Data Encryption and Hashing Schemes for Multimedia Protection

Kamran Khoshnasib Fallah

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

April 2012

CONCORDIA UNIVERSITY
School of Graduate Studies

This is to certify that the thesis prepared

By: Kamran Khoshnasib Fallah

Entitled: Data Encryption and Hashing Schemes for Multimedia Protection

and submitted in partial fulfillment of the requirements for the degree of

M.A.Sc. (Information Systems Security)

complies with the regulations of the University and meets the accepted standards with respect to originality and quality.

Signed by the final examining committee:

Dr. A. Awasthi Chair

Dr. J. Bentahar Examiner

Dr. K. Schmitt Examiner

Dr. A. Ben Hamza Supervisor

Approved by ____________________
Chair of Department or Graduate Program Director

____
Dean of Faculty

Date April 5, 2012

Abstract

Data Encryption and Hashing Schemes for Multimedia Protection

Kamran Khoshnasib Fallah

There are millions of people using social networking sites like Facebook, Google+, and Youtube every single day across the entire world for sharing photos and other digital media. Unfortunately, sometimes people publish content that does not belong to them. As a result, there is an increasing demand for quality software capable of providing maximum protection for copyrighted material. In addition, confidential content such as medical images and patient records require high level of security so that they can be protected from unintended disclosure, when transferred over the Internet. On the other hand, decreasing the size of an image without significant loss in quality is always highly desirable. Hence, the need for efficient compression algorithms.

This thesis introduces a robust method for image compression in the shearlet domain. Motivated by the outperformance of the Discrete Shearlet Transform (DST) compared to the Discrete Wavelet Transform (DWT) in encoding the directional information in images, we propose a DST-based compression algorithm that provides not only a better quality in terms of image approximation and compression ratio, but also increases the security of images via the Advanced Encryption Standard. Experimental results on a slew of medical images illustrate an improved performance in image quality of the proposed approximation approach in comparison to DWT, and also demonstrate its robustness against a variety of tests, including randomness, entropy, key sensitivity, and input sensitivity. We also present a 3D mesh hashing technique using spectral graph theory. The main idea is to partition a 3D model into sub-meshes, followed by the generation of the Laplace-Beltrami matrix of each sub-mesh, and the application of eigen-decomposition. This, in turn, is followed by the hashing of each sub-mesh using Tsallis entropy. The experimental results using a benchmark 3D models demonstrate the effectiveness of the proposed hashing scheme.

Table of Contents

Chapter

Chapter

Chapter

Chapter

List of Tables

LIST OF FIGURES

List of Acronyms

DFT	Discrete Fourier Transform
DWT	Discrete Wavelet Transform
CST	Continuous Shearlet Transform
DST	Discrete Shearlet Transform
DES	Data Encryption Standard
AES	Advanced Encryption Standard
STFT	Short Time Fourier Transform
VOD	Video On Demand
LAN	Local Area Network
SSL	Secure Socket layer
OPCA	Oriented Principal Component Analysis
PSNR	Peak Signal-to-Noise Ratio
MSE	Mean Square Error

Chapter 1

Introduction

1.1 Framework and Motivation

Nowadays, massive amounts of information can be stored in storage systems, using a wide variety of formats and methods. This data, which includes visual data like medical images, and non-visual data like patient records is a private and confidential property of its owners. There is, therefore, an increasing demand for data security, i.e. privacy, integrity and authentication [1]. Techniques like watermarking and image encryption are used to provide authentication, confidentiality, and privacy to images and other types of visual data like video and 3D objects. In this chapter, we introduce some applications for image encryption techniques and go on to discuss different cryptography methods for security.

1.2 Applications of Image Encryption Techniques

The following are some well-known usages of image security methods [1].

Video Conferencing

Video conferencing applications are an integral part of business and personal communication today. Software tools, such as Skype, Oovoo, Webex and Google video have made it possible for ordinary people to communicate easily and seamlessly. The information contained in these video streams can be sometimes priceless; for example, when companies discuss their new products on video conferences with their partners. Therefore, strong security protocols are required to protect the confidentiality of participants and the data stream.

Telemedicine

In the medical industry, it is essential for doctors to be able to access patient information. The storage of such information, due to sheer volume, often requires the use of a large number of storage devices. This problem can be solved through the establishment of a distributed database, through which patient data can be made available to doctors on demand, following authentication and authorization.

Surveillance

At airports, railway stations, and other public places, surveillance cameras record the activities of people to keep track of illegal activity. This data is always transferred to a central database. Transfer and storage of this data should be secure, so that individual privacy can be protected.

Video on Demand

Video on Demand (VOD) is a paid service, through which a client can stream his/her favorite movies. The provider of such a service needs a mechanism to protect the media stream from unauthorized users. Moreover, an authorized user may record the movie and sell it in the form of a DVD or CD, resulting in copyright violation. Hence, it is crucial for movie companies to find a secure method to prevent unauthorized access to/publication of the media, while ensuring that the customer continues to retain easy access to the stream.

DVD

DVD is a storage media, designed to overcome the storage limit of the CD. To protect a movie which is recorded in a DVD, the trusted hardware technology is used, meaning that a DVD can be played only on the hardware that is licensed by the DVD consortium. This protection is applied by an encryption algorithm. As a result, only the movie recorder and player hardware, which know both the encryption algorithm and the encryption key, have the ability to play the movie.

Pay-TV News

Pay-TV is a type of TV channel that a client pays to watch for a specific duration. Access to the service can be continually renewed through the payment of a fee. In order to implement security, Pay-TV encrypts the channel stream, using different encryption keys.

1.3 Encryption requirements

There are several important properties to design a good encryption algorithm.

1.3.1 Level of Security

Different levels of security are needed in different applications. For example, DVD and Pay-TV, which exist for the purpose of profit and revenue, require a higher level than video surveillance cameras, which basically exist to provide security [2].

1.3.2 Speed

Images and multimedia, compared to text, constitute a major part of internet traffic. Thus, image and multimedia encryption algorithms should be designed with more emphasis on execution speed. On the other hand, in applications that are used for sending patients's documents to a health center over the Internet, speed might not be an important issue [1].

1.3.3 Low Resource Usage

A good encryption algorithm uses low resources and CPU cycle. Despite the fact that computers and portable devices, nowadays, have faster CPU and more RAM than ever before, an efficient algorithm can make the difference between success and failure.

1.3.4 Ease of Implementation

A good encryption algorithm must be efficient and easy to implement, at the software as well as the hardware level [2].

1.4 Cryptographic Primitives

In [2], cryptographic functions or primitives are categorized as follows:

- Unkey primitives
  - Arbitrary length hash functions
  - One way permutation
  - Random sequences

- Symmetric key primitives
  - Symmetric key cipher
    - Block cipher
    - Stream cipher
  - MAC functions
  - Signatures
  - Pseudorandom sequence
  - Identification primitives
- Public key primitives
  - Public key ciphers
  - Signatures
  - Identification primitives

In this thesis, we examine the block cipher symmetric key primitive, which is explained in the following section.

1.4.1 Block Cipher Cryptography

Using block cipher cryptography, both the sender and the receiver use the same secret key to encrypt and decrypt data (Fig. 1.1). It is called block cipher, because a block of data (usually 64, 128 or 256 bits of data) is encrypted all together via a k-bit key (k usually is 56, 64, 128, 192 or 256 bits), and then sent to the receiver. At the receiver's end, the same key is used to decrypt the encrypted data. A major advantage of block ciphers is their extremely fast encryption/decryption speed, compared to other methods both in hardware and software implementation.

Some well-known block ciphers are:

- **MARS**: This encryption, introduced by IBM, encrypts a 128-bit block of data with a 128 to 448-bit key (in 32-bit increments).
- **XOR cipher**: In this method, a block of n bits of data is XORed with an n-bit key, to create an n-bit encrypted data. If n is large (e.g. 2048), it will be infeasible to break this encryption.

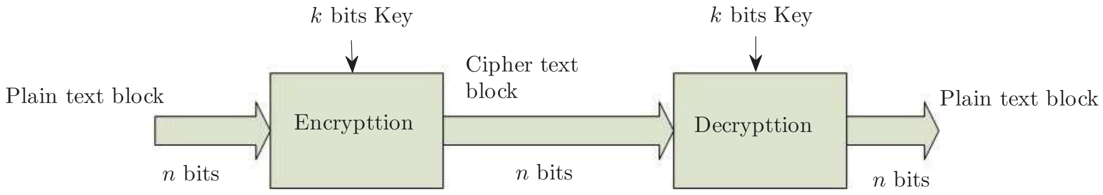


FIGURE 1.1: Block cipher encryption/decryption.

- **DES**: In this cipher, a block of 64 bits of data is encrypted with 56-bit key to a 64 bits cipher.
- **Triple-DES**: This cipher is created by three cascaded normal DES block ciphers, and comes with the advantages of a 168-bit key. However, the speed of cipher/decipher process is low.
- **AES**: In this encryption, 128 bits of data are encrypted with a 128,192, or 256-bit key to a 128-bit cipher text.
- **Chaotic maps**: In this method, chaotic map generators are used to create a random sequence of numbers that are employed to encrypt the data.
- **RC6**: This method has a block size of 128 bits and supports key sizes of 128, 192 and 256 bits.

1.5 Image Compression

Image or video compression is a process that is used to decrease the amount of data contained in representing an image or video, and to save the storage or transmission bandwidth. In compression, we decrease quality to the minimum desirable level. During the reconstruction phase, the image or video are recovered. Fig. 1.2 illustrates the process of compression, storage/transmission of images and videos, and also, the process of loading/receiving and reconstruction.

The quality of the reconstructed image depends on the application. In some applications, like movies, we do not need to recover the exact original video stream. Therefore, we can use lossy compression methods. In lossy compression, some information from the original image or video is ignored to gain more compression ratio. During the reconstruction phase, we can recover an approximation of the original media. In lossless methods, the compression algorithm never ignores any part of the data. As a result, the reconstructed image

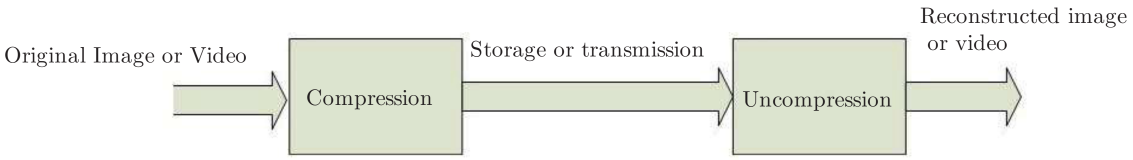


FIGURE 1.2: Image/video compress/uncompress block diagram.

or video exactly matches the original. However, in lossless methods, the compression ratio is not as good as in lossy methods [3]. Some of the lossless image compression methods are:

- Prediction based methods [4]
- DPCM and Predictive Coding
- Entropy encoding

Some lossy compression methods for image compression are:

- Fractal compression [5]
- Transform coding
- Chroma subsampling

The transform coding is the most commonly used lossy compression method. It involves the application of a Fourier-related transform such as the Discrete Fourier Transform (DFT) or the discrete wavelet transform, followed by quantization and entropy coding. We examine this idea in the next chapter. However, instead of the discrete wavelet Transform, we employ the discrete shearlet transform (DST) followed by a novel lossy compression method to compress the images.

1.6 Wavelet Transform for Image Coding

In this section, we introduce the wavelet transform that is used in image compression algorithms. Wavelet-based techniques have demonstrated the ability to provide high-coding efficiency as well as spatial and quality scalability features [3]. First, we introduce Short Time Fourier Transform (STFT), because it helps us understand DWT better.

1.6.1 STFT and DWT

Wavelets were introduced more than two decades ago to overcome some of the weaknesses of the Fourier transform. The wavelet transform is more suitable for transitional properties of signals, while the Fourier transform is mostly applicable to steady behaviors. To improve the Fourier transform to model the transient characteristics of a signal, STFT was designed. Let $f : \mathbb{R} \to \mathbb{R}$ be a real function, STFT is defined as

$$F(\omega, \tau) = \int_{-\infty}^{+\infty} f(t) w(t - \tau) e^{-j\omega t} dt \tag{1.1}$$

where $w(t)$ is called the time domain windowing function. There are numerous functions that can be used as $w(t)$, the simplest being a rectangular window that has a unit value over a time interval and has zero elsewhere.

$$w(t) = \begin{cases} 1 & \tau \leq t \leq \tau + t_1, \\ 0 & \text{otherwise.} \end{cases} \tag{1.2}$$

where $t_1 \in \mathbb{R}^+$ and $\tau \in \mathbb{R}$. Therefore, the STFT maps a one-dimensional function $f(t)$ to a two-dimensional function of time moment (τ) and frequency moment (ω) which is called $F(\omega, \tau)$. In a similar way, wavelet maps a one dimensional function to a 2D domain of a and τ, where a and τ denote dilation (scaling) and translation in time, respectively. The continuous wavelet transform of a function $f(t)$ with respect to a wavelet $\psi(t)$ is defined as [3]

$$W(a, \tau) = \int_{-\infty}^{+\infty} f(t) |a|^{-1/2} \psi^* [(t - \tau) a^{-1}] \, dt \tag{1.3}$$

Where a and τ are real variables, and * denotes the complex conjugation operator. The function, $\psi(t)$, is called the mother wavelet function and it must satisfy two conditions

- ψ should integrate to zero

$$\int_{-\infty}^{+\infty} \psi(t) \, dt = 0 \tag{1.4}$$

- ψ should have finite energy

$$\int_{-\infty}^{+\infty} |\psi(t)|^2 \, dt < +\infty \tag{1.5}$$

With these two conditions, the wavelet transform can be written as

$$W(a, \tau) = \int_{-\infty}^{+\infty} f(t) \psi_{a\tau}^*(t) \, dt \tag{1.6}$$

where

$$\psi_{a\tau}(t) = |a|^{-1/2} \psi[(t - \tau) a^{-1}] \tag{1.7}$$

The inverse wavelet transform is defined as

$$f(t) = \frac{1}{C}\int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} |a|^{-2} W(a,\tau)\psi_{a\tau}(t)\, da\, d\tau \tag{1.8}$$

where

$$C = \int_{-\infty}^{+\infty} |\hat{\psi}(\omega)|^2 |\omega|^{-1} d\omega \tag{1.9}$$

and C should be constant and finite. Two well-known $\psi(t)$ functions are: Morlet wavelet

$$\psi(t) = e^{-t^2/2}.e^{j\omega_0 t} \tag{1.10}$$

with the Fourier transform of

$$\hat{\psi}(\omega) = (2\pi)^{1/2} e^{(\omega-\omega_0)^2/2} \tag{1.11}$$

and Haar wavelet

$$\psi(t) = \begin{cases} 1, & 0 \le t \le 1/2, \\ -1, & 1/2 < t \le 1, \\ 0, & \text{else.} \end{cases} \tag{1.12}$$

with the Fourier transform of

$$\hat{\psi}(\omega) = je^{-j\omega/2}\,\frac{\sin^2(\omega/4)}{\omega/4} \tag{1.13}$$

The STFT uses sinusoidal functions as its base functions which maintain the same frequency over time, but the wavelet uses the mother wavelet functions. Hence, wavelets vary in both position and frequency over time. Fig. 1.3 depicts some popular mother wavelet functions ($\psi(t)$).

