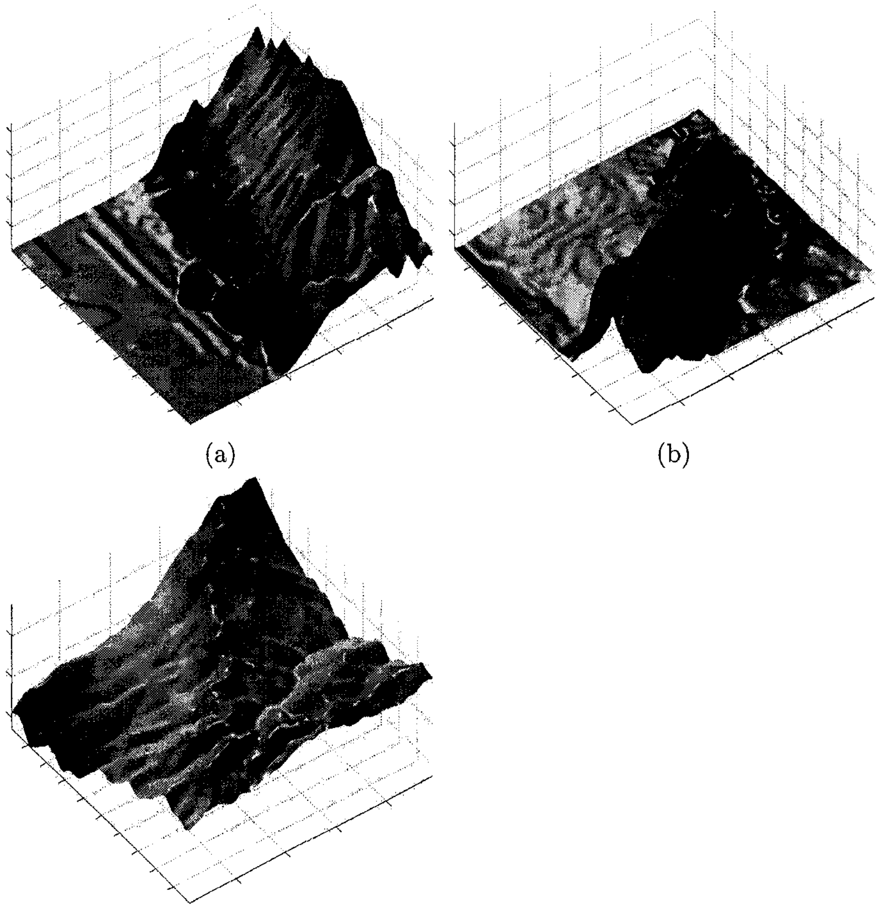
(a) (b) (c) (d)

(e) (f) (g) (h)

(i) (j) (k) (l)

(m) (n)

Figure 2.6: Illustation of the watermarked image with different attacks: (a) Gaussian noise,(b) salt & pepper, (c) Gaussian low pass filter attack, (d) Cropping attack, (e) histogram equalization, (f) blurring, (g) sharpening, (h) rescaling, (i) JPEG compression, (j) Gamma correction, (k) deblurring with oversized PSF,(l) deblurring with undersized PSF,(m) foreground,(n) multiplicative uniform noise



(a)

(b)

(c)

Figure 2.7: (a) Vine hill, (b) Santateresa hills,(c) Copper mountain

of the critical points:

$$\lambda_{wi} = (\lambda_w^i - \lambda_o^i)/(\alpha)$$

where λ_w^i denotes the transformed critical points, λ_o^i denotes the transformed original critical points, and α is a constant strength factor.

2) For each point we compute the watermark value inserted in x,y and z, then we compute the summation of these three values and divide the outcome by 3. If the result is less than 0.5, then the number is considered to be 0 otherwise it's 1.

3) We will store the output in a vector ν with the same size as the number of critical points and contains all the binary numbers that were inserted in all critical points.

4) The vector ν is divided by the number of times we inserted the binary number to get equal vectors.

5) Element by element summation is applied to all corresponding elements of each vector and divide the outcome by the number of vectors. If the result is less than 0.5, then number is considered to be 0 otherwise it's 1.This will produce the extracted binary number.

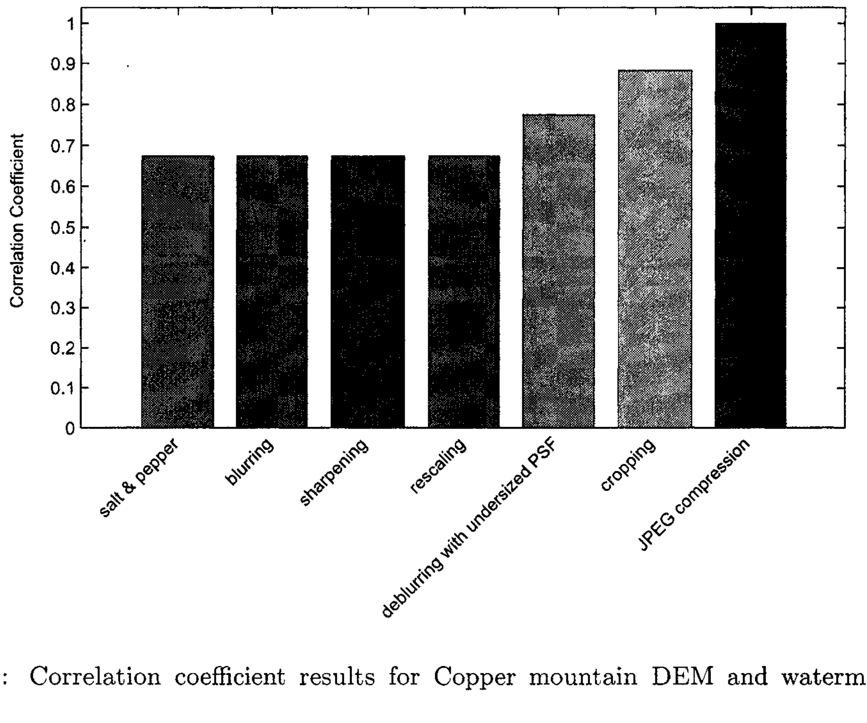


Figure 2.8: Correlation coefficient results for Copper mountain DEM and watermark binary number.

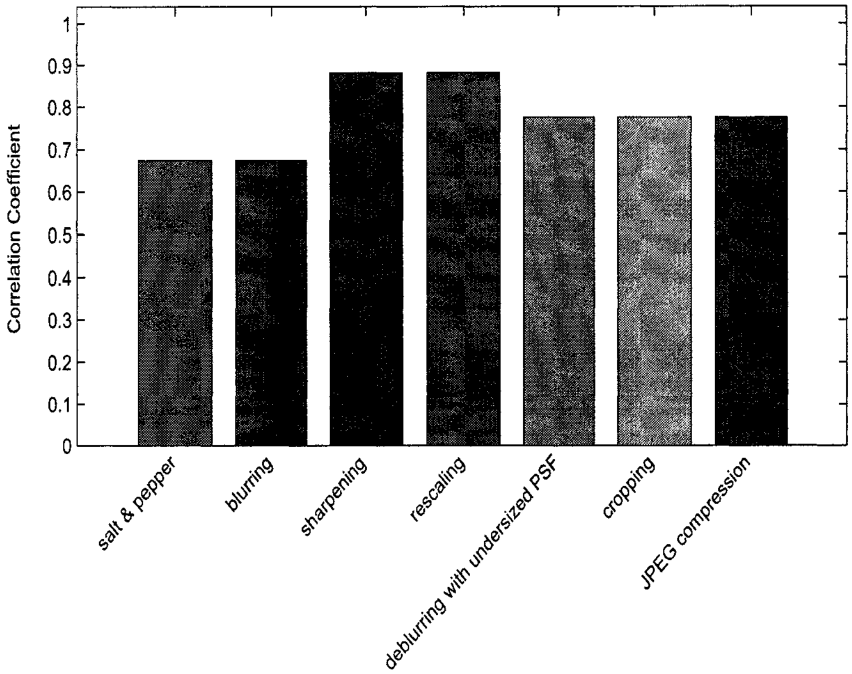


Figure 2.9: Correlation coefficient results for Vine hill DEM and watermark binary number.

2.3 Experimental Results

In this section, we performed several experiments on a DEM to test the effectiveness of our proposed scheme. Those test show the imperceptibility of the watermark and the robustness against attacks. The binary number we used for the watermark is 16 bits and the constant scaling factor a set to 0.9.

2.3.1 Robustness Evaluation

To verify the robustness of our proposed method, we applied several attacks to the DEM after the binary number insertion. The attacks include JPEG compression, Gaussian noise, multiplicative noise, Gaussian filter, deblurring with undersized point-spread function (PSF), deblurring

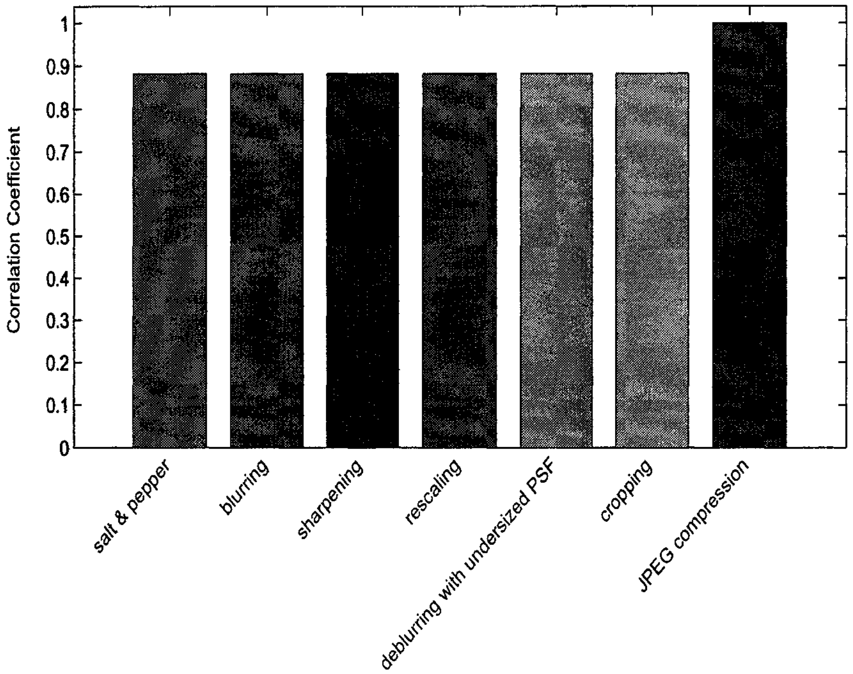


Figure 2.10: Correlation coefficient results for Santateresa hill DEM and watermark binary number.

with oversized point-spread function (PSF), Gamma- correction, histogram equalization, cropping, rescaling, sharpening, contrast adjustment, brightness change, motion blurring, and foreground see Fig. 4.2. To evaluate the performance of the watermark extraction we looked at the correlation coefficient ρ for the extracted watermark and the original embedded watermark:

$$\rho = \frac{\sum_{i,j=1}^{n}(W_{ij} - \overline{W})(\widehat{W}_{ij} - \overline{\widehat{W}})}{\sqrt{\left(\sum_{i,j=1}^{n}(W_{ij} - \overline{W})^2\right)\left(\sum_{i,j=1}^{n}(\widehat{W}_{ij} - \overline{\widehat{W}})^2\right)}}$$

where W is the original watermark, $\widehat{W}$ is the extracted watermark, $\overline{W}$ and $\overline{\widehat{W}}$ are the mean values of the original watermark and the extracted watermark respectively. From the results obtained from Fig. 2.8,Fig. 2.9 and Fig. 2.10 indicate that the proposed method shows great performances in terms of robustness against the attacks. Moreover, it is worth mentioning that we obtained the same correlation coefficient result for some attacks because these attacks has the same affects on the critical points we used to emmbed and extract the binary number.

2.3.2 Invisibility

To measure the perceptual quality of the DEM after the binary number insertion, we calculate the peak signal-to-noise ratio (PSNR) [32] which is defined as:

$$PSNR = 20\log_{10}\left(\frac{MAX_i}{\sqrt{MSE}}\right)$$

where MAX_i is the maximum value that the elements in the image can take, and MSE is the mean squared error between the original image and the watermarked image, and is defined as:

$$MSE = \frac{1}{m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}\|C_{ij} - \widehat{C}_{ij}\|^2$$

Fig. 2.11 show the PSNR experimental results.

2.4 Conclusions

In this chapter, we proposed a new digital watermarking technique to help protect DEM data from illegal redistribution. We conducted several experiments to test our proposed method. The experimental results demonstrate the good performance of the proposed fingerprinting method, imperceptibility and robustness against attacks.

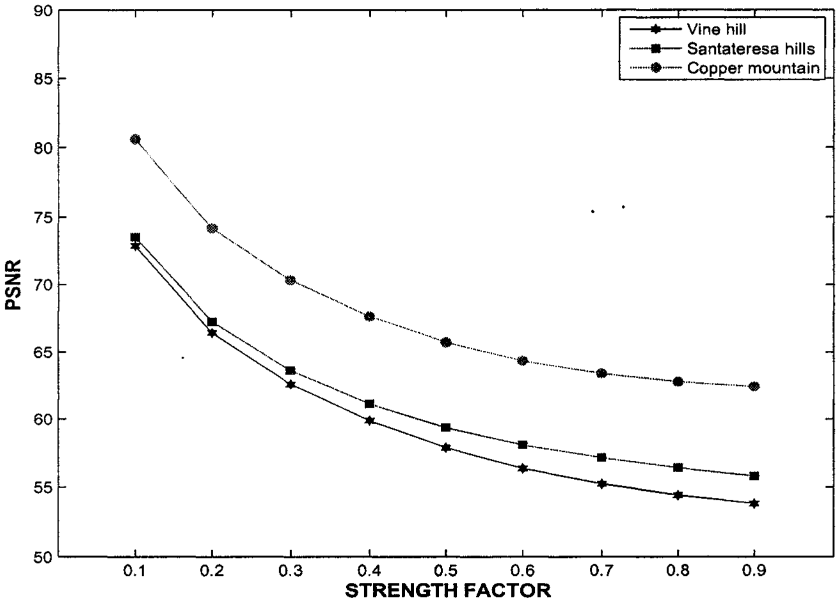


Figure 2.11: PSNR between different DEM's and their corresponding watermarked binary number with different strength factors.

Chapter 3

Fingerprinting of 3D objects

Identification and authentication of the 3D models has become one of the most important aspects of multimedia security. In this chapter we present a new robust hashing technique for 3D models. The main idea is to partition a 3D model into sub-meshes, then apply the eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh, followed by computing the hash value of each sub-mesh. The hash value is defined in terms of values and the entropy of the sub-mesh. The experimental results on a variety of 3D models demonstrate the effectiveness of the proposed technique in terms of robustness against the most common attacks including Gaussian noise, mesh smoothing, mesh compression, scaling, rotation as well as combinations of these attacks.

3.1 Introduction

The increasing use of 3D models in multimedia applications and the wide demand of online services have opened the doors for users to modify digital content without making any perceptual traces. To tackle this problem, cryptographic hash functions could help in ensuring the authentication and the integrity of data. Cryptographic hash functions play an important role in modern cryptography [33]. Hash functions take an input of arbitrary length to produce an output of fixed length referred to as hash.

The authenticity of the data can be verified by recalculating the hash value from the data and comparing it to the attached hash value. Recently, Venkatesan *et al.* [34] introduced a method for robust image hashing. This technique uses randomized signal processing strategies [35] for a non-reversible compression of images into random binary strings, and is shown to be robust against image changes due to compression, geometric distortions, and other attacks. Another robust image hashing technique was proposed in [36]. The algorithm presents a framework for perceptual image hashing using feature points [37]. Significant image features are extracted by using a wavelet-based feature detection algorithm using on the characteristics of the visual system [38]. The hash algorithm withstands standard benchmark attacks and common signal processing operations. In [39], a novel algorithm for generating an image hash based on Fourier transform features and controlled randomization was proposed. This scheme shows its resiliency to content-preserving modifications.

3D mesh hashing is a relatively new area compared to 2D hashing. It has received less attention partly because the technology that has been used for the image and video analysis cannot be easily adapted to 3D objects. Also, a large number of attacks can applied to 3D meshes. In [40], the mesh Laplacian matrix was used to encode the 3D shape into a more compact representation by retaining the smallest eigenvalues and associated eigenvectors which contain the highest concentration of the shape information. In [41], an enhanced geometric hashing method for object recognition was presented. This method identifies objects in the presence of noise and partial occlusion. In [42], a public authentication of 3D mesh models was presented. The signature is embedded within the mesh model for authentication. A new hash value is produced and compared with the value decrypted from the retrieved signature.

The primary motivation of the proposed method is to produce a hash value from a 3D model and protect it from multiple attacks. The hash value is then used for different purposes such as authentication, integrity,... etc. Our approach partitions a 3D mesh into sub-meshes and produces the hash values for each sub-mesh. To gain further insight into the proposed method, we performed extensive numerical experiments to demonstrate the potential and the much improved performance of the proposed scheme in 3D object authentication.

The remainder of this chapter is organized as follows. The next section is devoted to the problem formulation, followed by a brief background material about Laplace-Beltrami matrix and minimal spanning tree. We also implemented 3D extension of MeTiS mesh partitioning. Section 3.3 describes in detail the proposed method and the main algorithmic steps. In section 3.4, we present some experimental results to show the performance of the proposed method and its robustness against the most common attacks. Finally, we conclude in section 3.5. .

3.2 Problem formulation

In computer graphics and geometric-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is usually denoted by $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{v_1, \ldots, v_m\}$ is the set of m vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges with cardinality $|\mathcal{E}|$, and $\mathcal{T} = \{t_1, \ldots, t_n\}$ is the set of triangles.

The hash value of the 3D model can be defined in terms of spectral values and the entropy of the sub-meshes. The goal of the hashing 3D triangle meshes is to produce a unique identifier for the 3D model that satisfies three requirements [33]. First, given a 3D model $\mathbb{M}$ and a hash function H, the computation of the hash value $H(\mathbb{M})$ must be easy. Second, Given h, it is hard to find a 3D model $\mathbb{M}$ such that $h = H(\mathbb{M})$. Third, it is hard to find two different 3D model $\mathbb{M}_1$ and $\mathbb{M}_2$ that produce the same hash value.

3.2.1 Laplacian matrix of a triangle mesh

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

3.2.2 Laplace-Beltrami matrix of a triangle mesh

The Laplace-Beltrami operator $\Delta_m \boldsymbol{v}_i$ is defined as

$$\Delta_m \boldsymbol{v}_i = \frac{3}{A} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} (\cot \alpha_{ij} + \cot \beta_{ij})(\boldsymbol{v}_j - \boldsymbol{v}_i),$$

where $\boldsymbol{v}_i^\star$ is the neighborhood of a vector $\boldsymbol{v}_i$, α_{ij} and β_{ij} are the angles $\angle \boldsymbol{v}_i \boldsymbol{v}_{j-1} \boldsymbol{v}_j$ and $\angle \boldsymbol{v}_i \boldsymbol{v}_{j+1} \boldsymbol{v}_j$ respectively, and A is the sum of all the areas of neighboring triangles defined as

$$A = \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^\star)} A(\boldsymbol{t}_j)$$

Fig. 3.1 shows the angles α_{ij} and β_{ij} of a 3D triangle mesh. Fig. 3.2 illustrates an example of a 3D triangle mesh and its sparse Laplace-Beltrami matrix.

For a function $\varphi : V \to \mathbb{R}$, we may write the Laplace-Beltrami operator as

$$\Delta_m \varphi(\boldsymbol{v}_i) = \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} w_{ij}(\varphi(\boldsymbol{v}_i) - \varphi(\boldsymbol{v}_j)),$$

Where

$$w_{ij} = \frac{3(\cot \alpha_{ij} + \cot \beta_{ij})}{A}$$

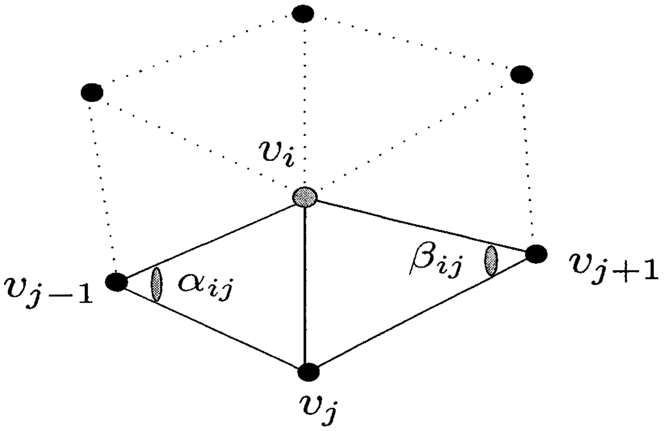


Figure 3.1: Illustration of the angles α_{ij} and β_{ij}

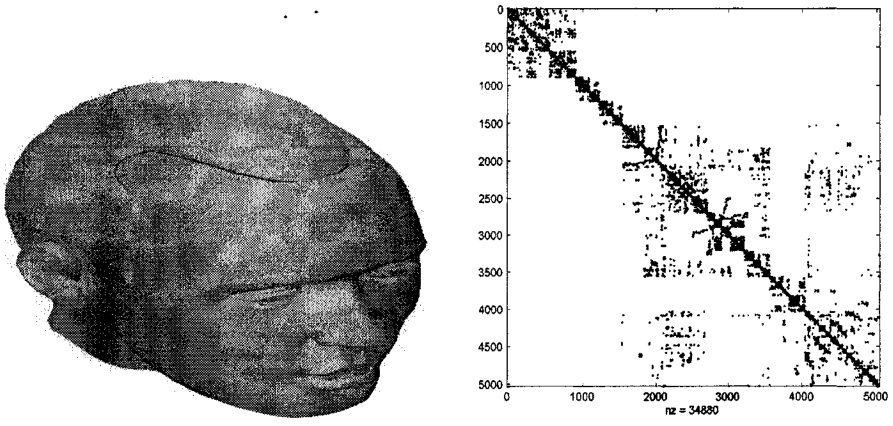


Figure 3.2: 3D triangle mesh and its Laplace-Beltrami matrix

In matrix form the Laplace-Beltrami matrix is given by $L_m = D - W$. The normalized Laplace-Beltrami matrix is given by $\mathcal{L}_m = D^{-1/2} L_m D^{-1/2} = I - D^{-1/2} W D^{-1/2}$.