1.6.2 Fourier and Wavelet Transforms

2D Fourier transform of a 2D function $f(t)$ is defined as

$$\mathcal{F}f(t) = \hat{f}(\omega) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(t)e^{-2\pi j<\omega,t>}\, dt \tag{1.14}$$

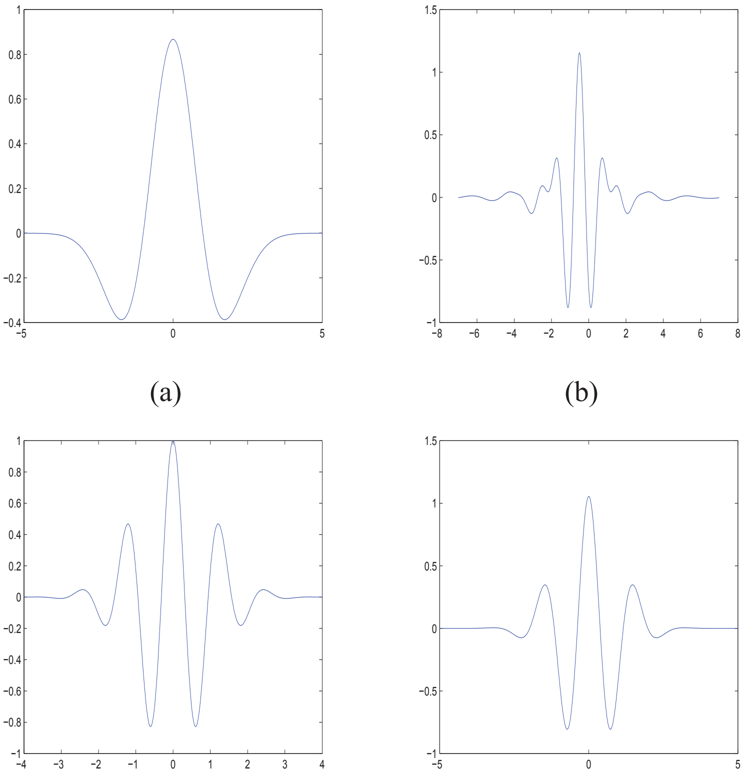
and inverse 2D Fourier transform is

$$\mathcal{F}^{-1}\hat{f}(\omega) = f(t) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} \hat{f}(\omega)e^{2\pi j<\omega,t>}\, d\omega \tag{1.15}$$

where $t \in \mathbb{R}^2$ and $\omega \in \mathbb{R}^2$.

Let $f(x,y)$ be a 2D function, we define the 2D wavelet transform as [3]

$$W(a,\tau_x,\tau_y) = \int_{-\infty}^{+\infty}\int_{-\infty}^{+\infty} f(x,y)\psi^*_{a\tau_x\tau_y}(x,y)\, dxdy \tag{1.16}$$



(a) (b)

(c) (d)

FIGURE 1.3: Some well-known wavelets: (a) Mexican hat, (b) Mayer, (c) Morlet, (d) Gaussian.

and the inverse 2D wavelet transform is

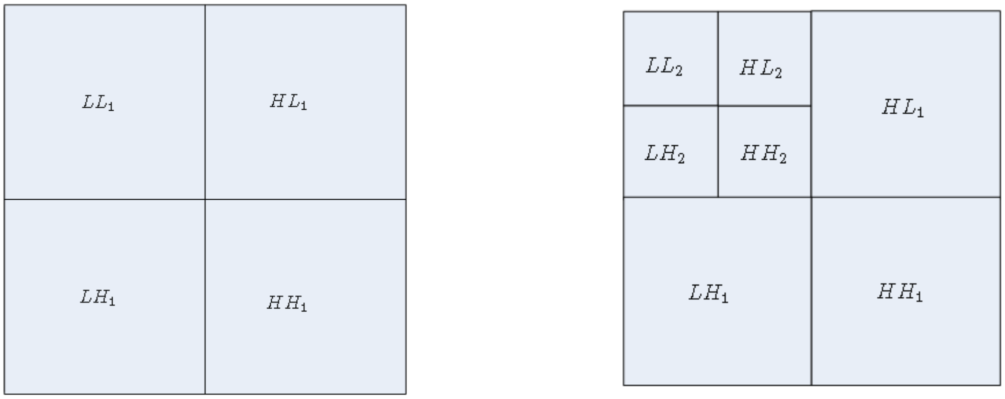
$$f(x,y) = \frac{1}{C} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} \int_{-\infty}^{+\infty} W(a, \tau_x, \tau_y) |a|^{-3} \psi_{a\tau_x\tau_y}(x,y)\, da d\tau_x d\tau_y \tag{1.17}$$

where C is defined as above and should be constant and finite, and the $\Psi(\omega)$ is the 2D Fourier transform, and $\psi(x,y)$ is the mother wavelet function.

1.7 Discrete Wavelet Transform in Image Compression

The 2D DWT can be used for the image encoding and compression. The steps of this process are: image data decomposition, quantization of the transformed coefficients, and coding of the quantized transformed coefficients. The image decomposition is mostly a lossless process which transfers image data from the spatial domain to spectral domain. The information loss takes place during the quantization step. During the coding step, we apply compression algorithms. Fig. 1.4 depicts the process. First we partition an image to four subbands and name them LL_1, LH_1, HL_1, HH_1, as in Fig. 1.4 where L means low band,

H means high band, and index 1 shows the level of decomposition. This decomposition can be repeated more than 1 level as shown in Fig. 1.4(b). The filters used to compute the DWT are generally the symmetric quadrature mirror filters (QMFs). Some well-known QMF filters are Haar, Beylkin, Coiflet, Daubechies, Symmlet, Vaidyanathan, and Battle. During quantization, each subband is quantized differently depending on its importance. In the coding step, compression algorithms, such as Embedded Zerotrees of Wavelet transforms (EZW) are used to compress data [3].



(a)

(b)

FIGURE 1.4: Wavelet decomposition: (a) first level decomposition, (b) second level decomposition.

1.8 SHEARLETS

The shearlet transform is a recent concept in image processing. It is a mathematical model that is designed to overcome some of the limitations of wavelet transform. Wavelets model the edges of an image and pointwise singularities better than the DFT. However, in edge detection, it is important to identify the location of an image's edge as well as its geometrical property. Wavelets are isotropic, and therefore unable to capture the geometric regularity along the singularities of the surface. On the other side, the shearlet transform can model the geometric property of an image better due to its anisotropic nature.

1.8.1 Continuous Shearlet Transform (CST)

Let $v : \mathbb{R} \to \mathbb{R}$ be a function defined as

$$v(x) = \begin{cases} 0, & x < 0, \\ 35x^4 - 84x^5 + 70x^6 - 20x^7, & 0 \leq x \leq 1, \\ 1, & x > 1. \end{cases} \tag{1.18}$$

another candidate for $v(x)$ is

$$\tilde{v}(x) = \begin{cases} 0, & x < 0, \\ 2x^2, & 0 \leq x \leq 1/2, \\ 1 - 2(1-x)^2, & 1/2 \leq x \leq 1, \\ 1, & x > 1. \end{cases} \tag{1.19}$$

and let $b : \mathbb{R} \to \mathbb{R}$ be

$$b(\omega) = \begin{cases} \sin((\pi/2)v(|\omega| - 1)), & 1 \leq |\omega| \leq 2, \\ \cos((\pi/2)v(|\omega|/2 - 1)), & 2 < |\omega| \leq 4, \\ 0, & \text{else.} \end{cases} \tag{1.20}$$

The function b is symmetric, positive, real function, and $\text{supp}(b) = [-4, -1] \cup [1, 4]$. Fig. 1.5 shows the $v(x)$ and $b(\omega)$ functions. We define ψ_1 via its Fourier transform as

$$\hat{\psi}_1(\omega) = \sqrt{b^2(\omega) + b^2(2\omega)} \tag{1.21}$$

The second function ψ_2 is defined again via its Fourier transform as

$$\hat{\psi}_2(\omega) = \begin{cases} \sqrt{v(1+\omega)}, & \omega \leq 0, \\ \sqrt{v(1-\omega)}, & \omega > 0. \end{cases} \tag{1.22}$$

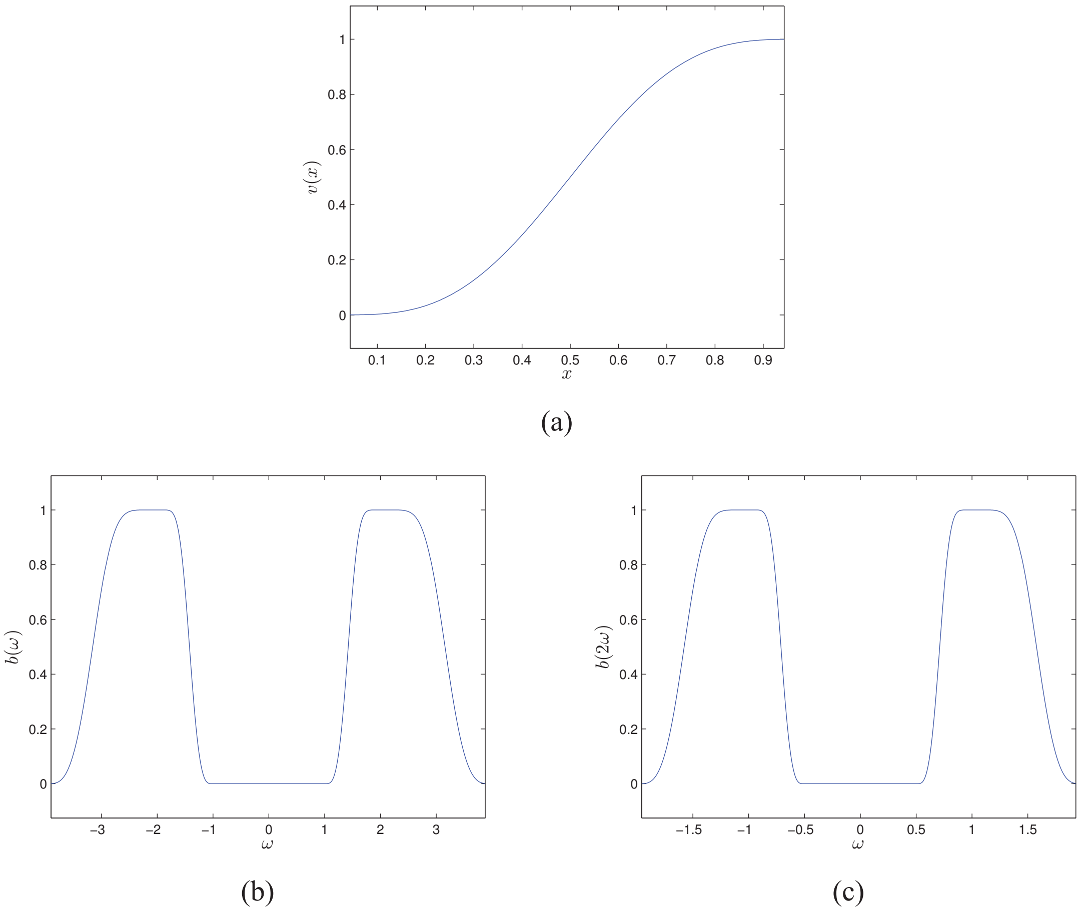
Fig. 1.6 shows these functions.

Let $a \in \mathbb{R}^+$, $s \in \mathbb{R}$, and $x,\ t,\ \omega \in \mathbb{R}^2$. The 2D continuous shearlet transform of a two dimensional function $f(x) = f(x_1, x_2)$ is defined as:

$$\mathcal{SH}_f(a, s, t) =< f, \psi_{a,s,t} > \tag{1.23}$$

and multiplication is defined as

$$(a, s, t).(a', s', t') = (aa', s + s'a^{1/2}, M_{a,s}t') \tag{1.24}$$

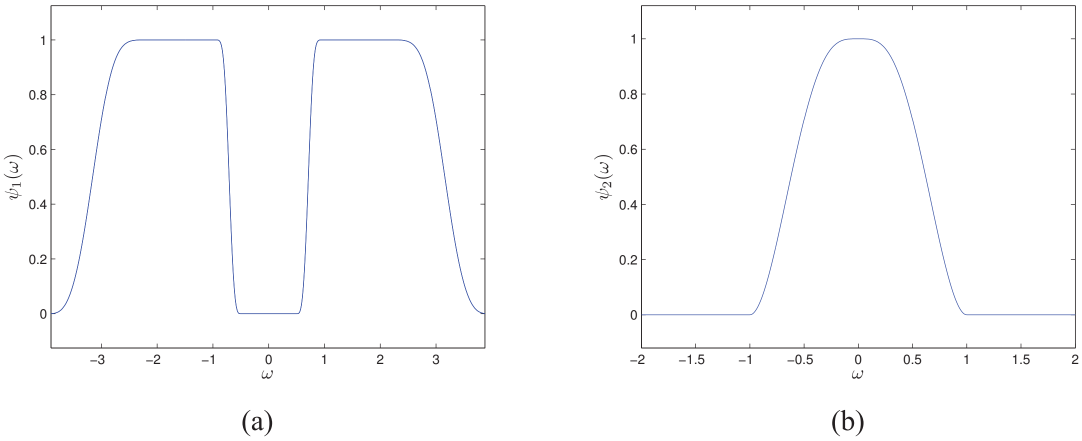


(a)

(b)

(c)

FIGURE 1.5: (a) $v(x)$ function, (b) $b(\omega)$ function, (c) $b(2\omega)$ function.