3.2.3 Laplace-Beltrami approximation

We used the MATLAB notations where ./ and sqrt are pointwise operations. The approximation of the eigenfunctions and eigenvalues of the Laplace-Beltrami operator are obtained from the diagonalization of the matrix K constructed as follows:

1) Form a matrix G with entries $\exp(- \parallel \boldsymbol{v}_i - \boldsymbol{v}_j \parallel^2 / \varepsilon)$, where

$$\varepsilon = \frac{1}{m} \sum_{i=1}^{m} \min_{j:\boldsymbol{v}_j \neq \boldsymbol{v}_i} \parallel \boldsymbol{v}_i - \boldsymbol{v}_j \parallel^2$$

2) Set $K_1 = A \cdot *G$, where A is the mesh adjacency matrix, and $\cdot *$ denotes element-by-element multiplication

3) Set $\boldsymbol{p} = K_1 * 1$, where $1 = (1, ..., 1)'$

4) Define $K_2 = K_1 \cdot /(\boldsymbol{p} * \boldsymbol{p}')$

5) Set $\boldsymbol{v} = sqrt(K_2 * 1)$

6) Define $K = K_2 \cdot /(\boldsymbol{v} * \boldsymbol{v}')$

7) Diagonalize K by $[U, S, V] = svd(K)$

8) The eigenvalues of the Laplace-Beltrami are approximated by those of K, and its eigenfunctions are approximated by $U(:, \boldsymbol{i}) \cdot /U(:, 1)\cdot$

3.2.4 Minimal Spanning tree

A spanning tree $\mathcal{E}$ is a connected acyclic graph that passes through all features and it is specified by an ordered list of edges e_{ij} connecting certain pairs $(\boldsymbol{v}_i, \boldsymbol{v}_j), i \neq j$, along with a list of edge adjacency relations. The edges e_{ij} connect all m features such that there are no paths in the graph that lead back to any given feature vector. The total length $L_{\mathcal{E}}(\mathcal{V})$ of a tree is given by

$$L_{\mathcal{E}}(\mathcal{V}) = \sum_{e_{ij} \in \mathcal{E}} \parallel e_{ij} \parallel .$$

The minimal spanning tree $\mathcal{E}^{\star}$ is the spanning tree (see Fig. 3.3) that minimizes the total edge length $L_{\mathcal{E}}(\mathcal{V})$ among all possible spanning trees over the given features

$$L^{\star}(\mathcal{V}) = \sum_{e_{ij} \in \mathcal{E}^{\star}} \parallel e_{ij} \parallel = \min_{\mathcal{E}} L_{\mathcal{E}}(\mathcal{V}).$$

The set $\mathcal{V}$ is called a random feature set if its elements are random variables with a probability density function f. Then it can be shown that

$$\lim_{m \to \infty} \frac{L^{\star}(\mathcal{V})}{m^{\alpha}} = \beta \int f(x)^{\alpha} dx \qquad a.s. \tag{1}$$

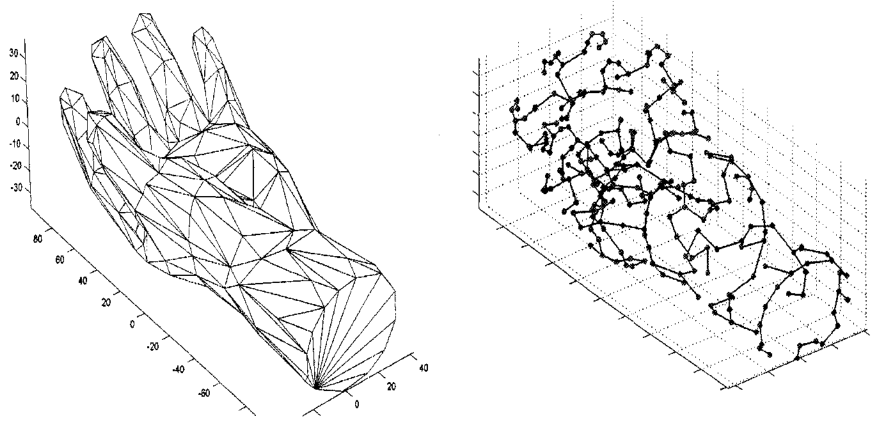
where the constant β plays a role of bias correction [43]. Hence, we may define an estimator $\widehat{H}_{\alpha}$ of Tsallis entropy as

$$\widehat{H}_{\alpha}(\mathcal{V}) = \frac{1}{1-\alpha}[\frac{L^{\star}(\mathcal{V})}{\beta m^{\alpha}} - 1], \tag{2}$$

with $L^{\star}(\mathcal{V})$ the total length of the MST. This estimator is asymptotically unbiased and almost surely consistent estimator of the Tsallis entropy [43–45].

3.3 Mesh Partitioning

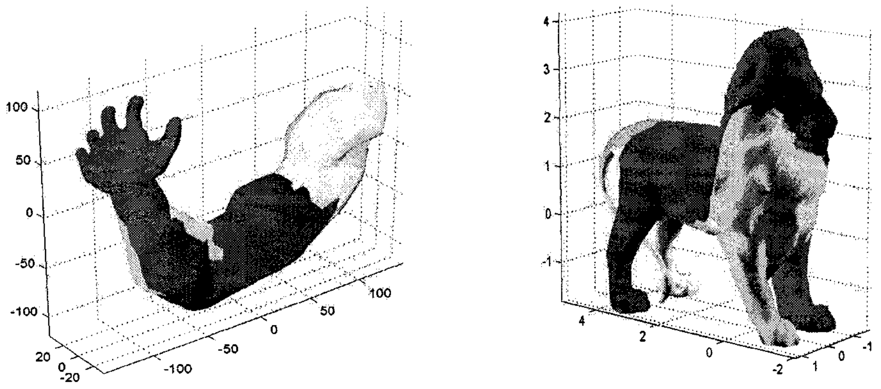
Calculating of the eigenvalues and the eigenvectors of a large $m \times m$ Laplace-Beltrami matrix is prohibitively expensive $\mathcal{O}(m^3)$. To circumvent this limitation, The hashing algorithm is applied to



(a) (b)

Figure 3.3: Illustration of an MST. (a) Hand model, (b) the MST.

the sub-meshes of the original 3D model, which is partitioned using the MeTiS mesh partitioning approach [46–48] Fig . 3.4 shows two sample 3D models partitioned into eight parts.



(a) (b)

Figure 3.4: MeTis mesh partitioning. Each sub-mesh is colored by a random color. (a) Arm model, (b) Lion model

3.3.1 Proposed Watermarking Method

In this section, we describe the main steps of the 3D hash algorithm. The goal of our proposed approach may be described as follows:

1) Divide a 3D object $\mathbb{M}$ into s sub-meshes $\mathbb{M} = \cup_{k=1}^{s} \mathbb{M}_k$.

1.1) Apply the MST to each sub-mesh $\mathbb{M}_k$ to obtain the entropy values ξ_k defined by:

$$\xi_k \simeq 2\Big(\frac{L^\star(\mathcal{V}_k)}{\sqrt{m_k}}\Big)$$

where $L^\star(\mathcal{V}_k)$ is the total length of the MST.

1.2) Apply the eigen-decomposition to the matrix L_k defined by: $L_k = B_k \Lambda_k B_k^T$, where L_k is the laplace-Beltrami matrix of each sub-mesh, $B_k = (\boldsymbol{b}_1, \boldsymbol{b}_2, \ldots, \boldsymbol{b}_{m_k})$ is an orthogonal matrix whose columns are the eigenvectors which we refer to as the spectral coefficient vectors, and $\Lambda_k = diag(\lambda_i : i = 1, \ldots m_k)$ is a diagonal matrix of eigenvalues arranged in decreasing order of magnitude.

1.3) Reduce the dimensionality of the spectral basis so that the most of the energy which is concentrated in the low frequency basis functions is used. Hence B_k and Λ_k may be expressed as B_{rk} and Λ_{rk} respectively, where r is chosen to be smaller than m_k.

1.4) Compute the hash values of each sub-mesh according to $\mu_k = \sum_{i=1}^{r} (\boldsymbol{b}_i \boldsymbol{b}_i{}^T)(\lambda_i)^{\xi_k}$

2) Combine all the values obtained from μ_k into one vector which we refer to as the hash vector H, by $H = (\mu_1, \mu_2 \ldots, \mu_k)$

3.4 Experimental Results

In this section, we present results with our experiments to assess the performance of the proposed method. A variety of 3D models are used in the experiments as depicted in Fig. 3.5.

Fig. 3.6 depicts examples of the 3D camel and 3D cow partitioned meshes.

We tested our proposed scheme by applying the algorithm to the 3D cow model. Fig. 3.8(a) through Fig. 3.8(h) show the MST of each 3D camel sub-mesh, while Fig. 3.10(a) through Fig. 3.10(h) show the MST of each 3D cow sub-mesh. The labels below the sub-meshes indicate their hash values.

3.4.1 Robustness Evaluation

To test the robustness of the method, we applied multiple attacks to the 3D models including scaling, rotating, mesh smoothing, mesh simplification, Gaussian noise, and Gaussian noise combined with compression. We evaluate the performance of the proposed scheme by computing the normalized correlation ρ between two hash values according to [49] :

$$\rho = \frac{|H_1 \ . \ H_2|}{||H_1||_2 \ . \ ||H_2||_2} \tag{3}$$

where H_1 is the hash value before the attack and H_2 is the hash value after the attack. Fig. 3.12 and Fig. 3.14 illustrate the 3D camel and 3D cow models with the listed attacks. The label below each 3D model indicates the normalized correlation between the original hash value and the hash value after the attack. The normalized correlation results clearly demonstrate the good performance of the proposed method in terms of robustness against the attacks.

Table 3.1: Normalized hash correlation with different 3D models

Attacks	Camel	Cow	Shark	Triceratops	Baby	arm
Mesh Scaling with X*2	0.9674	0.9908	0.7539	0.7755	0.5907	0.7704
Mesh Scaling with Y*2	0.9625	0.9910	0.7565	0.7864	0.6017	7784
Mesh Scaling with Z*2	0.9553	0.9221	0.7433	0.7637	0.5916	7673
Rotation around X 45°	0.9534	0.9880	0.7486	0.7807	0.5864	0.7812
Rotation around Y 45°	0.9534	0.9880	0.7486	0.7807	0.5864	0.7812
Rotation around Z 45°	0.9534	0.9880	0.7486	0.7807	0.5864	0.7812
Mesh Smoothing 10 iterations	0.9531	0.9886	0.7343	0.7765	0.5922	0.7643
Mesh Simplification 70%	0.7965	0.8448	0.7247	0.7641	0.8862	0.7621
Gaussian Noise $\sigma = 0.25$	0.9572	0.9893	0.7606	0.7931	0.6153	0.7886
Gaussian Noise + Compression 25%	0.9617	0.9902	0.7450	0.7868	0.6198	0.7843

In addition, we applied the listed attacks to the 3D models bunny, horse, hippo, and max planck. Table 1 lists the normalized hash correlation results for different 3D models. It is clearly shown that our proposed approach gives very good results and shows its consistency with a variety of 3D models.

3.4.2 Uniqueness

Uniqueness is an important factor that needs to be taken into consideration when dealing with hash functions. As mentioned earlier, the hash value produced by our proposed method should

Table 3.2: Normalized correlation between different 3D model hashes

	Camel	Cow	Shark	Triceratops	Baby	arm
Camel	1	0.7987	0.7458	0.7668	0.8112	0.7834
Cow	0.7987	1	0.7842	0.8590	0.8855	0.7921
Shark	0.7458	0.7842	1	0.7816	0.5129	0.7172
Triceratops	0.7668	0.8590	0.7816	1	0.7179	0.7145
Baby	0.8112	0.8855	0.5129	0.7179	1	0.8328
Arm	0.7834	0.7921	0.7172	0.7145	0.8328	1

always be unique. Therefore, we compared the hash vectors between different 3D models using the normalized correlation coefficient to check whether the proposed hash vector fulfills the requirement of uniqueness. The results are listed in Table 2. It is apparent that the proposed method shows very good performance in terms of the ability to distinguish different 3D models and to produce different hash values.

3.5 Conclusions

In this chapter, we proposed a simple and robust hashing scheme for 3D models. The key idea is to partition a 3D model into sub-meshes, followed by applying the eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh and obtain the hash values of all sub-meshes. The performance of the proposed method was evaluated through extensive experiments which clearly showed excellent resiliency against multiple attacks.

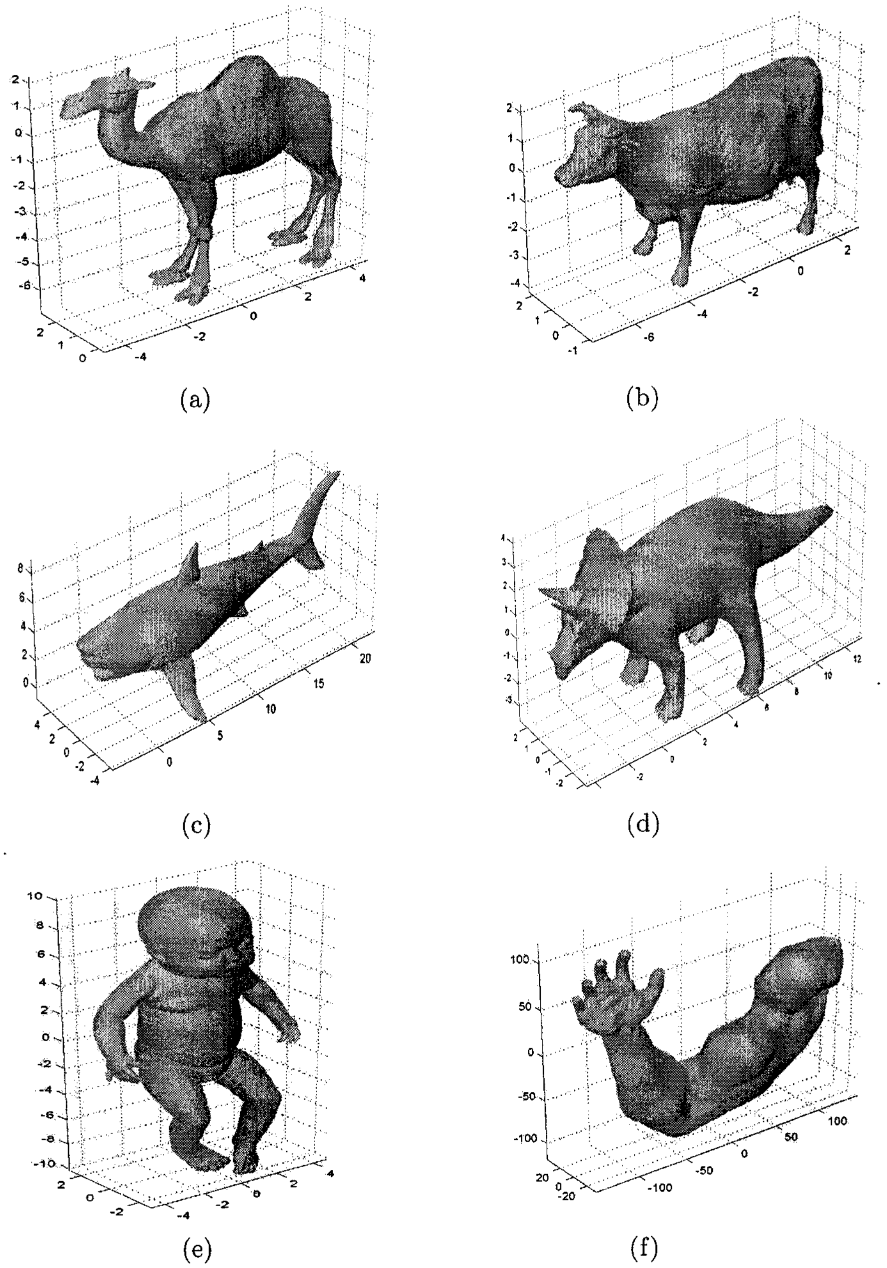
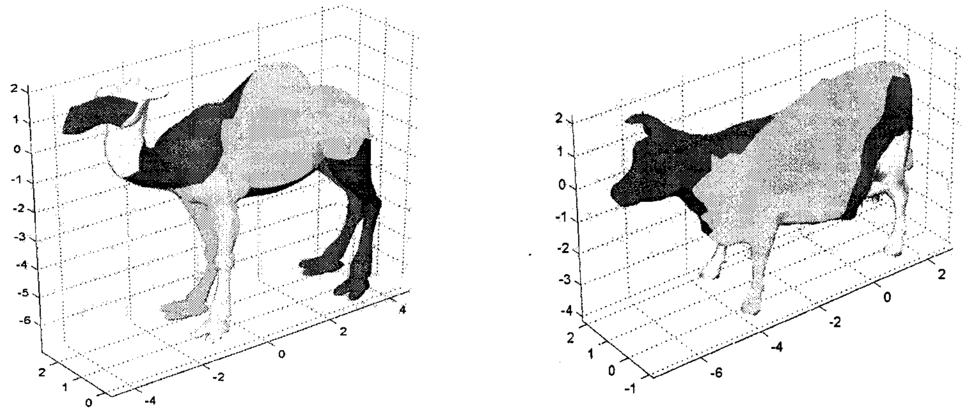
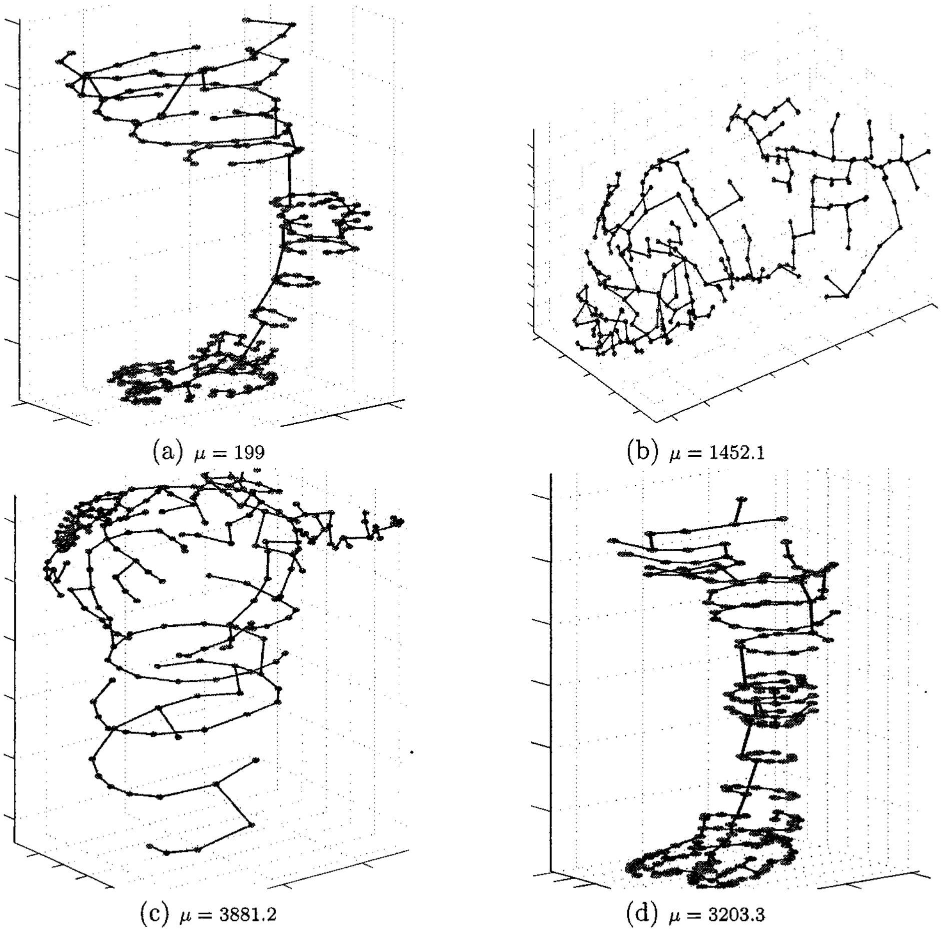


Figure 3.5: Original 3D models used for experimentation: (a) camel, (b) cow, (c) shark, (d) triceratops, (e) baby, (f) arm.



(a) (b)

Figure 3.6: Partitioned 3D models. (a) camel, (b) cow



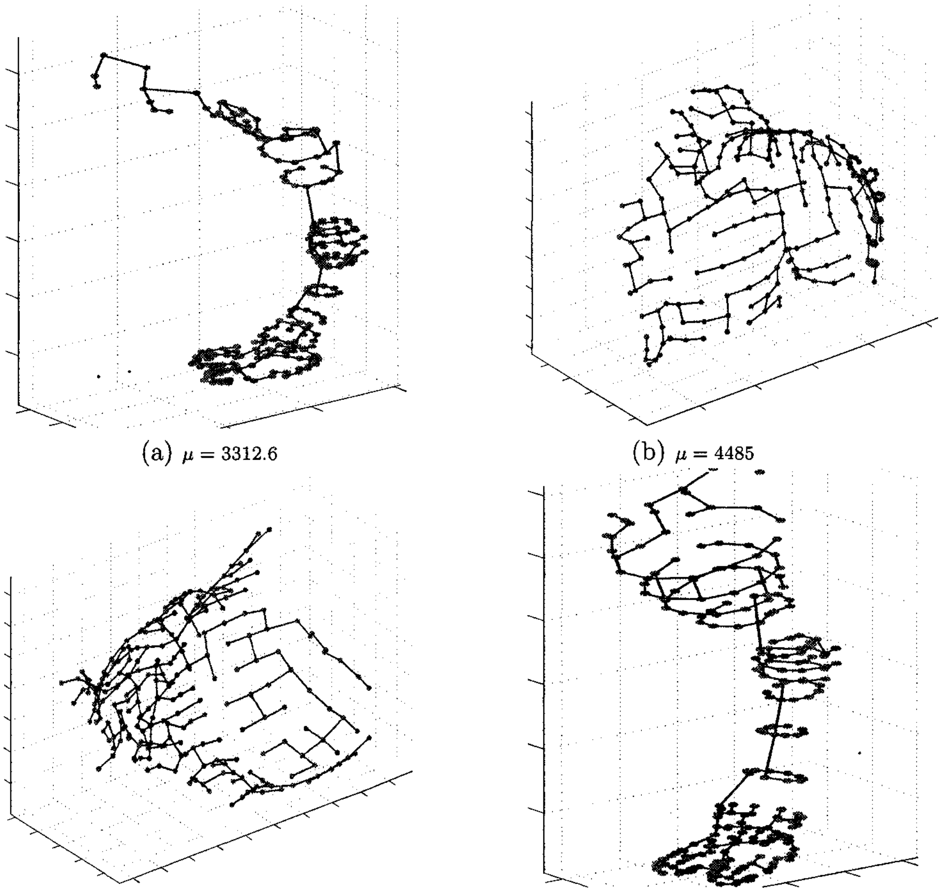
(a) $\mu = 199$

(b) $\mu = 1452.1$

(c) $\mu = 3881.2$

(d) $\mu = 3203.3$

Figure 3.7: MST of the 3D camel sub-meshes and their corresponding hash values : (a) Head, (b) Neck, (c)-(d) Front feet.