(a)

(b)

FIGURE 1.6: (a) $\hat{\psi}_1(\omega)$ function, (b) $\hat{\psi}_2(\omega)$ function.

where

$$\psi_{a,s,t} =| \det M_{a,s} |^{-0.5} \psi(M_{a,s}^{-1}x - t) \tag{1.25}$$

and

$$M_{a,s} = \begin{bmatrix} a & a^{1/2}s \\ 0 & a^{1/2} \end{bmatrix} = B_s A_a \tag{1.26}$$

and

$$B_s = \begin{bmatrix} 1 & s \\ 0 & 1 \end{bmatrix} \tag{1.27}$$

and

$$A_a = \begin{bmatrix} a & 0 \\ 0 & \sqrt{a} \end{bmatrix}. \tag{1.28}$$

It means the matrix $M_{a,s}$ is the product of anisotropic dilation (scaling) matrix A_a and sharing matrix B_s. For $\omega_1 \neq 0$, the value of $\hat{\psi}$ at ω is chosen to be of the form

$$\hat{\psi}(\omega) = \hat{\psi}(\omega_1, \omega_2) = \hat{\psi}_1(\omega_1)\hat{\psi}_2(\omega_2/\omega_1) \tag{1.29}$$

so the shearlet transform can be written as

$$\mathcal{SH}_f(a,s,t) = a^{\frac{3}{4}}\mathcal{F}^{-1}\left(\hat{f}(\omega_1,\omega_2)\hat{\psi}_1(a\omega_1)\hat{\psi}_2\left(a^{-1/2}(\frac{\omega_2}{\omega_1} + s)\right)\right)(t) \tag{1.30}$$

Where $\mathcal{F}^{-1}$ denotes the inverse Fourier transform in 2D. The shearlet transform has support in $\{[-2/a, -1/(2a)] \cup [1/(2a), 2/a], |(\omega_2/\omega_1) + s| < \sqrt{a}\}$.

1.8.2 Discrete Shearlet Transform (DST)

By sampling the continuous shearlet transform on an appropriate discrete set, a frame or even Parseval frame for $L^2(\mathbb{R})$ can be archived. To obtain the discrete transform, the three parameters should be sampled as: $a_j = 2^{-j}$ $(j \in \mathbb{Z})$, $s = -l$ $(l \in \mathbb{Z})$, so that a collection of the matrices $M_{2^{-j},-l}$ can be obtained. In addition

$$M^{-1}_{2^{-j},-l} = M_{2^j,l} = \begin{bmatrix} 2^j & l2^{\frac{j}{2}} \\ 0 & 2^{\frac{j}{2}} \end{bmatrix} = B_0^l A_0^j \tag{1.31}$$

where

$$A_0 = \begin{bmatrix} 2 & 0 \\ 0 & \sqrt{2} \end{bmatrix} \tag{1.32}$$

and

$$B_0 = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}. \tag{1.33}$$

Using the continuous $\psi_{a,s,t}$ equation, we can obtain

$$\psi_{j,l,k} =\mid detA_0 \mid^{\frac{j}{2}} \psi(B_0^l A_0^j x - k) \tag{1.34}$$

for $j, l \in \mathbb{Z},\ k \in \mathbb{Z}^2$. The discrete shearlet is able to deal with multi-dimensional functions as its continuous counterpart [7].

1.8.3 Applications of Shearlet Transform

There are various fields where the shearlet is used. Generally, all applications of the DWT can employ the DST too. The following list shows some popular usages:

- Image approximation
- Image denoising
- Edge analysis and identification

In this thesis, we employ the shearlet transform in image approximation and compression.

1.9 Entropy

Entropy is the statistical measure of uncertainty in a random variable. The Shannon entropy of a gray scale image is defined as [14]

$$H = -\sum_{i=1}^{K} p_i \log_2 p_i \tag{1.35}$$

where p_i is the probability of color with value i, and K is the total number of different colors ($K = 256$ for a gray scale image). When a gray scale message is ideally encrypted, the entropy is eight. An entropy less than eight means some information can be predicted or there is information leakage.

1.10 Thesis Overview and Contributions

The organization of this thesis is as follows

- ❑ The first Chapter contains a brief review of essential concepts and definitions which we refer to throughout the thesis, and presents a short summary of material relevant to image encryption, compression, wavelets and shearlets.

- ❑ In Chapter 2, we present a novel shearlet-based image compression method for efficient image transmission over the Internet, combined with the Advanced Encryption Standard (AES) for image security. We use the DST to approximate an image and use a novel compression method to reduce its size. Then AES is used to encrypt the compressed data. The main motivation for this contribution is saving transmission bandwidth and providing security against unauthorized disclosure for medical images when they are send over the internet (see Telemedicine subsection). Simulation results indicate that the proposed method provides better image quality than DWT, improves compression ratio, and increases the security of transmitted images over the Internet.

- ❑ In Chapter 3, we present a hashing technique for 3D models using spectral graph theory and entropic spanning trees [39]. The main idea is to partition a 3D triangle mesh into an ensemble of sub-meshes, then apply eigen-decomposition to the Laplace-Beltrami matrix of each sub-mesh, followed by the computation of the hash value of each sub-mesh. The main motivation for this contribution is security of 3D mesh objects against unauthorized copy and usage. Obviously, it is very important for game companies to have trustable methods to protect their 3D mesh objects against unauthorized usage. This hash value is defined in terms of spectral coefficients and Tsallis entropy estimate. 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, tessellation, welder, scaling, rotation, as well as combinations of these attacks.

- ❑ In the Conclusions Chapter, we summarize the contributions of this thesis, and propose several future research directions that are directly or indirectly related to the ideas developed therein.

CHAPTER 2

IMAGE COMPRESSION AND ENCRYPTION USING SHEARLETS

This chapter presents a novel shearlet-based image compression method for efficient image transmission over the Internet, combined with Advanced Encryption Standard (AES) for image security. The discrete shearlet transform (DST) is a mathematical concept that was recently introduced to overcome some of the limitations of the discrete wavelet transform (DWT) in image processing applications. Because of its isotropic property, DWT fails to capture the geometric regularity along the singularities of the surfaces. On the other hand, the quality of the discontinuous images reconstructed via DST is higher than the quality of reconstructed images obtained using the DWT method. AES is a widely known symmetric key encryption/decryption method that is employed in data security applications. The simulation results indicate that the proposed method provides a better image quality than DWT, improves compression ratio, and increases the security of images transmitted over the Internet.

2.1 INTRODUCTION

Security and privacy are important issues in all aspects of life. In recent years, with the increasing demand for Internet-based applications and services, security has become an extremely important concern. In spite of this, data is mostly transferred unencrypted. In medical applications, image encryption is of paramount importance. Medical images are confidential records and documents, and should be protected using the highest levels of security. Transmission channel coding (channel encrypting) and data encrypting are two well-known methods. In channel coding, IP security (IPSec), Secure Socket Layer (SSL) or similar methods are used to create a secure path to transfer clear and unencrypted data. In data coding, encrypted data is sent through an insecure channel. Data Encryption Standard (DES) and Advanced Encryption Standard (AES) are widely-used methods of data encryption [8]. Channel encryption and data encryption can be combined to improve security, but this process requires more resources and more expensive hardware.

Although the bandwidth price has decreased considerably in the past years, images and multimedia streams are the major bandwidth consumers in both Local Area Networks (LAN) and Wide Area Networks (WAN). To save the bandwidth, data should be compressed as much as possible. In the subject area of image compression, there are two widely used methods: Lossy and Lossless [15]. Using the former method, some parts of data are deleted during the compression to achieve better compression ratio. It is usually applied to multimedia signals, where the quality is not a big concern at the receiver's end. MPEG-2 Video codec for DVD compression is an example of this method. In the latter method, reconstructed data is exactly the same as the original. Some lossless examples are the LempelZiv (LZ) or DEFLATE method that are used in PKZIP, GZIP, ZLIB, and PNG. The application determines the method of compression. In the proposed method, both lossy and lossless methods are employed in different places.

The rest of this chapter is organized as follows. Section 2.2 briefly explains some earlier practices to compress and securely transfer images over the Internet. Section 2.3 introduces AES encryption briefly. Section 2.4 reviews the concept of DST. Section 2.5 describes the proposed method based on the shearlet, ZLIB, and AES for image compression and encryption. Decryption and decompression are explained in section 2.6. In section 2.7, simulation results and security analysis are provided. In section 2.8, we present our conclusions.

2.2 Overview of Wavelet-based Image Encryption and Compression

In this section we briefly explain two methods of image encryption: the wavelet transform followed by DES encryption, and chaos-based encryption.

2.2.1 Wavelet Transform and DES Encryption

As described in [8], one method for secure image transmission and compression is the Embedded Zerotree Wavelet (EZW) and Data Encryption Standard (DES) symmetric encryption/decryption. EZW is an effective coding technique for image compression. However, DES is not a secure method by today's security standards, and there are numerous ways to attack it, such as the improved Davies' attack, and the attacks discussed in [11, 10]. More secure methods should, therefore, be chosen.

2.2.2 Chaos Based Encryption

In recent years, a slew of methods were proposed to improve encryption using chaos-based algorithms [12, 13, 14]. Awad in [14] proposed four chaos based cryptosystems. Using these algorithms, a two-dimensional (2D) chaotic map is used to shuffle the pixels' locations in an image. Then, two perturbed chaotic PWLCM maps are used in the multiple substitution/permutation rounds. Although the chaotic maps can create high quality random numbers, they are more prone to different type of attacks [21].

2.3 Advanced Encryption Standard (AES)

AES is a cryptography symmetric key encryption standard that was adopted for commercial encryption usages [19, 20]. It is a new version of symmetric key encryption/decryption protocol that supersedes DES. The major weakness of DES or Triple DES (3DES) is low encryption speed, in software as well as hardware implementations. In addition, the encryption key length in DES is 56 bits, which is not enough for most applications. However, AES was designed with speed as one of the constraints. AES uses 128, 192, or 256-bit key for 128-bit block of data as shown in Fig. 2.1.

AES has different rounds of permutation/substitution for different key lengths. 10 rounds for 128-bit, 12 rounds for 196-bit, and 14 rounds for 256-bit encryption key. The details of AES encryption are beyond the scope of this thesis, and can be found in [19].

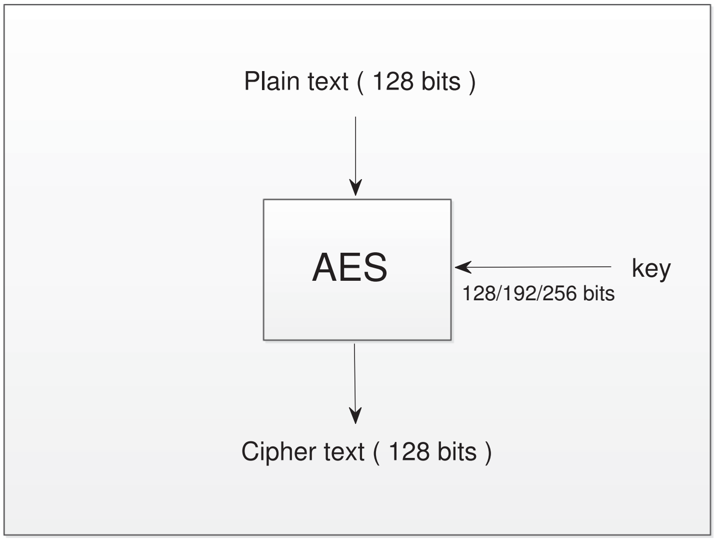


FIGURE 2.1: AES block cipher.

2.4 Shearlet Transform and Approximation

2.4.1 Motivation

Shearlet is a new concept in image processing that was designed to overcome some of the limitations of wavelets. Wavelets model the edges of images and point-wise singularities better than DFT. However, in edge detection, it is important not only to identify the location of an image edge, but also the geometrical property of the image. Wavelet is isotropic, and is therefore able to capture the geometric regularity along the singularities of surfaces [16]. Shearlet, on the other hand, is more efficient in modeling the geometrical property of an image due to its anisotropic nature. There are different ways to define Shearlet and DST [6, 16, 17].

2.4.2 Continuous Shearlet Transform

The Fourier transform reveals only the frequency attributes of an image. The wavelet transform provides a powerful insight into an image's spatial and frequency characteristics, and has two parameters: translation and scale. The basic idea underlying the definition of the continuous shearlet is the use of a 2-parameter dilation (scaling) group, that is a product of parabolic scaling matrices and shear matrices. Therefore, shearlet has three parameters: the anisotropic dilation (scaling) parameter $a > 0$, the shear parameter $s \in \mathbb{R}$, and the translation parameter $t \in \mathbb{R}^2$.

Let $x \in \mathbb{R}^2$ and $\xi \in \mathbb{R}^2$, continuous shearlet in time domain is defined as:

$$\psi_{a,s,t}(x) = a^{-3/4}\psi(D_{a,s}^{-1}(x-t)) \tag{2.1}$$

where

$$D_{a,s} = \begin{bmatrix} a & -a^{1/2}s \\ 0 & a^{1/2} \end{bmatrix} = B_s A_a \tag{2.2}$$

and

$$B_s = \begin{bmatrix} 1 & -s \\ 0 & 1 \end{bmatrix} \tag{2.3}$$

and

$$A_a = \begin{bmatrix} a & 0 \\ 0 & \sqrt{a} \end{bmatrix}. \tag{2.4}$$

In addition, the mother shearlet function ψ is defined like a tensor product in the frequency domain:

$$\hat{\psi}(\xi_1, \xi_2) = \hat{\psi}_1(\xi_1)\hat{\psi}_2(\xi_2/\xi_1) \tag{2.5}$$

where $\hat{\psi}$ is the fourier transform of the mother shearlet ψ, $\hat{\psi}_1$ is the fourier transform of a wavelet ψ_1, and $\hat{\psi}_2$ is the fourier transform of a bump function ψ_2. The continuous shearlet transform associated to this continuous shearlet is defined by:

$$\mathcal{SH}_f(a,s,t) =< f, \psi_{a,s,t} > \tag{2.6}$$

and multiplication is defined as:

$$(a,s,t).(a',s',t') = (aa', s + s'a^{1/2}, D_{a,s}t') \tag{2.7}$$

Fig. 2.2 shows the effect of parameters a and s in shearlet transform of a square. This transform can be shown as matrix coefficients of the unitary representation:

$$(\sigma(a,s,t)\psi)(x) = \psi_{a,s,t}(x) = a^{-3/4}\psi(D_{a,s}^{-1}(x-t)) \tag{2.8}$$

of the shearlet group $S = \mathbb{R}^+ \times \mathbb{R} \times \mathbb{R}^2$. The location parameter t can precisely detect the location of singularities, and the shear parameter s shows the perpendicular direction to singularity [18].

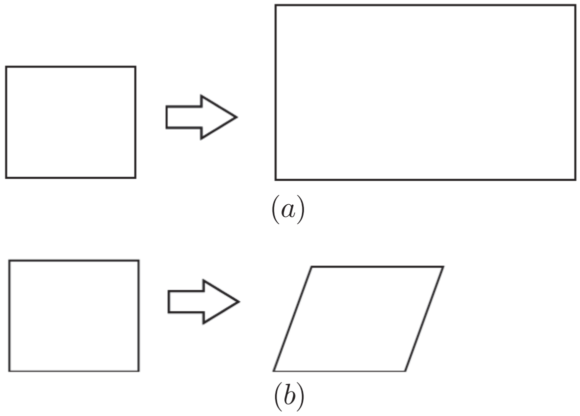


FIGURE 2.2: Geometric transformation in shearlet: (a) anisotropic dilation (scaling) parameter a, (b) shear parameter s.

2.4.3 DISCRETE SHEARLET TRANSFORM (DST)

By sampling continuous Shearlet transform on the appropriate discrete set, a frame or even Parseval frame for $L^2(\mathbb{R})$ can be achieved. To obtain the discrete transform, three parameters should be sampled as follows:

$$a_j = 2^j (j \in \mathbb{Z}), \quad s_{j,k} = ka_j^{1/2} = k2^{j/2}(k \in \mathbb{Z}), \quad \text{and} \quad t_{j,k,m} = D_{a_j,s_j,k}(m \in \mathbb{Z}^2) \tag{2.9}$$

The mother discrete shearlet that is chosen is the same fashion as in the continuous case, i.e., ψ_1 is chosen to be a discrete wavelet, and ψ_2 is chosen to be a bump function with certain weak additional properties. This system forms an optimally sparse Parseval frame in $L^2(\mathbb{R})$.

2.5 Image Compression and Encryption

In the first stage, DST approximation method is used to create the DST matrix. The proposed compression method employs this matrix to create a smaller one aimed at decreasing the size of the data. It is a lossy approximation, as a result of which some data are lost. In the second stage, ZLIB lossless algorithm is used to compress the new coefficient matrix. In the last stage, AES is used to encrypt the compressed matrix of selected coefficients. Fig. 2.3 shows the compression and encryption block diagrams. The details of this process are described in the following sub sections.

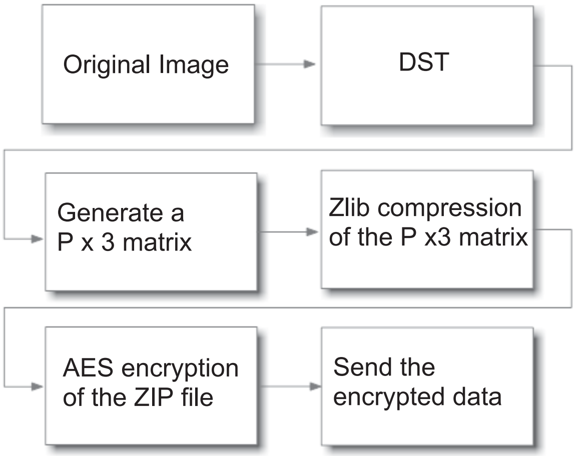

Original Image
DST
Generate a P x 3 matrix
Zlib compression of the P x3 matrix
AES encryption of the ZIP file
Send the encrypted data

FIGURE 2.3: Compression and encryption block diagram

2.5.1 DST Approximation

The proposed method uses DST to generate approximation shearlet matrix of the image. Let the original image be an $M \times N$ gray scale image, so that the image size is $M \times N$ bytes. After calculating DST, the size of the DST matrix is $4 \times M \times N$, assuming its elements are 4-byte single precision floating point numbers, that is 4 times bigger than the original image size. In the next step, DST approximation algorithm which is introduced in [16] is employed to create image an approximation matrix, which is the same size as the DST matrix.

To decrease the size of the approximation matrix, we retain only n percent of its elements having the largest absolute value, and ignore the others (e.g. write zero on the other elements). The number n indicates the percentage of the total image pixels that should be retained, and controls the precision of our proposed method. A larger value of n means a higher image quality. However it also means a higher bandwidth. The new generated matrix is denoted wc.

2.5.2 Compression

After computing wc, a new $P \times 3$ matrix is generated according to Algorithm 1.

Algorithm 1 Proposed algorithm to generate a new $P \times 3$ matrix.

```
for each non-zero element in $wc$ do
    Go to the first empty row of the $P \times 3$ matrix.
    The first element of the row corresponds to the row number of the non-zero elements in $wc$.
    The second element of the row corresponds to the column number of the non-zero elements in $wc$.
    The third element of the row corresponds to the value of the non-zero elements in $wc$ in single
    precision floating point format.
end for
```

If we retained n percent of the wc coefficients, then $P = \text{round}(\frac{n}{100}MN)$. By now, the size of data is changed from $M \times N$ to $P \times 3$. To retain the size of the new $P \times 3$ matrix less than the original $M \times N$ image, n must be selected as

$$P \times 8 \leq MN, \quad \text{or} \quad \text{round}\left(\frac{n}{100}MN\right) \times 8 \leq MN, \quad \text{or} \quad n \leq 12$$

In this calculation, each row in the $P \times 3$ matrix is considered as eight bytes of data: two bytes for each of the first and the second columns, which are integers between 1 and 32500, and four bytes for the third column that is a single precision floating point number. In real applications, $2 \leq n \leq 5$ is a good choice for images. Finally, to decrease the size of $P \times 3$ matrix furthur, we use the ZLIB lossless compression method.

2.5.3 AES Encryption

The compressed $P \times 3$ matrix is sent to AES module to be encrypted via a private key K. The receiver side uses the same K to decrypt the encrypted data. In most commercial applications, 128-bit AES is used. In more secure applications, on the other hand, 256-bit AES is a good candidate.

2.6 Image Decryption and Reconstruction

For decryption and reconstruction, the reverse process should be applied. First, the encrypted data is decrypted. Then the data are uncompressed to extract the $P \times 3$ matrix. In the next step, the matrix wc is constructed. Finally, using the DST approximation method in [16] the original image can be recovered (Fig. 2.4).

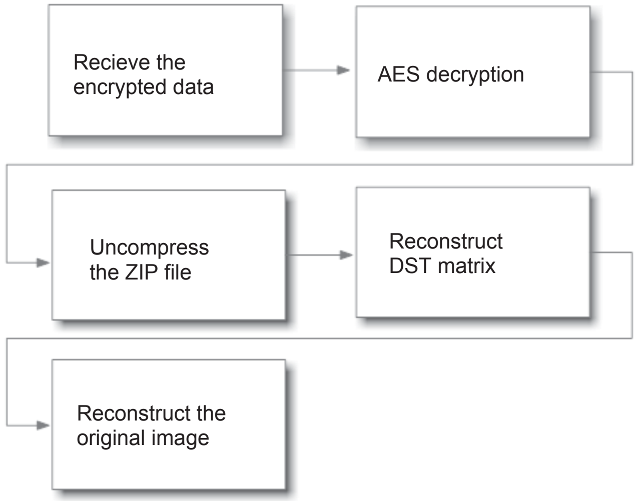
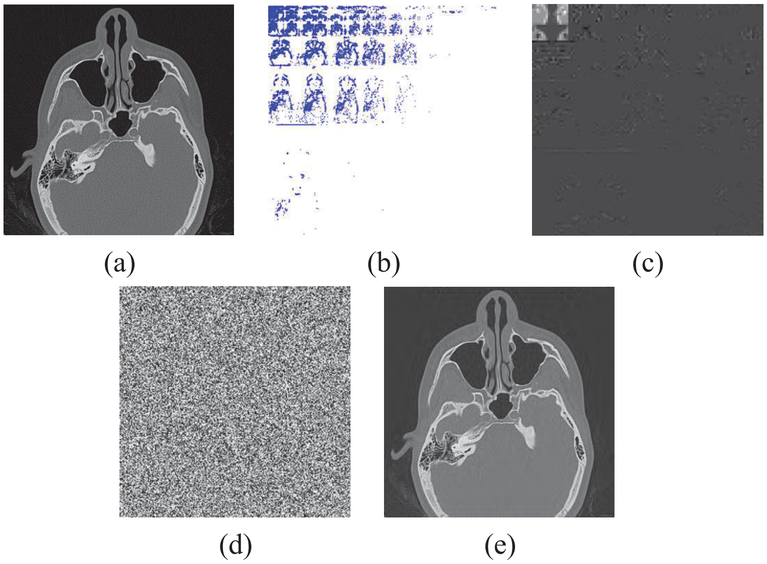


Figure 2.4: Decryption and reconstruction block diagram.

2.7 Simulation Results and Security Analysis

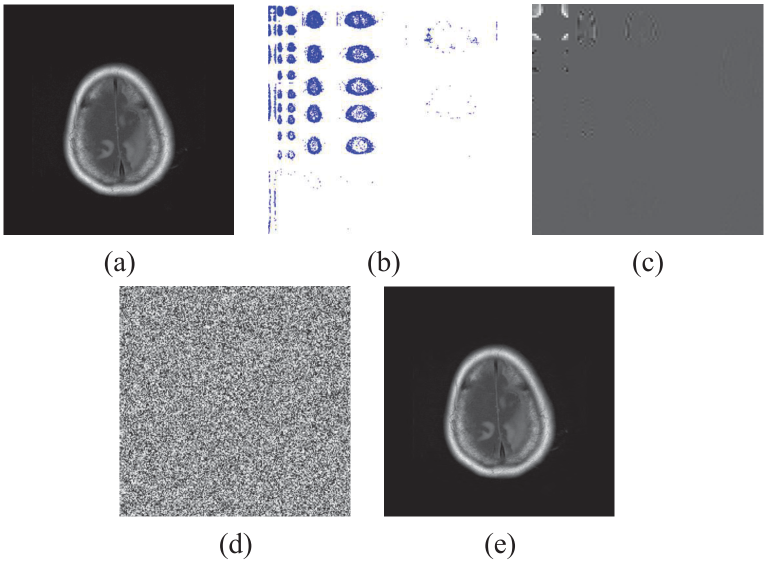
The proposed method was applied to four images, and the results are shown in the Fig. 2.5-2.8. In addition, $n = 4$ in all experiments, and 128-bit AES is used for encryption/decryption. The first image in each set is the original image. The second image is the non-zero coefficients of approximation matrix wc. The third image is the normal representation of wc. The fourth image is the zoomed view of the compressed and encrypted matrix of coefficients. The last image in each set shows the reconstructed image on the receiver's side.



(a) (b) (c)

(d) (e)

FIGURE 2.5: Dental medical image: (a) original image, (b) non-zero coefficients of the approximation matrix, (c) approximation matrix wc, (d) small matrix of coefficients after ZLIB compression and AES encryption, (e) reconstructed image.



(a) (b) (c)

(d) (e)

FIGURE 2.6: Brain medical image: (a) original image, (b) non-zero coefficients of the approximation matrix wc, (c) approximation matrix wc, (d) small matrix of coefficients after ZLIB compression and AES encryption, (e) reconstructed image.

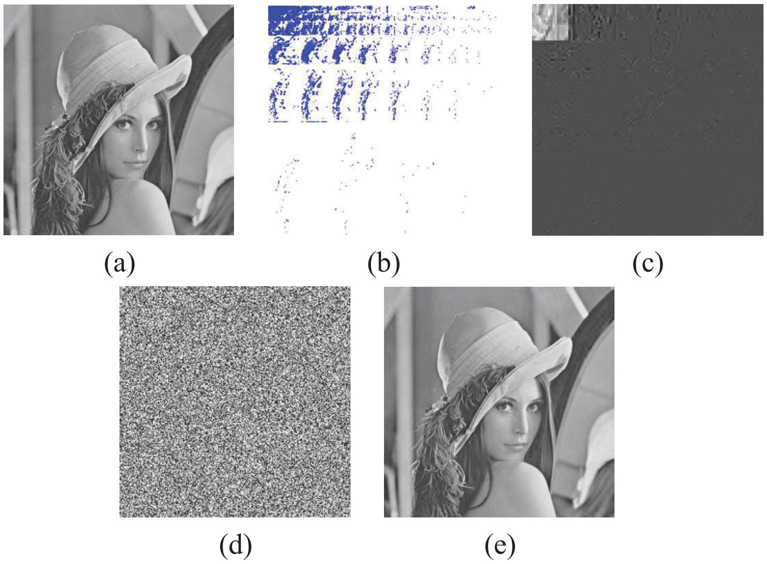


FIGURE 2.7: Lena image: (a) original image, (b) non-zero coefficients of the approximation matrix wc, (c) approximation matrix wc, (d) small matrix of coefficients after ZLIB compression and AES encryption, (e) reconstructed image.