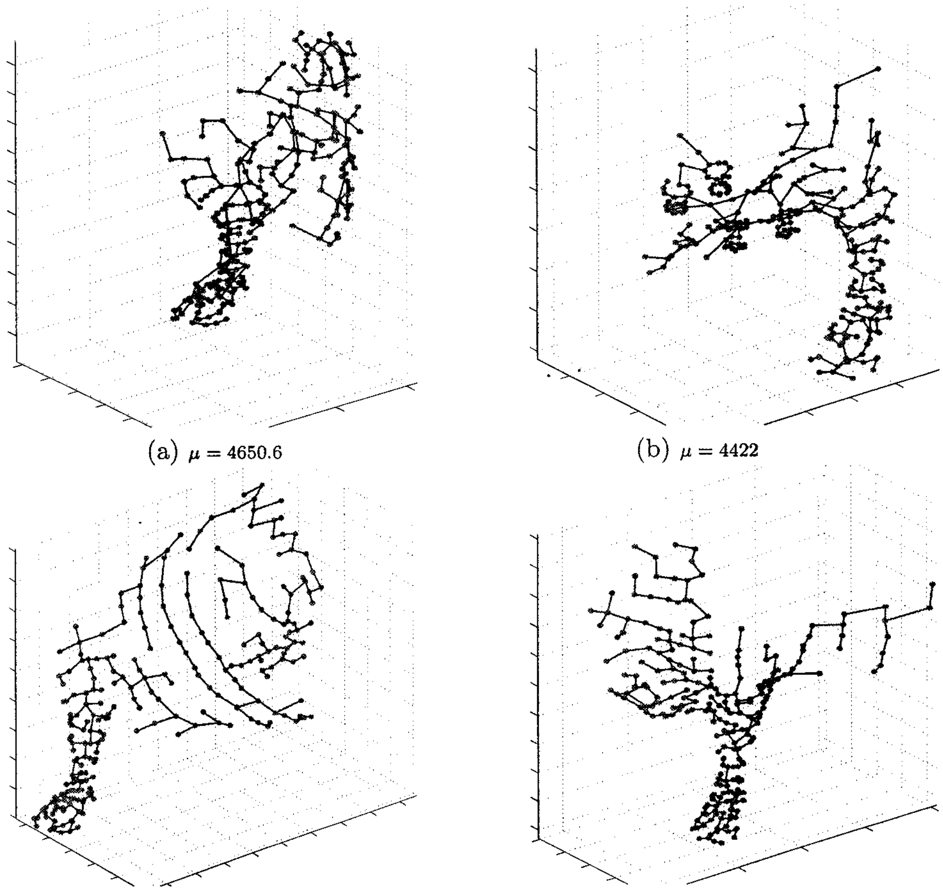
(a) $\mu = 3312.6$

(b) $\mu = 4485$

(c) $\mu = 3411$

(d) $\mu = 4069.1$

Figure 3.8: MST of the 3D camel sub-meshes and their corresponding hash values : (a) Hump, (b) Shoulders, (c)-(d) Back.



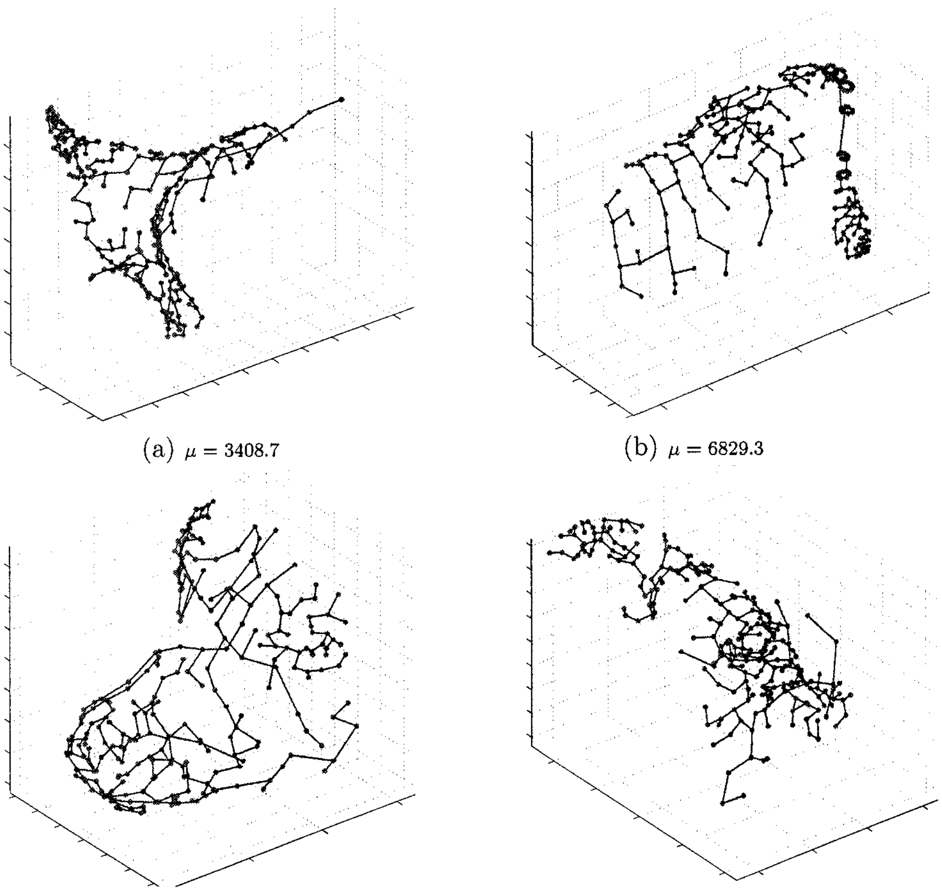
(a) $\mu = 4650.6$

(b) $\mu = 4422$

(c) $\mu = 5771$

(d) $\mu = 4264.9$

Figure 3.9: MST of the 3D cow sub-meshes and their corresponding hash values : (a)-(b) Back feet, (c)-(d) Front feet.



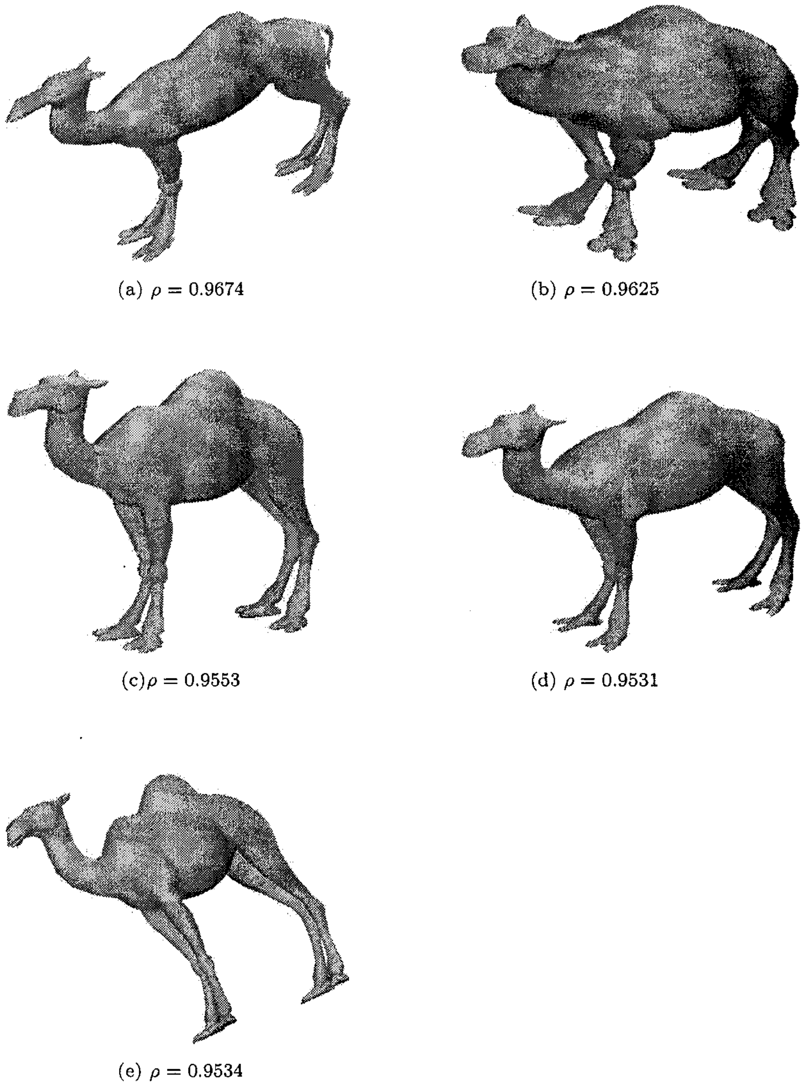
(a) $\mu = 3408.7$

(b) $\mu = 6829.3$

(g) $\mu = 1453.4$

(h) $\mu = 1696.4$

Figure 3.10: MST of the 3D cow sub-meshes and their corresponding hash values : (a) Neck, (b)Back-tail, (c)-(d) Head-horn.



(a) $\rho = 0.9674$

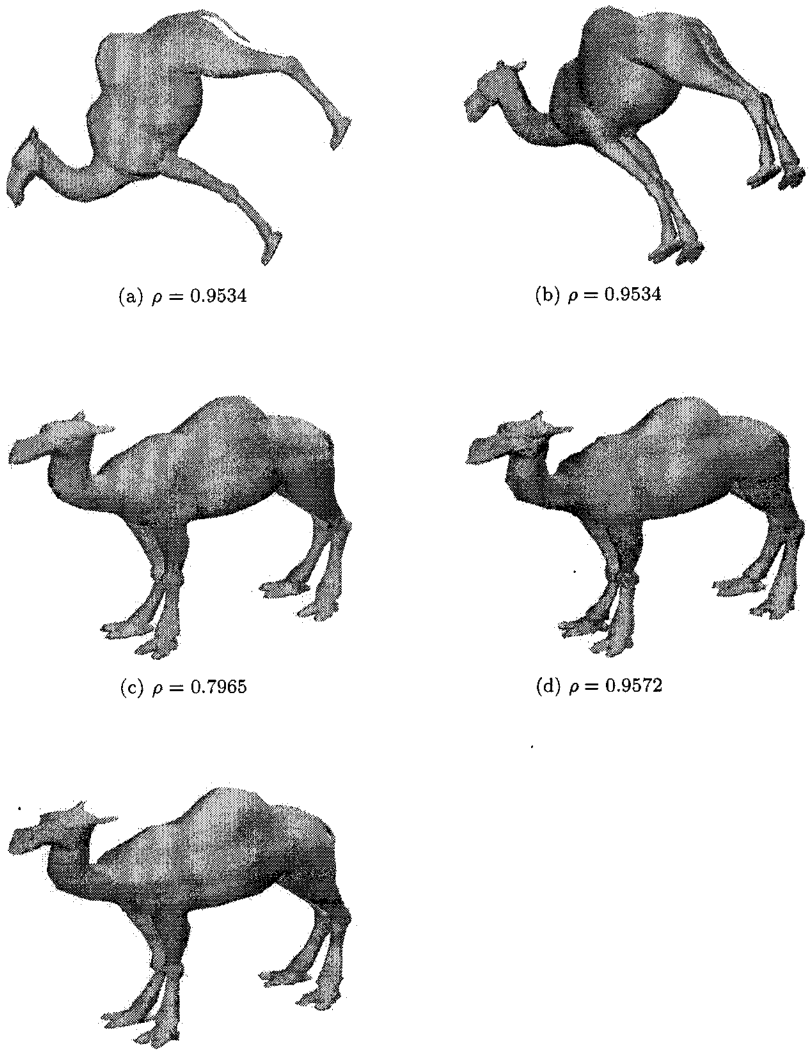
(b) $\rho = 0.9625$

(c)$\rho = 0.9553$

(d) $\rho = 0.9531$

(e) $\rho = 0.9534$

Figure 3.11: Illustration of the 3D camel model with different attacks. (a) scaling with X-axis, (b) scaling with Y-axis, (c) scaling with Z-axis, (d) mesh smoothing 10 iterations, (e) rotating around X-axis 45^o.