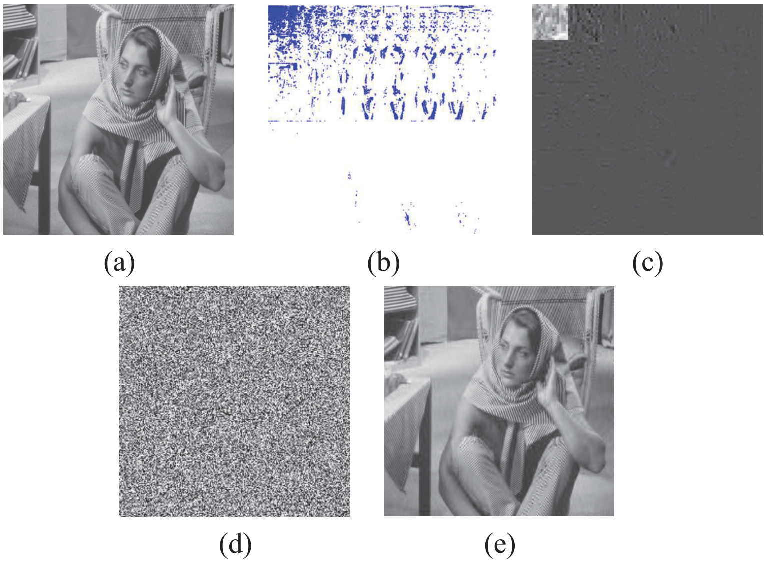


FIGURE 2.8: Barbara image: (a) original image, (b) non-zero coefficients of the approximation matrix wc, (c) approximation matrix wc, (d) small matrix of coefficients after ZLIB compression and AES encryption, (e) reconstructed image.

TABLE 2.1: PSNR for wavelet and shearlet (dB) ($n = 4$).

Image	Wavelet PSNR	Shearlet PSNR
Dental image	40.186	43.356
Brain image	54.797	59.405
Lena	29.830	34.539
Barbara	24.656	29.589

TABLE 2.2: MSE for wavelet and shearlet ($n = 4$).

Image	Wavelet MSE	Shearlet MSE
Dental image	6.230	3.002
Brain image	0.215	0.075
Lena	67.618	22.866
Barbara	222.596	71.484

2.7.1 Comparison Between Reconstructed Image Quality In Shearlet And Wavelet

To evaluate the quality of the reconstructed image the Peak Signal-to-Noise Ratio (PSNR) is used:

$$\mathrm{PSNR} = 10 \log \left(\frac{255^2 MN}{\sum_{i=1}^{M} \sum_{j=1}^{N} (P_{ij} - Q_{ij})^2} \right) dB \tag{2.10}$$

where MN is the total number of pixels, and $P = (P_{ij})$ and $Q = (Q_{ij})$ are the original and the reconstructed image. A larger PSNR means a better quality of reconstructed image.

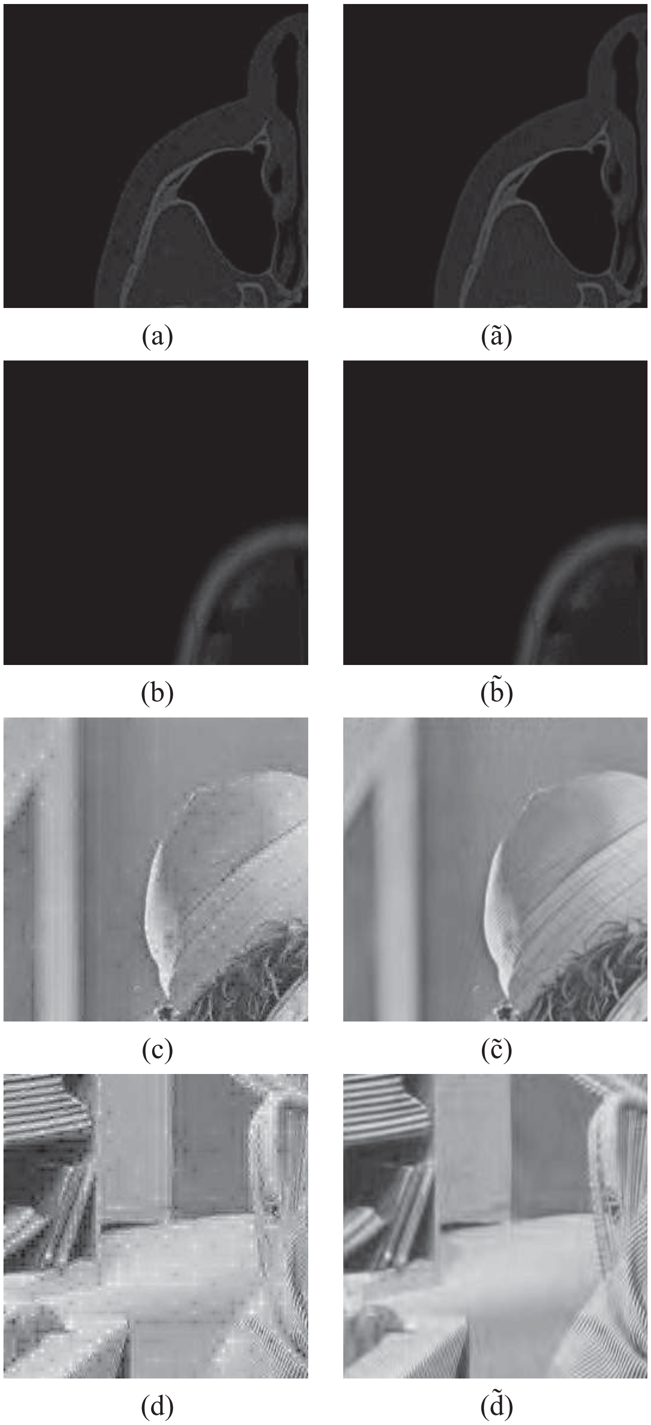
The Mean Square Error (MSE) is another widely-used measure to evaluate reconstructed image quality:

$$\mathrm{MSE} = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} (P_{ij} - Q_{ij})^2}{MN} \tag{2.11}$$

A small MSE means a smaller error in image reconstruction. The quality and error of the reconstructed images in both shearlet and wavelet methods are shown in Tables 2.1 and 2.2, which illustrates that DST approximation has better quality and less error than DWT. Fig. 2.9 shows the zoomed area in the reconstructed DWT and DST images.

2.7.2 Compression Ratio

The ratio of the encrypted image to the size of the original image is defined as compression ratio in the proposed method. As shown in Table 2.3, the proposed method provides a high compression ratio.



(a) (ã)

(b) (b̃)

(c) (c̃)

(d) (d̃)

FIGURE 2.9: Comparing wavelet approximation and shearlet approximation: (a)-(d) zoomed wavelet approximation, (ã)-(d̃) their equivalent zoomed shearlet approximations.

TABLE 2.3: Image compression ratio in the proposed method ($n = 4$).

Image	Compression ratio (Transmitted data / Original file)
Dental image	0.0935
Brain image	0.0909
Lena	0.3824
Barbara	0.3183

2.7.3 Correlation Of Adjacent Pixels

Correlation coefficient of a sequence of adjacent pixels is defined by [14]:

$$E(P) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} P_{ij}}{MN} \tag{2.12}$$

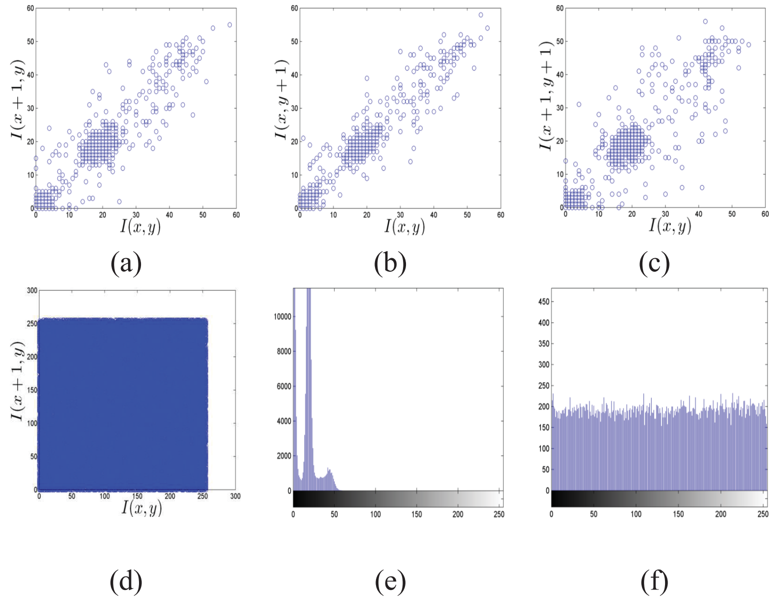
$$D(P) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} [P_{ij} - E(P_{ij})]^2}{MN} \tag{2.13}$$

$$\mathrm{cov}(P, C) = \frac{\sum_{i=1}^{M} \sum_{j=1}^{N} [P_{ij} - E(P_{ij})][C_{ij} - E(C_{ij})]^T}{MN} \tag{2.14}$$

$$\mathrm{std}(P) = \sqrt{D(P)}, \quad \mathrm{std}(C) = \sqrt{D(C)} \tag{2.15}$$

$$r_{P,C} = \frac{\mathrm{cov}(P, C)}{\mathrm{std}(P) \times \mathrm{std}(C)} \tag{2.16}$$

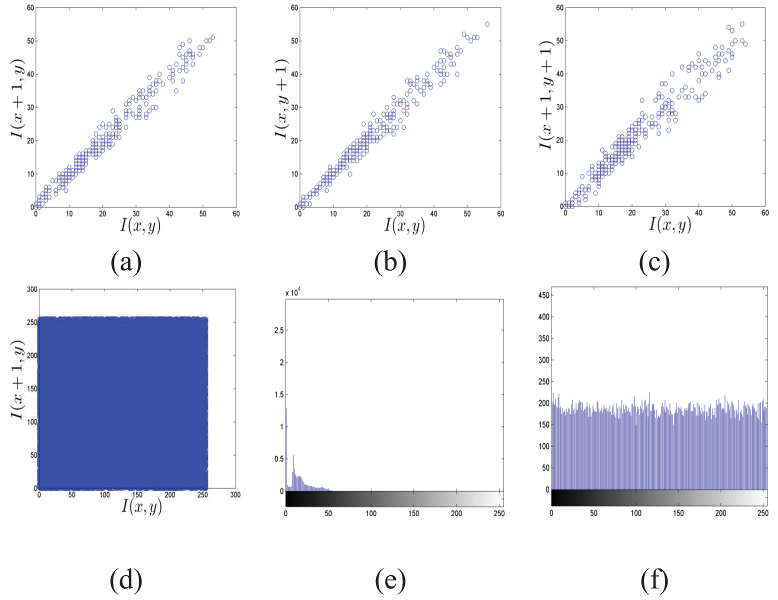
where $P = (P_{ij})$ is the original image, $C = (C_{ij})$ is the encrypted image, $\mathrm{cov}(P, C)$ is covariance between P and C, and $\mathrm{std}(P)$ and $\mathrm{std}(C)$ are their respectively standard deviations. Finally, $r_{P,C}$ is the correlation between the two matrices. Fig. 2.10-2.13 show horizontal, vertical and diagonal pixel correlation distribution in the original image and horizontal correlation in the encrypted images. The vertical and diagonal correlation of adjacent pixels in the encrypted images are similar to the horizontal. According to the table, the encrypted images have low pixel correlation, while the originals have high. To draw these diagrams, 1500 different pairs of horizontally adjacent pixels are randomly selected from the image. Next, pixel values on location $(x + 1, y)$ over the pixel values on location (x, y) are plotted. A similar process is employed for the vertical and the diagonal adjacent pixels, and the results are similar to the horizontal correlation.



(a) (b) (c)

(d) (e) (f)

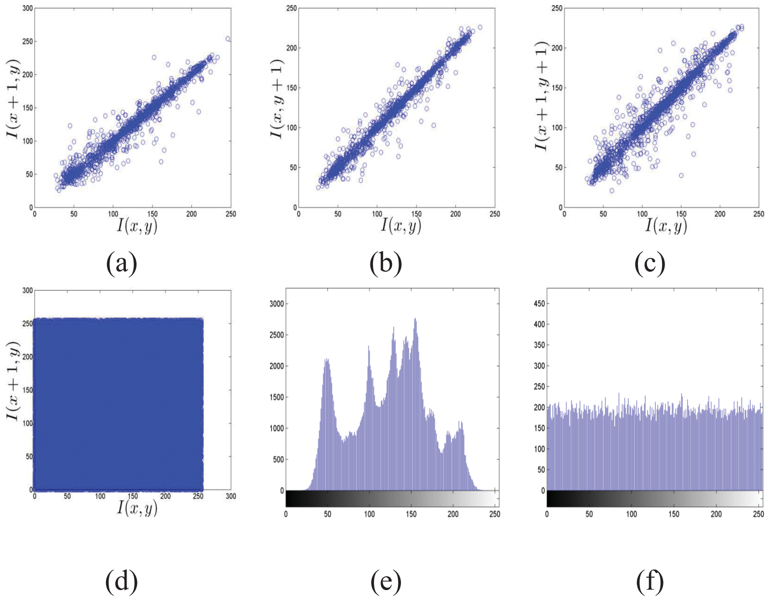
FIGURE 2.10: Dental medical image: (a) original image horizontal adjacent pixel correlation, (b) original image vertical adjacent pixel correlation, (c) original image diagonal adjacent pixel correlation, (d) encrypted image horizontal adjacent pixel correlation, (e) original image histogram, (f) encrypted image histogram.



(a) (b) (c)

(d) (e) (f)

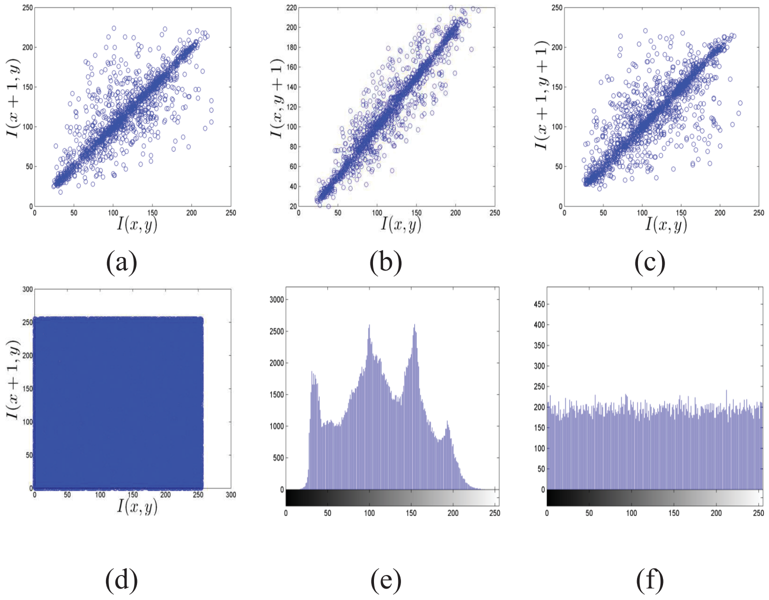
FIGURE 2.11: Brain medical image: (a) original image horizontal adjacent pixel correlation , (b) original image vertical adjacent pixel correlation, (c) original image diagonal adjacent pixel correlation, (d) encrypted image horizontal adjacent pixel correlation, (e) original image histogram, (f) encrypted image histogram.



(a) (b) (c)

(d) (e) (f)

FIGURE 2.12: Lena image: (a) original image horizontal adjacent pixel correlation, (b) original image vertical adjacent pixel correlation, (c) original image diagonal adjacent pixel correlation, (d) encrypted image horizontal adjacent pixel correlation, (e) original image histogram, (f) encrypted image histogram.



(a) (b) (c)

(d) (e) (f)

FIGURE 2.13: Barbara image: (a) original image horizontal adjacent pixel correlation, (b) original image vertical adjacent pixel correlation, (c) original image diagonal adjacent pixel correlation, (d) encrypted image horizontal adjacent pixel correlation, (e) original image histogram, (f) encrypted image histogram.

TABLE 2.4: Entropy of the original and encrypted images.

Image	Original	Encrypted
Dental image	4.3387	7.9960
Brain image	2.2238	7.9966
Lena	7.4464	7.9963
Barbara	7.4604	7.9962

2.7.4 Histogram Analysis

The image histogram shows how many pixels for each color can be found on the image. In Fig. 2.10-2.13(e)-(f), the histogram of the original image and the encrypted images are depicted. The encrypted images have fairly uniform histograms and are significantly different from the original images.

2.7.5 Information-Theoretic Analysis

Entropy is the statistical measure of randomness in a random variable. The entropy of an image is defined as [14]

$$H(m) = -\sum_{i=0}^{K-1} p(m_i)\log_2 p(m_i) \tag{2.17}$$

where $p(m_i)$ is the probability of message m_i, and K is the total number of different messages ($K = 256$ for a gray scale image). When a gray scale message is ideally encrypted, the entropy is equal to eight. An entropy less than eight means some information can be predicted or there is information leakage. Table 2.4 depicts the entropy of the original images and the entropy after encryption.

2.7.6 NIST Statistical Tests

There are numerous tests to evaluate the randomness of an encryption or a data matrix. One widely-used test is the National Institute of Standards and Technology (NIST) statistical test. The software package can be downloaded from NIST web site (http://csrc.nist.gov). The encrypted images were tested using this software package. In each test, the default value of $\alpha = 0.01$ is employed, and at the same time, the P-values of the images are created. An image passes a test if the output is greater than seven, and at the same time, the P-value $\geq \alpha$. The results are shown in Tables 2.5 and 2.6. An encrypted image failing to pass a test is denoted by a star. The encrypted images passed all except the universal test, where the P-values were zero and the test failed. In addition, it is important to mention that the non-overlapping template matching tests

TABLE 2.5: NIST tests results.

ID	Test Name	Dental	Brain
1	Frequency	10	9
2	Block Frequency	10	10
3	Cumulative Sums	10	10
4	Runs	10	10
5	Longest Run	10	10
6	Rank	10	10
7	FFT	10	10
8	Non-Overlapping Template (148 tests)	8 to 10	8 to 10
9	Overlapping Template	10	10
10	Universal	10*	10*
11	Approximate Entropy	10	10
12	Serial	10	10
13	Linear Complexity	10	10

TABLE 2.6: NIST tests results.

ID	Test Name	Lena	Barbara
1	Frequency	10	10
2	Block Frequency	10	10
3	Cumulative Sums	10	10
4	Runs	10	10
5	Longest Run	10	10
6	Rank	10	10
7	FFT	10	10
8	Non-Overlapping Template (148 tests)	9 to 10	7* to 10
9	Overlapping Template	10	10
10	Universal	10*	10*
11	Approximate Entropy	10	10
12	Serial	10	10
13	Linear Complexity	10	10

is a collection of 148 tests. According to the results, all the encrypted images, except Barbara, passed all NIST non-overlapping tests.

2.7.7 KEY SENSITIVITY TEST

An ideal encryption must be sensitive to any small or large change of encryption key. In other words, a tiny change in the key should create significicate change in the output of encryption.

To calculate the difference between two images $P = (P_{ij})$ and $Q = (Q_{ij})$ of size $M \times N$, the Number

TABLE 2.7: Encryption key sensitivity test results.

Image	NPCR (%)	UACI (%)
Dental Image	99.5636	16.9389
Brain Image	99.6332	16.8799
Lena	99.5919	16.7221
Barbara	99.6597	16.7257

TABLE 2.8: Plain text sensitivity test results.

Image	NPCR (%)	UACI (%)
Dental Image	99.5879	16.8417
Brain Image	99.6541	16.7777
Lena	99.5758	16.7333
Barbara	99.6020	16.7489

of Pixel Change Rate (NPCR) is used [14].

$$\text{NPCR} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} D_{ij}}{MN} \times 100 \tag{2.18}$$

where D_{ij} is

$$D_{ij} = \begin{cases} 0, & P_{ij} = Q_{ij}, \\ 1, & P_{ij} \neq Q_{ij}. \end{cases} \tag{2.19}$$

A higher value of NPCR (close to 100%) shows that the two images are extremely different. Moreover, the UACI is the average intensity of differences between two matrices, and it is defined as follows

$$\text{UACI} = \frac{\sum_{i=1}^{M}\sum_{j=1}^{N} |P_{ij} - Q_{ij}|}{255MN} \times 100 \tag{2.20}$$

A high value of UACI also shows that the two images are extremely different. Table 2.7 shows the NPCR and UACI between the new encrypted image and the old one, when one character in the encryption key is changed.

2.7.8 Plain Text Sensitivity Test

Like the key change sensitivity, a good encryption must be sensitive to any tiny change of input plain text. In the plain text sensitivity tests, only one pixel of each sample image was changed, and the NPCR between the new and old encrypted images are depicted in Table 2.8.

2.8 Conclusions

In this chapter, the discrete shearlet transform (DST) was used for image approximation and compression combined with AES encryption to secure the compressed data. The simulation and tests section shows that the quality of the reconstructed image is better that using the DWT. We also applied NIST test, correlation, adjacent pixel correlation, histogram, plain text sensitivity, and key change sensitivity tests to show the quality of encryption. In more secure applications, 256-bit AES encryption provides a much better application security.

In chapter 2, we proposed a method for security of multimedia data against unauthorized disclosure, in the next chapter we will propose a hashing algorithm to provide security of multimedia data against unauthorized copy and usage (copyright protection).

Chapter 3

Information-Theoretic Hashing of 3D Objects

In this chapter, we present a robust 3D hashing technique. We use the spectral analysis of a mesh and the entropic spanning tree to extract the important futures of a 3D mesh to create a unique identifier, which is called the hash. The main idea is to partition a 3D mesh into sub-meshes. Then, the Laplace-Beltrami matrix and Tsallis entropy for each sub-mesh are calculated. We then define a hash value, in terms of spectral coefficients and Tsallis entropy estimator. A vector containing all these hashes is called the hash of the original 3D mesh. The experimental results on a variety of 3D objects demonstrate the robustness and effectiveness of the proposed method against the most common 3D attacks, including tessellation, bending, stretching, rotation, scaling, and pushing.

3.1 Introduction

3D models are used in many multimedia applications, including 3D movies, and 3D games. Even in digital libraries, GIS applications, visual reality, e-commerce, education, and medical applications. There is an increasing demand for 3D object modeling, which makes the security of 3D models an important issue. Furthermore, the ability of users to modify digital content without any perceptual traces adds to the complexity of the issue. To tackle this problem, 3D watermarking and hash functions can be helpful. In 3D watermarking, some data are hidden in a 3D object, and can be used for authorization [22]. The cryptographic hash functions can also be used to control the integrity and the authentication of data. A cryptography hash function is a map that takes an input with arbitrary length and generates a fixed length output. Usually this output is a binary string, which is referred to as the hash of input [23].

By recalculating the hash value from underlying data and comparing it to the attached hash value, we can verify the authenticity of data [39]. In [45], a robust method for image watermarking was introduced that generates hashes based on extracting higher order spectral features from the radon projection of an input image. This projection process is non-invertible and non-linear, so that different hashes can be extracted from the same image if random permutations of the input are used. Another method was introduced in [25], which employs the use of non-negative matrix factorization (NMF) for robust image hashing. The basic idea is to employ the image matrix to create an algorithm for randomized dimensionality reduction that retains the essence of the original image matrix, while resisting different attacks. Recently, Hu *et al.* in [26] used a new variation of the discrete wavelet transform (DWT) algorithm. First, this method extracts robust bits of High pass-Low pass (HL), Low pass-High pass (LH) and High pass-High pass (HH) sub-bands in the same middle-frequency scale. A a hash is then generated, based on these sub-bands. Other recent development includes the presentation of a two stage cascade of dimensionality reduction constructs for face image hashing [27]. The first stage aims to project the face image to a space where geometric distortions manifest approximately as additive noise using the NMF method. In the second stage, Oriented Principal Component Analysis (OPCA), based on estimating signal, is employed.

Typically, a hash algorithm should have three properties: one-way, collision-free, and strongly collision-free [23]. Normal cryptographic hashes are designed to be extremely sensitive to input, meaning that a small change in the input creates a large change in the hash value. This is not desirable in 3D authentication applications. We need a hash algorithm which is resistance against a variety of attacks. In other words, a small change in a 3D mesh should create a small change in the hash value. This is called a perceptual or robust hash.

The 3D mesh hashing problem is much complicated that the 2D image hashing. The technologies that are normally used for 2D cannot be easily extended to 3D. Furthermore, a large number of attacks can be directed against 3D meshes.

In recent years, the use of spectral analysis in 3D mesh analysis, computer vision, graph theory, and machine learning has become more widespread. Generally speaking, a spectral method calculates the eigenvalues and eigenvectors corresponding to a linear operator and try to use them to solve a given problem [33]. This method can be used in variety of problems, including mesh parametrization, mesh authorization, and mesh compression.

The primary motivation of this chapter is to propose a method for 3D mesh authorization and security by encoding the geometrical and topological properties of a 3D mesh. In order to reduce the computational cost associated with finding the eigenvalues and eigenvectors of a large matrix, our method partitions a mesh to small sub-meshes and generates the hash values for each sub-mesh. We perform extensive numerical experiments on different 3D meshes to give expanded insight into the potential of the proposed method. In our approach, we employ the Laplace-Beltrami matrix to convert the 3D mesh to a compact set of data. This is achieved by retaining a small portion of eigenvalues and eigenvectors of that matrix, containing the major properties of a 3D shape, and ignoring the others.

The layout of this chapter is as follows. Section 3.2 is devoted to the mathematical background governing spectral analysis basics, Laplace-Beltrami matrix, spanning tree, and the entropy of a mesh. Section 3.3 presents the proposed 3D hash algorithm. Experimental results using the proposed method are provided in section 3.4. Finally, section 3.5 summarizes the chapter.

3.2 Problem Formulation

A 3D mesh can be represented by using different modeling methods, such as 3D triangle mesh, polygonal mesh, spline and patch, and primitive mesh. A triangle mesh $\mathbb{M}$ can be defined as $\mathbb{M} = (\mathcal{V}, \mathcal{T})$ or $\mathbb{M} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = (\boldsymbol{v}_1, \boldsymbol{v}_2, ..., \boldsymbol{v}_m)$ is a set of vertices, and $\mathcal{E} = \{\boldsymbol{e}_{ij}\}$ is a set of edges, and $\mathcal{T} = (\boldsymbol{t}_1, \boldsymbol{t}_2, ..., \boldsymbol{t}_n)$ is a set of triangles. A pair of vertices $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$ are connected by the edge $\boldsymbol{e}_{ij} = [\boldsymbol{v}_i, \boldsymbol{v}_j]$. Two adjacent vertices are shown as $\boldsymbol{v}_i \sim \boldsymbol{v}_j$ (or simply $i \sim j$) if an edge $\boldsymbol{e}_{ij}$ is connects them. The neighborhood (sometimes referred as the ring) of a vertex $\boldsymbol{v}_i$ is defined as $\boldsymbol{v}_i^* = \{\boldsymbol{v}_j \in \mathcal{V} : \boldsymbol{v}_i \sim \boldsymbol{v}_j\}$. The degree d_i of a vertex $\boldsymbol{v}_i$ is simply the cardinality of $\boldsymbol{v}_i^\star$. We denote by $\mathcal{T}(\boldsymbol{v}_i^\star)$ the set of triangles of the ring $\boldsymbol{v}_i^\star$. Fig. 3.1(a) depicts an example of a neighborhood $\boldsymbol{v}_i^\star$, where the degree of the vertex $\boldsymbol{v}_i$ is $d_i = 7$, and the number of triangles of the set $\mathcal{T}(\boldsymbol{v}_i^\star)$ is also equal to 7.