(a) $\rho = 0.9534$

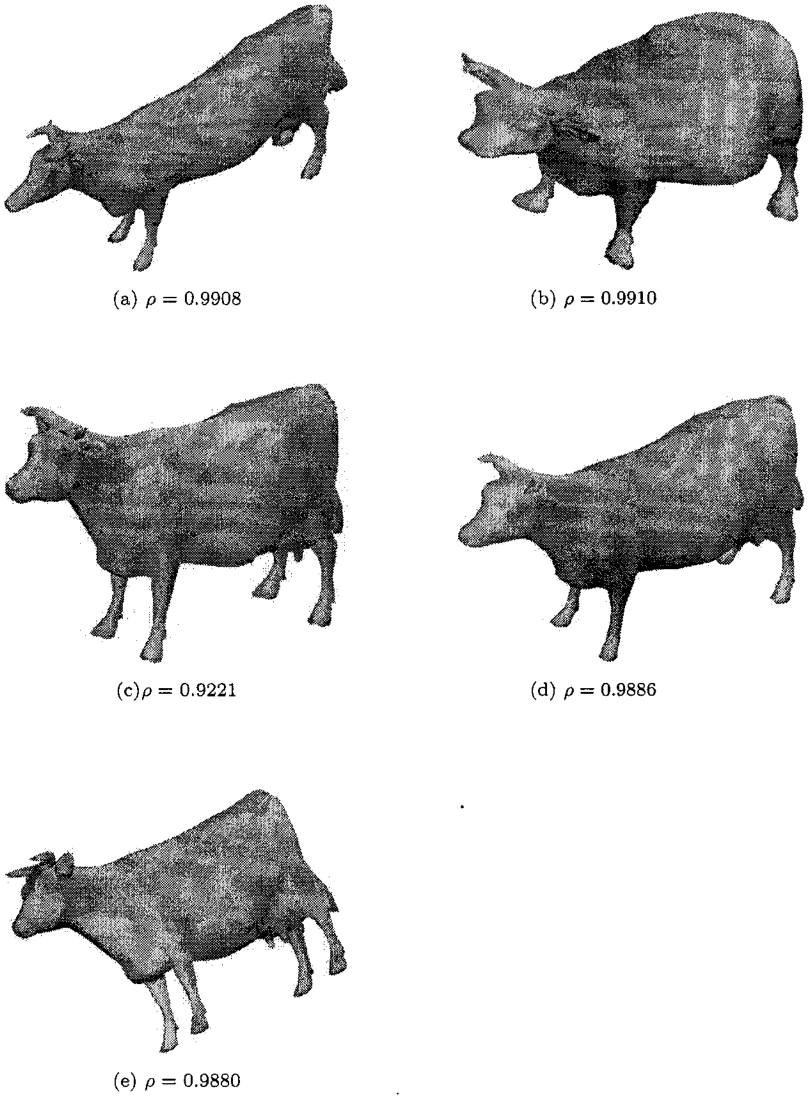
(b) $\rho = 0.9534$

(c) $\rho = 0.7965$

(d) $\rho = 0.9572$

(f) $\rho = 0.9617$

Figure 3.12: Illustration of the 3D camel model with different attacks. (a) rotating around Y-axis 45^o, (b) rotating around Z-axis 45^o, (c) simplification 70% , (d) Gaussian noise $\sigma = 0.25$, (e) Gaussian noise combined with compression 25%.



(a) $\rho = 0.9908$

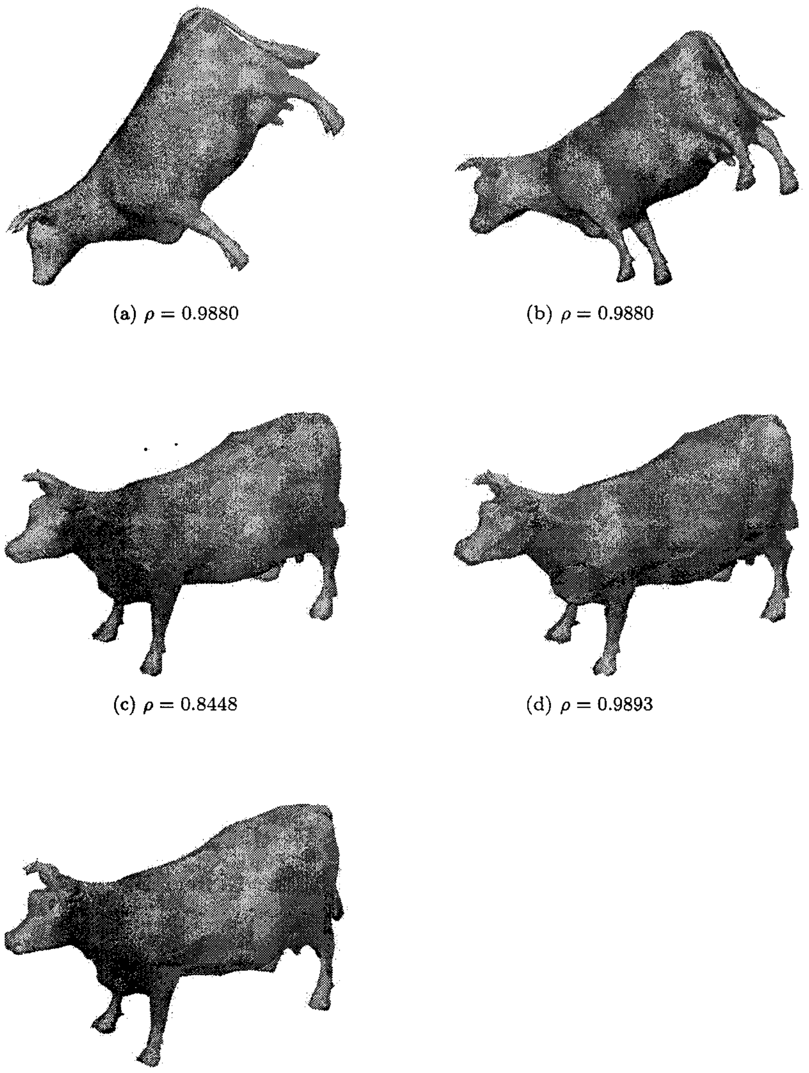
(b) $\rho = 0.9910$

(c)$\rho = 0.9221$

(d) $\rho = 0.9886$

(e) $\rho = 0.9880$

Figure 3.13: Illustration of the 3D cow model with different attacks. (a) scaling with X-axis, (b) scaling with Y-axis, (c) scaling with Z-axis, (d) mesh smoothing 10 iterations, (e) rotating around X-axis 45°.



(a) $\rho = 0.9880$

(b) $\rho = 0.9880$

(c) $\rho = 0.8448$

(d) $\rho = 0.9893$

(e) $\rho = 0.9902$

Figure 3.14: Illustration of the 3D cow model with different attacks.(a) rotating around Y-axis 45^o, (b) rotating around Z-axis 45^o, (c) simplification 70% , (d) Gaussian noise $\sigma = 0.25$, (e) Gaussian noise combined with compression 25%.

CHAPTER 4

Image Hashing By Higher-Order Statistics and Wavelets

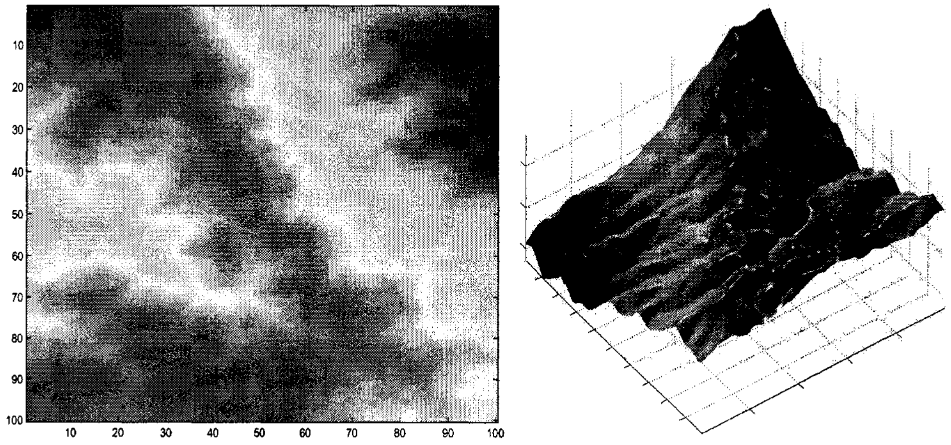
In this chapter, we propose a robust digital elevation maps (DEM) fingerprint scheme. The key idea is to apply content-based hashing algorithm to the DEM using higher-order statistics and discrete wavelet transform to define the image fingerprint. We conduct several experiments to compare the performance of our proposed method with existing techniques. Experimental results show the great performance of the proposed method in terms of robustness against a number of challenging attacks.

4.1 Introduction

The recent rapid growth of digital media contents and the increase use of online services have triggered the need for multimedia protection. Watermarking plays an important role to solve the problem of unauthorized replication [27]. Watermarking can be defined as the process of embedding data called "watermark" into a digital object without making changes to the quality of the host substantially [1,28]. The digital object could be an image, video, or audio. The watermark is used as a signature to prove ownership and can only be detected or extracted by the owner. One of the applications used for digital watermarking is fingerprinting. Image fingerprinting defines as the process of extracting a unique description of an image called a fingerprint. Image fingerprint can be used as human fingerprint to compare identity, identification and authentication.

A variety of watermarking techniques have been proposed. These techniques can be divided into two types according to the embedding domain of the cover image: spatial domain methods and transform domain methods [4, 8]. The spatial domain method directly modifies the intensities of selected pixels. Where the transform domain method is modifies the values of selected transformed coefficients. Based on a variety of studies, it is proved that the frequency domain method is superior to the spatial domain method in terms of robustness against watermarking attacks.

A digital elevation map is a raster of elevation values, and consists of an array of points of elevations, sampled systematically at equally spaced intervals. We may represent a DEM as an image $I : \Omega \subset \mathbb{R}^2 \longrightarrow \mathbb{R}$ (see Fig 4.1(a)) where each image location denotes a height value. DEMs are usually constructed from aerial photographs and require at least two images of a scene [29].



(a) (b)

Figure 4.1: Illustration of a DEM in 2D and 3D

(DEMs) or Digital elevation maps are used to provide a digital representation of surface terrains in three dimensional space (Fig 4.1(b)). Because of the large amount of effort in acquiring DEMs they have high commercial value. Also, DEMs can carry critical geospatial information when they used in sensitive military applications. For these reasons we used fingerprinting to prevent unauthorized distribution and to trace back illegally produced copies.

Recently, a digital watermark technique to protect DEM data from illegal re-distribution has been proposed [30]. The main idea of this technique is to extract a set of critical contours from a DEM depending on the important topographic features of the terrain. A watermark is then embedded into these critical contours by using parametric curve modeling and spread spectrum embedding. Finally, watermarked DEM is constructed to include the marked 2-D contours.

In this chapter we introduce a robust method for digital fingerprinting and secure copyright protection of digital elevation maps. The proposed watermarking method companied a forth-order cumulants with discrete wavelet transform (DWT) to get the image hashing which will be used as the DEM fingerprinting . The result of our propose method have an improved performance than the existing techniques.