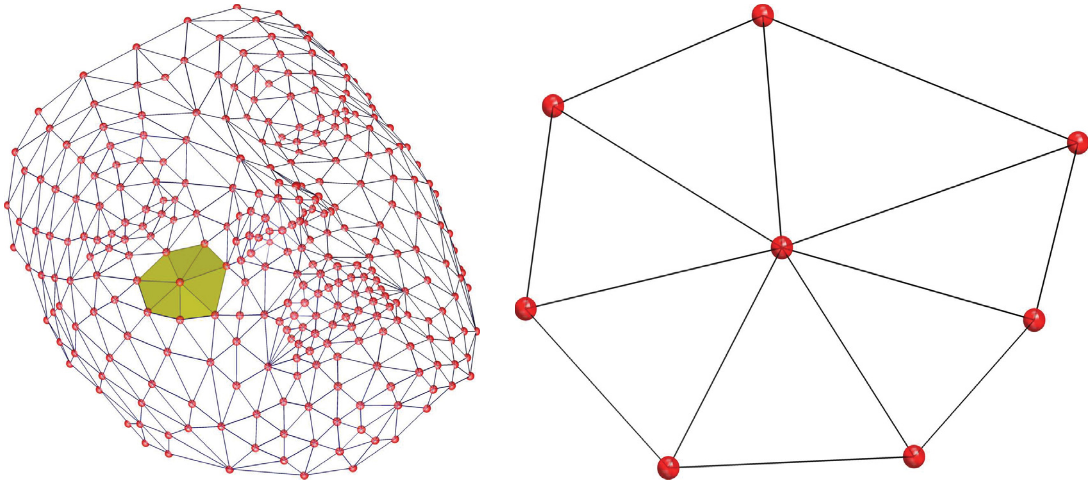


Figure 3.1: Illustration of a triangle mesh (left) and vertex ring (right).

3.2.1 Hash Function Requirements

The object of this chapter is to design a robust hash function that produces a unique identifier for a 3D mesh. The vast majority of cryptographic hash functions take an arbitrary length string (binary or text) as input and produce a fixed-length binary string as output, known as the message digest or the hash. There are many information security applications for hash functions, including the message authentication codes (MAC), and indexing data in the hash tables for fingerprint. They can also be used as checksums to evaluate the data integrity [2]. Fig. 3.2 shows the block diagram of a hash function. Some commonly-used cryptography hash functions are MD5, SHA-1, SHA-512.

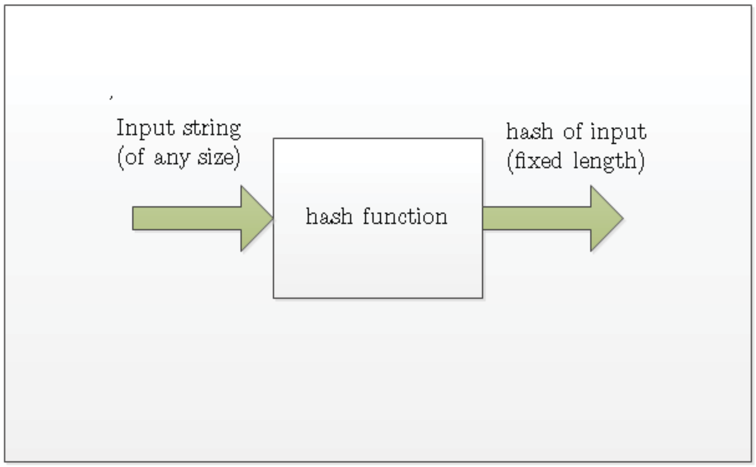


Figure 3.2: Block diagram of a hash function.

A hash function should have three properties: one-way, collision-free, and strongly collision-free.

- **One-way:** Given a 3D mesh $\mathbb{M}$, it should be easy to generate $h = H(\mathbb{M})$.
- **Collision free:** Given h, it must be infeasible to find $\bar{\mathbb{M}}$ such that $h = H(\bar{\mathbb{M}})$
- **Strongly collision-free:** if $\mathbb{M} \neq \bar{\mathbb{M}}$ then $H(\mathbb{M}) \neq H(\bar{\mathbb{M}})$. In other words, two different meshes should have two different hash values.

Conventional hash functions like MD5 or SHA-1 are extremely sensitive to input variation. i.e. 1 bit change in the input message changes the hash value dramatically [28]. Perceptual image hash or perceptual 3D mesh hash functions are different. They create similar hash values for two 3D objects which look the same to the human eye. These perceptual hash algorithms are used for image or 3D mesh search/identification in a large database. In this chapter, we propose a perceptual 3D mesh hashing algorithm.

3.2.2 Eigen-Decomposition of Matrices

Let $\boldsymbol{x} = (x_1, x_2, ..., x_n)^T$ be a non-zero $n \times 1$ vector and $\boldsymbol{A}$ is an $n \times n$ square matrix. The vector $\boldsymbol{x}$ is called the eigenvector of $\boldsymbol{A}$ if and only if it satisfies the following equation [29]

$$\boldsymbol{A}\boldsymbol{x} = \lambda\boldsymbol{x} \tag{3.1}$$

where λ is called the eigenvalue corresponding to $\boldsymbol{x}$ and is a scalar. To find the eigenvalues and eigenvectors we write

$$\boldsymbol{A}\boldsymbol{x} - \lambda\boldsymbol{x} = \boldsymbol{0} \tag{3.2}$$

$$(\boldsymbol{A} - \lambda\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0} \tag{3.3}$$

where $\boldsymbol{I}$ is the $n \times n$ identity matrix. Because $\boldsymbol{x}$ is assumed to be non-zero, so the above equation implies

$$\det(\boldsymbol{A} - \lambda\boldsymbol{I}) = 0 \tag{3.4}$$

The above equation is called *characteristic equation*, and is an nth order polynomial equation in the unknown λ. It has n_i distinct solutions where $1 \leq n_i \leq n$. The set of eigenvalues is sometimes called the *spectrum* of $\boldsymbol{A}$. For each eigenvalue, λ_i, we have

$$(\boldsymbol{A} - \lambda_i\boldsymbol{I})\boldsymbol{x} = \boldsymbol{0} \tag{3.5}$$

that gives $1 \leq m_i \leq n_i$ solutions for each eigenvalue. These vectors are called the eigenvectors associated with the specific eigenvalue λ_i. It is important to mention that m_i may or may not be equal to n_i, but always satisfies $1 \leq m_i \leq n_i$. After calculating the eigenvalues and eigenvectors, we can decompose the matrix as

$$\boldsymbol{A} = \boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{B}^{-1} \tag{3.6}$$

where $\boldsymbol{B} = (\boldsymbol{b}_1, ..., \boldsymbol{b}_n)$ is an orthogonal matrix whose columns are the eigenvectors and $\boldsymbol{\Lambda} = \mathrm{diag}\{\lambda_i : i = 1, 2, ..., n\}$ is a diagonal matrix of the eigenvalues sorted in descending order of magnitude. The eigenvectors are mostly normalized, but it is not necessary. If $\boldsymbol{A}$ is a symmetric matrix ($\boldsymbol{A} = \boldsymbol{A}^T$), then it has n independent eigenvectors, and the above equation can be written as

$$\boldsymbol{A} = \boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{B}^T \tag{3.7}$$

In addition, if $\boldsymbol{A}$ is a small matrix, eigenvalues and eigenvectors can be computed symbolically using the characteristic polynomial. Otherwise, numerical methods should be used. In practice mostly numerical methods are employed because, according to Abel's theorem, roots of a polynomial equation of order 5th and higher cannot be expressed simply using nth roots. The iterative algorithms like

- Power method [30]
- Arnoldi iteration [31]
- QR algorithm [32]

are often employed to compute eigenvalues.

3.2.3 Spectral Analysis of a 3D Mesh

Most spectral methods have a basic framework that can be summarized in three steps [33]

- A matrix $L = (L_{ij})$ is defined according to a mesh $\mathbb{M}$. Each entry L_{ij} posses a relation between two vertices $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$ of the mesh. Sometimes two triangles or two edges are employed to define entry L_{ij}.
- Decomposition of L is performed. i.e., the eigenvalues and eigenvectors of matrix are calculated.
- These eigenvalues and eigenvectors are employed to solve a problem.

According to the above framework, different ways of using the eigenvalues and the eigenvectors generate different spectral methods. Spectral methods can be classified into three important categories:

- **Based on the operator used to define $\boldsymbol{L}$.** i.e. Laplacian operator, adjacency matrix or Lagrangian, and discrete Schrödinger operators.
- **Based on the eigenstructures**. Some methods use eigenvalues, some employ eigenvectors, and some made use of both.
- **Based on the dimensionality of the eigenstructure**. The number of eigenvalues and eigenvectors that are used to solve problems.

3.2.4 Laplace-Beltrami Matrix

The Laplace-Beltrami operator is defined as

$$\triangle_m \boldsymbol{v}_i = \frac{3}{\eta} \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^*} (\cot \alpha_{ij} + \cot \beta_{ij})(\boldsymbol{v}_j - \boldsymbol{v}_i) \tag{3.8}$$

where α_{ij} and β_{ij} are the angles $\angle \boldsymbol{v}_i \boldsymbol{v}_{j-1} \boldsymbol{v}_j$ and $\angle \boldsymbol{v}_i \boldsymbol{v}_{j+1} \boldsymbol{v}_j$ respectively (Fig. 3.3), and

$$\eta = \sum_{\boldsymbol{t}_j \in \mathcal{T}(\boldsymbol{v}_i^*)} \text{area}(\boldsymbol{t}_j) \tag{3.9}$$

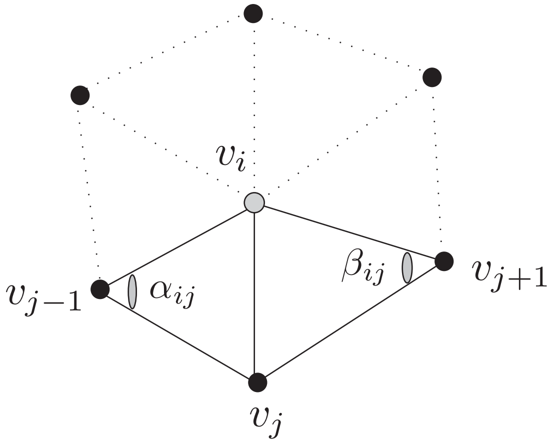


FIGURE 3.3: Illustration of Laplace-Beltrami angles α_{ij} and β_{ij}.

We define the Laplace-Beltrami matrix as

$$L = D - W = \begin{cases} d_i - \omega_{ii}, & \text{if} \quad \boldsymbol{v}_i = \boldsymbol{v}_j, \\ -\omega_{ij}, & \text{if} \quad \boldsymbol{v}_i \sim \boldsymbol{v}_j, \\ 0, & \text{otherwise.} \end{cases} \tag{3.10}$$

where $W = (\omega_{ij})$ is the weighted adjacency matrix given by

$$\omega_{ij} = \frac{3}{\eta}(\cot \alpha_{ij} + \cot \beta_{ij}) \tag{3.11}$$

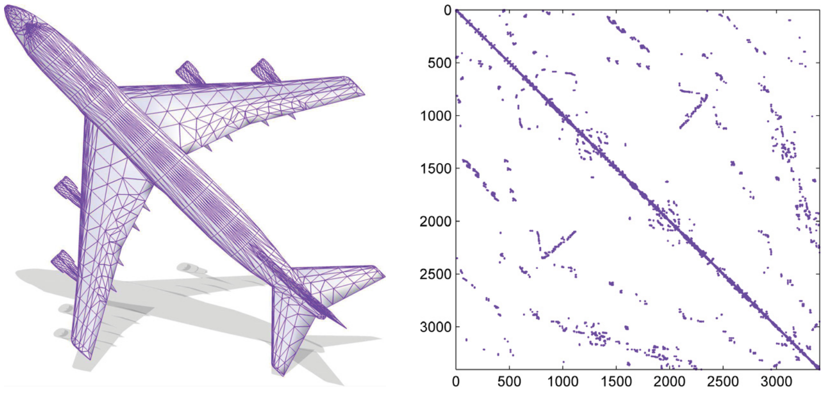
and D is a diagonal matrix D given by

$$D = \text{diag}\{d_i : \boldsymbol{v}_i \in \mathcal{V}\} \tag{3.12}$$

where

$$d_i = \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^*} \omega_{ij} \tag{3.13}$$

Fig. 3.4 illustrates a triangular 3D mesh and its sparse Laplace-Beltrami matrix.



(a) (b)

FIGURE 3.4: (a) 3D airplane model; (b) sparsity pattern of the Laplace-Beltrami matrix.

3.2.5 Spanning Tree and Minimum Spanning Tree

A spanning tree of a graph G is a connected graph that passes through all vertices of G. There are many places where the spanning trees are used, i.e., minimizing the cabling expenses in a LAN, in routing algorithms, connecting different factories together with roads, and bioinformatics. A spanning tree is acyclic and specified by an ordered list of the edges connecting certain pairs $[\boldsymbol{v}_i, \boldsymbol{v}_j], i \neq j$, along with a list of edge adjacency relations. Fig. 3.5 shows a 4-vertex complete graph and Fig. 3.6 depicts some of its spanning trees.

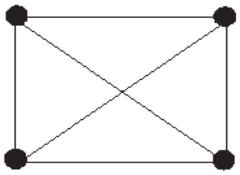


FIGURE 3.5: A 4-vertex complete graph.

As shown in [34] a graph with n vertices has K_n distinct spanning trees.

$$K_n = n^{n-2} \tag{3.14}$$

For example, a graph with $n = 6$ vertices has 1296 distinct spanning trees. A weighted graph associates a label (a real number that is called the weight or cost) with every edge in the graph. Minimum Spanning Tree (MST) of a graph G is a spanning tree that has two properties: (1) it includes every vertex on the graph

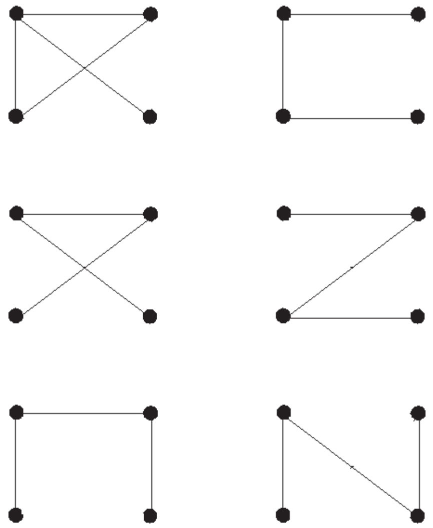


FIGURE 3.6: Some of the spanning trees of the complete 4-vertex graph.