The remainder of the chapter is organized as follows. In section 4.2, we provide a brief background material about higher-order and DWT. In section 4.3, we introduce the proposed fingerprinting algorithm and describe in more details its fundamental steps. In section 4.4, we present some experimental results to demonstrate the much improved performance of the proposed method in comparison with exiting technique. Finally, we conclude in section 4.5.

4.2 Background

4.2.1 Cumulants for image hashing

The main goal of a content-based algorithm is to produce hash output which is tolerant to content-preserving distortion but sensitive to content modification. This somehow vague criterion can be interpreted in various ways. Considering images, one possible approach is to assume that the relative relationship between pixels in a neighborhood remains approximately the same after authentic distortion, otherwise not. This motivates using joint statistics as image features, for if the pixel sequence in a neighborhood is assumed to be a random process, its auto-statistics characterize the relative relationship. In literature, Yu et al. first proposed to use higher order auto-cumulants as robust image features. The cumulant is a quantity in statistics for measuring deviation from Gaussian. Given a set of n real random variables $\{x_1, x_2, ..., x_n\}$, their joint cumulants of order $k_1 + k_2 + ... + k_n$ are defined as

$$C_{k_1...kn} \equiv (-j)^r \frac{\partial^r ln\Phi(w_1, w_2, ..., w_n)}{\partial w^{k_1} \partial w^{k_2} ... \partial w^{k_n}}|_{w_1=...=w_n=0} \tag{1}$$

Where

$$\Phi(w_1, w_2, ..., w_n) = E\{\exp j(w_1 x_1 + ... + w_n x_n)\} \tag{2}$$

Is their joint characteristic function and E means expectation. The first- and second- order cumulants are also known as the mean and the autocorrelation. Orders higher than two are called higher orders. Assuming a zero-mean stationary process $X(n)$, its second-, third-, and fourth-order cumulants are defined by:

$$C_{2x}(k) = E\{x(n)x(n+k)\} \tag{3}$$

$$C_{3x}(k,l) = E\{x(n)x(n+k)x(n+l)\} \tag{4}$$

$$\begin{aligned} C_{4x}(k,l,m) = E\{x(n)x(n+1x(n+m)\} \\ -C_{2x}(k)C_{2x}(k)C_{2x}(l-m) \\ -C_{2X}(l)C_{2x}(k-m) \\ -C_{2x}(m)C_{2x}(k-l) \end{aligned} \tag{5}$$

There might be two important reasons to use higher-order statistics for image hashing: 1) because higher-order statistics vanish for a Gaussian process, they show better resistance to Gaussian noise than lower order statistics; 2) because natural image are known to be non-Gaussian, non-Gaussianity might be used to characterize image content, i.e., content-preserving distortion should preserve the non-Gaussianity, while content modification tend to drastically change it. Assuming zero-mean random variables X and Y are independent and y is Gaussian, the particular advantage of higher-order cumulants is that

$$C_{n,X+Y} = C_{nX} + C_{nY} = C_{nX}, n \geqslant 3. \tag{6}$$

Therefore, if content-preserving distortion can be modeled as Gaussian, they can be separated by higher-order cumulants. Another advantage of using higher-order cumulants is that the one-way and collision-resistant properties can be well satisfied. In general, due to the high complexity, it is difficult to reconstruct meaningful image from given higher-order statistics [49].

4.2.2 Discrete Wavelet Transform (DWT)

The DWT provides a number of powerful image processing algorithms including noise reduction, edge detection, and compression. The DWT is computed by successive lowpass and highpass filtering of the discrete time-domain signal. Its significance is in the manner it connects the continuous time mutiresolution to discrete-time filters. At each level, the high pass filter associated with scaling function produces coarse approximations. We use a 2D version of the analysis and synthesis filter bank by applying a 1D analysis filter bank to the columns of the image and then to the rows. If the image has m rows and m columns, then after applying the 2D analysis filter bank we obtain four sub-band images(LL,LH,HL, and HH), each having $m/2$ rows and $m/2$ columns [23].

4.3 Proposed Method

In this section, we will explain the main steps of the proposed fingerprint algorithms. The goal of our proposed approach is to extract image hashing by dividing the DEM into blocks, and compute the fourth-order cumulants with DWT for each block. The proposed method consists of the following steps:

1) Divide the image into 32 x 32 blocks.

2) Re-order the pixels into a vector by a raster scan.

3) Compute forth-order cumulants for the vector.

4) Apply the discrete wavelet transform (DWT) to the fourth-order cumulants.

5) Keep and quantized the first row of the approximation coefficients matrix (CA) to use it as hash output.

4.4 Experimental Results

In this section, we perform several experiments on a DEM to test the effectiveness of our proposed scheme. Those tests show the imperceptibility of the fingerprint and the robustness against attacks.

4.4.1 Robustness Evaluation

To verify the robustness of our proposed method, we applied several attacks to the DEM. The attacks include JPEG compression, Gaussian noise, multiplicative noise, Gaussian filter, deblurring with undersized point-spread function (PSF), deblurring with oversized point-spread function (PSF), Gamma- correction, histogram equalization, cropping, rescaling, sharpening, contrast adjustment, brightness change, motion blurring, and foreground. Fig. 4.2 shows the fingerprint DEM with different kinds of attacks. Hash comparison is done by correlation. Assuming H1 and H2 are two hash vectors, the normalized correlation is used to evaluate their similarity, which is defined as:

$$\frac{\mid H_1.H_2 \mid}{\|H_1\|_2.\|H_2\|_2}$$

Recall that similar content should result in similar hashes. If two DEM's contain similar content, the normalized correlation between their hash vectors should approach 1, otherwise approach 0. [49]. The results of the normalized correlation by the cumulant algorithm using the exciting technique and our proposed technique with two different wavelet filters are shown in Fig. 4.3. Most results are quite similar, especially for non-geometric distortion. If quantization is further applied, the difference might be even smaller [49].

4.5 Conclusion

In this chapter, we proposed a new digital fingerprinting technique to help protect DEM data from illegal redistribution. We conducted several experiments to test our proposed method. The experimental results demonstrate the good performance of the proposed fingerprinting method, imperceptibility and robustness against attacks.

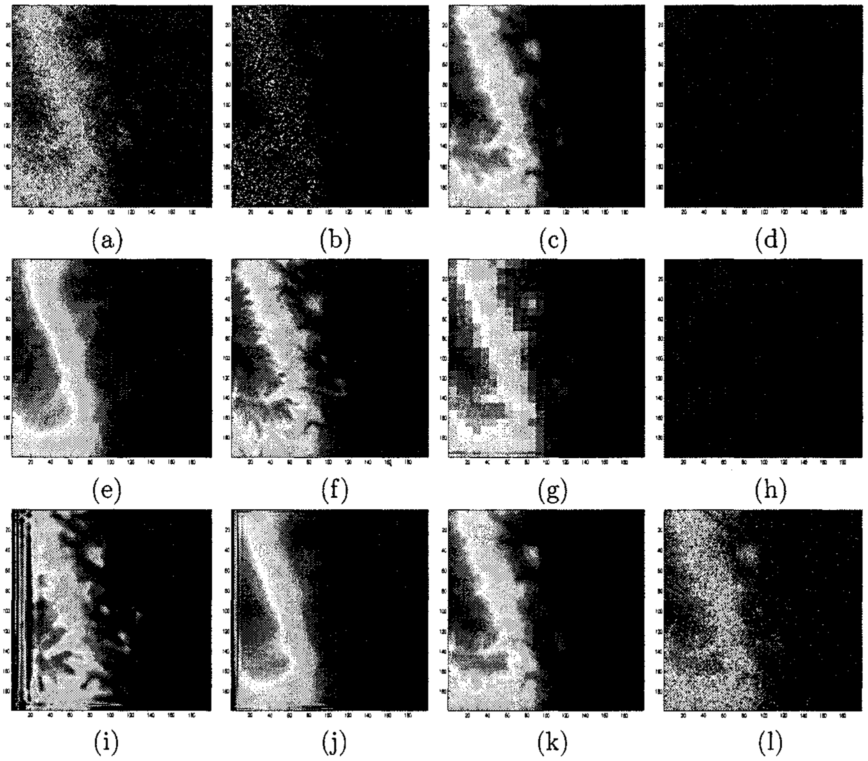


Figure 4.2: Illustation of the fingerprint DEM with different attacks: (a) Gaussian noise,(b) salt & pepper, (c) Gaussian low pass filter attack, (d) histogram equalization, (e) blurring, (f) sharpening, (g) JPEG compression, (h) Gamma correction, (i) deblurring with oversized PSF,(j) deblurring with undersized PSF,(k) foreground,(l) multiplicative uniform noise

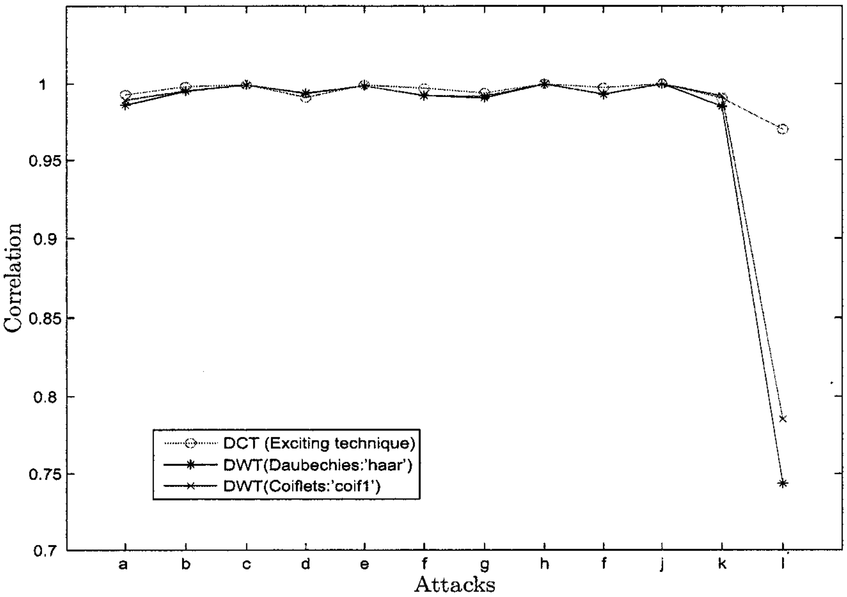


Figure 4.3: Illustation of the hash comparison with different attacks: (a) Gaussian noise,(b) salt & pepper, (c) Gaussian low pass filter attack, (d) histogram equalization, (e) blurring, (f) sharpening, (g) JPEG compression, (h) Gamma correction, (i) deblurring with oversized PSF,(j) deblurring with undersized PSF,(k) foreground,(l) multiplicative uniform noise

CHAPTER 5

Conclusions and future work

This thesis has presented robust watermarking schemes for multimedia protection as well as 3D mesh fingerprinting. We have demonstrated the performance of the proposed algorithms through extensive experiments, and we compared our techniques with existing methods. A variety of images are used in the experiments to show the effectiveness of the proposed schemes. We have achieved to balance between the imperceptibility of the watermarked image and its robustness against intentional and geometric attacks. In addition, we have developed a spectral compression technique for 3D models as well as a 3D mesh fingerprinting technique.

In the next Section, the contributions made in each of the previous chapters and the concluding results drawn from the associated research work are presented. Suggestions for future research directions related to this thesis are provided in Section 5.2.

5.1 Contributions of the thesis

5.1.1 Watermarking of digital elevation maps

We proposed a robust digital elevation map (DEM) watermarking scheme by inserting a binary number into a set of critical points. Our method starts by extracting the critical pointes from a DEM depending on the important topographic features of the terrain. Then we embedded the fingerprints into x, y and z values for all the critical points. In the experimental results, we test the robustness of the proposed method against a number of challenging attacks.

5.1.2 Fingerprinting of 3D objects

We proposed a simple and robust hashing scheme for 3D models. The key idea is to partition a 3D model into sub-meshes, followed by applying the eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh and obtain the hash values of all sub-meshes. The performance of the proposed method was evaluated through extensive experiments which clearly showed excellent resiliency against multiple attacks.

5.1.3 Image Hashing By Higher-Order Statistics and Wavelets

We proposed a robust digital elevation maps (DEM) fingerprint scheme. The key idea is to apply content-based hashing algorithm to the DEM using higher-order statistics technique and discrete wavelet transform (DWT) to define the image fingerprint. We conducted several experiments to compare the performance of our proposed method with existing techniques. Experimental results show the great performance of the proposed method in terms of robustness against a number of challenging attacks.

5.2 Future research directions

Several interesting research directions motivated by this thesis are discussed next. In addition to designing robust watermarking schemes for multimedia protection, we intend to accomplish the following projects in the near future:

5.2.1 Image watermarking using fast Hadamard, MPDFRF and wavelets

We plan to study a watermarking scheme that uses MPDFRFT, fast Hadamard transform(FHT), and DWT. Based on a variety of studies, the 2D Hadamard transform has been used with great success for image compression and image watermarking. The elements of the basis vectors of the Hadamard transform take only the binary values +1 and -1. Therefore, the FHT is well suited for digital image processing applications where computational simplicity is required.

5.2.2 3D image watermarking scheme using nonnegative transition matrix factorization and wavelet transform

We would like to develop a robust watermarking scheme for 3D models. The key idea is to apply the transition matrix to the 3D mesh and decompose it into four wavelet sub-bands and then apply NMF to the blocks of each sub-band.

5.2.3 Spectral 3D mesh watermarking

3D mesh compression technique can play a good role in improving the watermarking system. The 3D compression technique has led us to find a better way to embed the watermark. A 3D model can be partitioned into smaller sub-meshes, then apply the umbrella compression technique to each sub mesh, followed by embedding a watermark in the spectral coefficients of the compressed 3D meshes.

List of References

[1] I. Cox, M. L. Miller, and J. A. Bloom, *Digital watermarking*, Morgan Kaufmann Publishers Inc., 2001.

[2] M. Arnold, M. Schmucker, and S. D. Wolthusen, *Techniques and Applications of Digital Watermarking and Content Protection*, Artech House, 2003.

[3] A. Nikolaidis, S. Tsekeridou, A. Tefas, and V. Solachidis, "A survey on watermarking application scenarios and related attacks," *Proc. IEEE Int. Conference on Image Processing*, vol. 3, pp. 991- 994, October 2001.

[4] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proceedings of the IEEE*, vol. 87, no. 7, pp.1079-1107, 1999.

[5] S. P. Mohanty, K. R. Ramakrishnan, and M. S. Kankanhalli, "A DCT domain visible watermarking technique for images," *Proc. IEEE Int. Conference on Multimedia and Expo*, pp. 1029- 1032, 2000.

[6] S. Craver, N. Memon, B. L. Yeo, and M. M. Yeung, "Resolving rightful ownerships with

invisible watermarking techniques: Limitations, attacks, and implications," *IEEE Journal on Selected Areas in Communications,* vol. 16, no. 4, pp. 573- 586, May 1998.

[7] W. Yongdong, "On the security of an SVD-based ownership watermarking," *IEEE Transaction on Multimedia,* vol. 7, no. 4, pp.624- 627, August 2005.

[8] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing* , vol. 6, no. 12, pp. 1673-1687, 1997.

[9] R. Liu and T. Tan, "A SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 121-128, 2002.

[10] M. Barni, F. Bartolini, V. Cappelini and A. Piva, "A DCT-domain system for robust image watermarking," *Signal Processing,* vol.66, pp. 357-372, 1998.

[11] M. Kutter and F. Hartung. Introduction to Watermarking Techniques, Chapter 5 of "Information hiding: Techniques for Steganography and digital watermarking," S. Katzenbeisser and F. A. P. Peticolas(eds.), Norwood, MA: Artech house, pp. 97-120, 2000.

[12] J. Dittmann and F. Nack, "Copyright - copywrong," *IEEE Transactions on Multimedia*, vol. 7, pp. 14-17, 2000.

[13] V. Fotopoulos and A. N. Skodras, "A subband DCT approach to image watermarking," *Proc. European Signal Processing Conference*, Finland, 2000.

[14] Y. Wang, J. F. Doherty, R. E. Van Dyck, "A Wavelet-Based Watermarking Algorithm-for Ownership Verification of Digital Images," *IEEE Transactions on Image Processing*, vol. 11, no. 2, pp. 77-88, Feb 2002.

[15] E. Ganic and A. M. Eskicioglu, "Robust embedding of visual watermarks using DWT-SVD," *Journal of Electronic Imaging*, October-December 2005.

[16] E. Ganic, N. Zubair, and M. Eskicioglu, "An optimal watermarking scheme based on singular value decomposition," *Proc. Comm., Network, and Information Security*, pp. 89-90, 2003.

[17] J. Kusyk and A. M. Eskicioglu, "A Semi-blind logo watermarking scheme for color images by comparison and modification by comparison and modification of DFT coefficients," *Proc. Multimedia Systems and Applications Conference*, Boston, MA, October 2005.

[18] A. Sverdlov, S. Dexter, and A. M. Eskicioglu, " Robust DCT-SVD domain image watermarking for copyright protection : embedding data in all frequencies," *Proc. Euro. Signal Processing Conference,* Turkey, 2005.

[19] P. Tao and A. M. Eskicioglu, "A robust multiple watermarking scheme in the DWT domain," *Proc. Optics East Symposium, Internet Multimedia Management*, 2004.

[20] R. Mehul and R. Priti, "Discrete wavelet transform based multiple watermarking scheme," *Proc. IEEE Conference on Convergent Technologies for the Asia-Pacific*, India, 2003.

[21] O. G. Pla, E. T. Lin, and E. J. Delp, "A Wavelet Watermarking Algorithm Based on a Tree Structure," *proc. International Conference on Security, Steganography, and Watermarking of Multimedia Contents*, vol. 5306, pp. 571-580, San Jose, Jan. 2004.

[22] S. Gilani and A. Skodras,"Watermarking by multiresolution Hadamard transform," *proc. Electronic Imaging and Visual Arts*, pp. 73- 77, Florence, Italy, 2001.

[23] E. E. Abdallah, A. Ben Hamza, and P. Bhattacharya, "An improved image watermarking

scheme using fast Hadamard and discrete wavelet transforms," *Journal of Electronic Imaging*, 2007.

[24] E. E. Abdallah, A. Ben Hamza, and P. Bhattacharya, "A robust block-based image watermarking scheme using fast Hadamard transform and singular value decomposition," *Proc. Interntional Conference on Pattern Recognition*, vol. 3, pp. 673-676, 2006.

[25] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure Spread Spectrum Watermarking of Images, Audio and Video," *Proc. IEEE International Conf. on Image Processing*, pp. 243-246, Switzerland, September 1996.

[26] R. Liu and T. Tan, "An SVD-based watermarking scheme for protecting rightful ownership," *IEEE Trans. Multimedia*, vol. 4, no. 1, pp. 121- 128, 2002.

[27] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Data hiding for multimedia personalization, interaction, and protection," *Proceeding of the IEEE*, vol. 86, no. 6, pp. 1064-1087, 1998.

[28] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When seeing isn't believing," *IEEE Signal Processing Magazine*, vol. 21, no. 2, pp. 40-49, 2004.

[29] M. Aquilera and A. Ben Hamza, "An Information-Theoretic Approach to Georegistration of Digital Elevation Maps," *The 3rd Canadian Conference on Computer and Robot Vision*, 2006

[30] H. Gou and M. Wu, "Fingerprinting Digital Elevation Maps," *SPIE Conference on Security, Watermarking and Steganography*, Jan.2006.

[31] S. Takahashi, T. Ikeda, Y. Shinagawa, T. L. Kunii, and M. Ueda, "Algorithms for extracting correct critical points and constructing topological graphs from discrete geographical elevation data," *Computer Graphics Forum*, 14(3): 181-192, 1995.

[32] A. N. Netravali and B. G. Haskell, "Digital Pictures: Representation, Compression, and Standards," *Plenum Press, New York*, 1995.

[33] A. J. Menezes , P.C van Oorschot, and S.A. Vanstone, *Handbook of Applied Cryptography,* CRC Press, 1996.

[34] R. Venkatesan, S. M. Koon, M.H. Jakubowski, and P. Moulin, "Robust image hashing," *Proc. IEEE Int. Conf. On Image Processing,* Canada, 2000.

[35] A. K. Jain, "Fundamentals of Digital Image Processing," *V ed. Upper Saddle River, NJ: Prentice-Hall,* 2000.

[36] V. Monga and B. L. Evans, "Perceptual Image Hashing Via Feature Points: Performance Evaluation and Tradeoffs," *IEEE Trans. Image Processing,* vol. 15, no. 11, pp. 3452-3465, 2006.

[37] V. Monga and B. L. Evans, "Robust perceptual image hashing using feature points," *in Proc. IEEE Int. Conf. Image Processing,* Singapore, Oct. 2004.

[38] A. Meixner and A. Uhl, "Analysis of a wavelet based robust hash algorithm," *Proc. Security, Steganography Watermarking of Multimedia Contents VI,* vol. 5306, San Jose, CA, 2004.

[39] A. Swaminathan, Y. Mao and M. Wu, " Robust and Secure Image Hashing," *IEEE Tran. on Information Forensics and Security,* vol. 1, no. 2, pp. 215-230, 2006.

[40] Z. Karni and C. Gotsman, "Spectral compression of mesh geometry," *Pro. SIGGRAPH 2000,* pp. 279-286, ACM, 2000.

[41] B. Lamiroy and P. Gros, "Rapid Object Indexing and Recognition Using Enhanced Geometric Hashing," *Proc. Euro. Conf. Comput. Vision* pp. 59-70, 1996.

[42] H. Wu, Y. Cheung, "Public Authentication of 3D Mesh Models," *Proc. IEEE / WIC / ACM Int. Conf. on Web Intelligence,* pp. 940-946, Hong Kong, 2006.

[43] A. O Hero, B. Ma, O. Michel and J. Gorman, "Applications of entropic spanning graphs," *Proc. IEEE Signal Processing Magazine,* vol. 19, no. 5, pp. 85-95, 2002.

[44] P. Hall and S. C.Morton, "On the estimation of entropy," *Ann. Inst. Statist. Math.,* vol. 45, pp. 6988, 1993.

[45] J. Beardwood, J. H. Halton, and J. M. Hammersley, "The shortest path through many points," *Proc. Cambridge Philosophical Society,* vol. 55, pp. 299327, 1959.

[46] G. Karypis and V. Kumar, "MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fillreducing orderings of sparse matrices," *Version 4.0, Univ. Minnesota, Dept. Computer Sci.,* 1998.

[47] G. L. Miller, S.H. Teng, W. Thurston, And S. A. Vavasis. "Finite element meshes and geometric seperators," 1994.

[48] G. L. Miller, S.H. Teng, W. Thurston, And S. A. Vavasis. "Automatic mesh partitionaing," *IMA Volumes in Mathematics and its Applications,* 1993.

[49] L. Weng and B. Preneel, "On secure image hashing by higher-order statistics," *Proc. IEEE Int. Conf. On Signal Processing and Communications,* UAE, 2007.