(2) the total weight of all the edges is as low as possible. Generally, the MST of a graph is not unique. A number of algorithms can be used to calculate MST, such as:

- Boruvka's algorithm
- Prim's algorithm
- Reverse-Delete algorithm
- Kruskal's algorithm

Kruscal's algorithm is a fast method and is described in Algorithm 2 [35].

3.2.6 Entropy of a Graph

Entropy is the measure of unpredictability in a random process. For example, a fair coin has higher entropy than an unfair coin, because a series of tosses with a fair coin has higher unpredictability than the same number of tosses with an unfair coin. Réyni entropy of a discrete system expresses the quantification of diversity, and the uncertainty or randomness of a system, and can be defined as

$$\widehat{H}_\alpha(X) = \frac{1}{1-\alpha}\log_2\left(\sum_{i=1}^{n} p_i^\alpha\right) \tag{3.15}$$

Algorithm 2 Kruscal's algorithm for computing MST.

```
Create a forest $F$ (A forest is a set of trees), where each vertex in the graph is considered a separate tree
Create a set $S$ containing all the edges $\{e_i\}$ in the graph in the ascending order of weight
$i \leftarrow 1$
while ($S \neq$ empty) and ($F \neq$ spanning) do
    Delete the edge $e_i$ from $S$
    if $e_i$ connects two different trees then
        Add $e_i$ to the forest
        Combine two trees into a single tree
    else
        Discard $e_i$
    end if
    $i \leftarrow i + 1$
end while
```

where X is a discrete random variable with possible values $\{x_1, x_2, ..., x_n\}$, $p_1, p_2, ..., p_n$ are the probabilities of $x_1, x_2, ..., x_n$ respectively, and $\alpha \geq 0,\ \alpha \neq 1$ is the order of entropy [36]. If $\alpha \to 1$, then Shannon entropy is achieved

$$\widehat{H}_\alpha(X) = -\sum_{i=1}^{n} p_i \log_2 p_i \tag{3.16}$$

As a generalization of Boltzmann-Gibbs entropy, Tsallis entropy has recently gained a great deal of interest in statistical physics. When a system is composed of two statistically independent subsystems, Shannon-Réyni entropy is the simple addition (sum) of the entropy of each subsystem and the correlations between subsystems are not accounted for. To overcome this weakness, we use the Tsallis entropy of a system [37]. Tsallis entropy of a discrete distribution is defined as

$$\widehat{H}_\alpha(X) = \frac{1}{\alpha - 1}\left(1 - \sum_{i=1}^{n} p_i^\alpha\right) \tag{3.17}$$

where α is a real parameter. When α is close to 1, the normal Boltzmann-Gibbs entropy or Shannon-Réyni entropy is recovered [37]. Let two random variables $X = \{x_i\}$ and $Y = \{y_i\}$, with $\{p_i\}$ and $\{q_i\}$ as the probabilities of $\{x_i\}$ and $\{y_i\}$ respectively. The joint probability density is defined as

$$p(x_i, y_j) = p(X = x_i,\, Y = y_j) \tag{3.18}$$

If X and Y are independent then $p(x_i, y_j)$ satisfies the following equality

$$p(x_i, y_j) = p(x_i)p(y_j) = p_i q_j \tag{3.19}$$

then Tsallis entropy of combined system can be written as

$$\widehat{H}(X, Y) = \widehat{H}(X) + \widehat{H}(Y) + (1 - \alpha)\widehat{H}(X)\widehat{H}(Y) \tag{3.20}$$

Proof:

$$\widehat{H}(X) + \widehat{H}(Y) + (1-\alpha)\widehat{H}(X)\widehat{H}(Y) =$$

$$\frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n} p_i^\alpha\right) + \frac{1}{\alpha-1}\left(1-\sum_{j=1}^{m} q_j^\alpha\right) + (1-\alpha)\frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n} p_i^\alpha\right) \times \frac{1}{\alpha-1}\left(1-\sum_{j=1}^{m} q_j^\alpha\right) =$$

$$\frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n} p_i^\alpha \sum_{j=1}^{m} q_j^\alpha\right) = \frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{m} p_i^\alpha q_j^\alpha\right) = \frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{m} (p_i q_j)^\alpha\right) =$$

$$\frac{1}{\alpha-1}\left(1-\sum_{i=1}^{n}\sum_{j=1}^{m} p(x_i, y_j)^\alpha\right) = \widehat{H}(X,Y)$$

This means that the correlation of two subsystems are accounted in the Tsallis entropy of the composed system. This property is called *pseudo-additivity*. If $\alpha \to 1$, the Shannon entropy is obtained, and the entropy of the composed system is the simple addition of the entropy of subsystems (*additive* system).

The estimator of Tsallis entropy for a 3D mesh is given by

$$\widehat{H}_\alpha(\mathcal{V}) = \frac{1}{1-\alpha}\left[\frac{L^*(\mathcal{V})}{\beta m^\alpha} - 1\right] \tag{3.21}$$

where $L^*(\mathcal{V})$ is the total length of the minimum spanning tree [34]. α is referred to as an entropic index, and β is a constant playing a role of bias correction [38]. To calculate the minimal spanning tree we use Kruskal's algorithm.

3.3 Proposed Method

The proposed method uses the Laplace-Beltrami matrix eigen-decomposition and Tsallis entropy of the spanning tree to create a hash vector for a 3D mesh [39]. If the Laplace-Baltrami matrix is of size $m \times m$, then calculating the eigenvalues and eigenvectors of a large sparse matrix is a time and resource consumer process and is proportional to $\mathcal{O}(m^3)$. Thus, we need partition a mesh to smaller sub-meshes to increase the hash generation speed. The implementation of partitioning algorithm is based on MeTiS method [40, 41, 42]. The proposed method is summarized in Algorithm 3 and Fig. 3.7.

Figs. 3.20-3.21 depict the 3D shark and dragon models and their eight corresponding sub-meshes. The entropy of each partition is written under the displayed pictures.

Algorithm 3 Proposed algorithm to generate the hash of a 3D mesh.

```
Partition the 3D mesh $\mathbb{M}$ into $s$ sub-meshes:
    $\mathbb{M} = \mathbb{M}_1 \cup \mathbb{M}_2 \cup ... \cup \mathbb{M}_s$    (3.22)
  where the cardinality of each vertex set of each sub-mesh $\mathbb{M}_k$ is $m_k$.
Compute the entropy of each sub-mesh using
    $\xi_k \simeq 2 \left[ \frac{L^*(\mathcal{V}_k)}{\sqrt{m_k}} - 1 \right], \quad k = 1, 2, ..., s$    (3.23)
  where $\xi_k$ is the entropy of $k$th sub-mesh, and $\alpha$ index of Tsallis entropy is set to 0.5.
Apply the eigendecomposition to Laplace-Beltrami matrix of each sub-mesh ($L_k$) to generate the $m_k$ eigenvalues and the corresponding eigenvectors.
    $L_k = B_k \Lambda_k B_k^T$    (3.24)
  where $B_k = (\boldsymbol{b}_1, ..., \boldsymbol{b}_{m_k})$ is an orthogonal matrix whose columns are eigenvectors and $\Lambda_k = \mathrm{diag}\{\lambda_i : i = 1, 2, ..., k\}$ is a diagonal matrix of eigenvalues sorted in the descending order of the magnitude.
Retain $r$ ($r < m_k$) significant eigenvalues and their corresponding eigenvectors (which account for most of the energy of the mesh).
Compute the hash of each sub-mesh with the following formula
    $\mu_k = \sum_{i=1}^{r} \lambda_i^{\xi_k} ||\boldsymbol{b}_i||^2$    (3.25)
Stack the hash values of all sub-meshes into one single vector to create the hash vector of the original mesh.
    $\boldsymbol{h} = (\mu_1, \mu_2, ..., \mu_s)$    (3.26)
```

3.4 Experimental Results

We applied the propose method to different 3D meshes. In all experiments, we selected $s = 8$. Higher s value does not decrease the execution time significantly, but smaller s increases the execution time of the program. Obviously for more complex 3D meshes, higher s value i.e. 16 or 32 or 64 should be selected.

The following attacks are applied to the sample 3D objects [44]

1. **Tessellate**: This modifier can be used to subdivide the faces in the current selection area. Tessellate is mostly used to smooth curved surfaces and create additional mesh resolution for other modifiers. A positive tension value causes meshes to become convex, while negative value makes the shape concave. Fig. 3.8 shows a 3D mesh and the tessellation applied to polygonal or triangular faces.

2. **Welder**: It welds all vertices in a distance d_w of a vertex. It is used to make a 3D mesh cleaner when

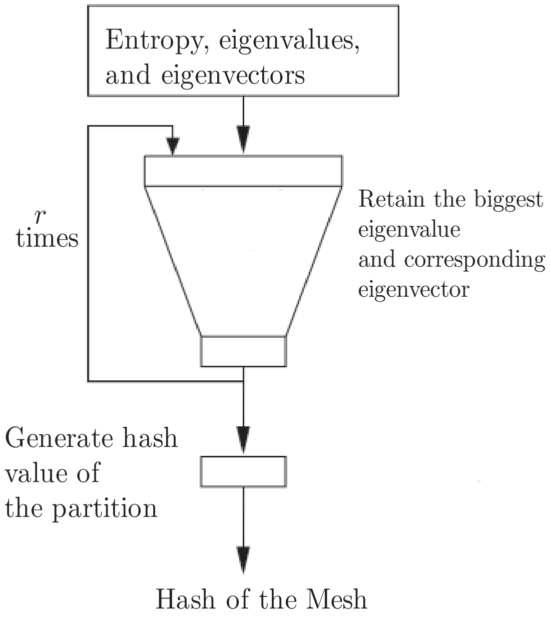

Entropy, eigenvalues, and eigenvectors
r times
Retain the biggest eigenvalue and corresponding eigenvector
Generate hash value of the partition
Hash of the Mesh

FIGURE 3.7: Block diagram of the proposed hash function.

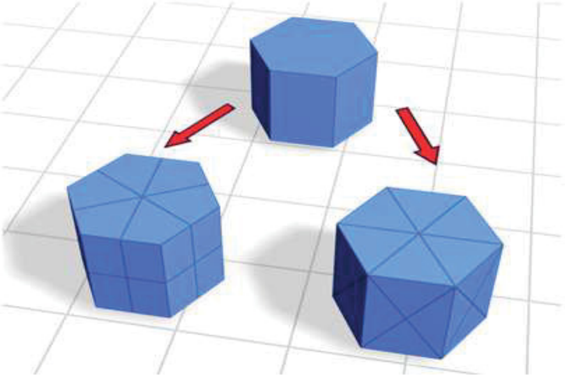


FIGURE 3.8: Tessellate modifier: top: original mesh object, lower left: tessellation applied to polygonal facets, lower right: tessellation applied to triangular faces [43].

the vertices are close or overlapped. In other words, it is used to smooth the tears in a mesh. If d_w is selected high, many vertices are welded and small meshes are removed. This deforms the shape of the 3D object.

3. **Optimize**: It reduces the number of mesh vertices to make it simpler, without major change in mesh quality. Sometimes, when the optimization is applied, no visual change can be detected in the 3D

shape. It is, therefore, necessary to change the optimization value to get desirable results. Optimization improves the rendering time of a 3D mesh. Fig. 3.9 shows a 3D model and its optimized version.

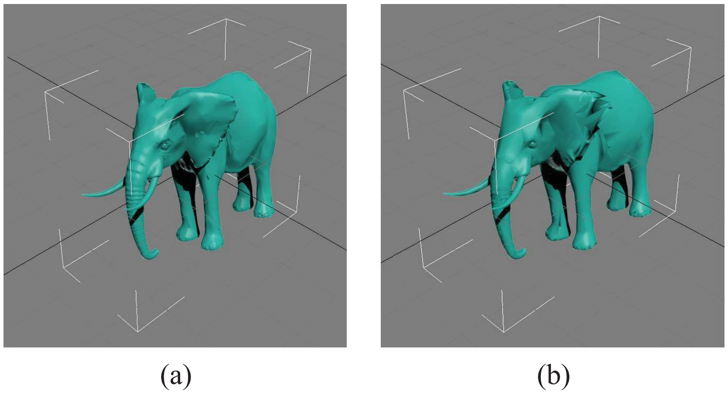

(a) (b)

FIGURE 3.9: Optimize modifier: (a) original model, (b) optimized model. (Face threshold = 10, Edge threshold = 5)

4. **Bend**: It bends a 3D mesh in the x, y, or z direction by k degrees. Fig. 3.10 shows a 3D mesh and the bent model.

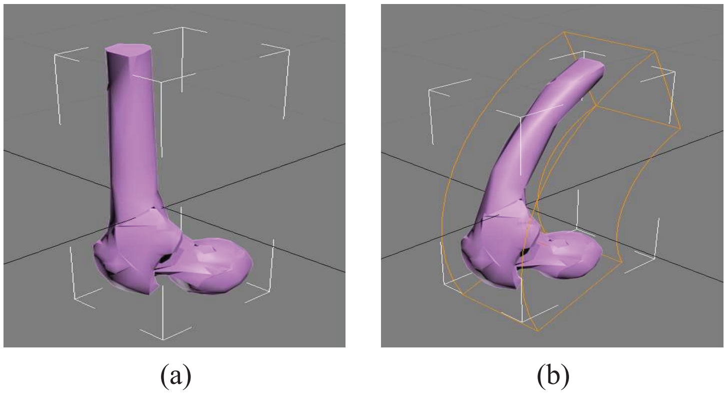

(a) (b)

FIGURE 3.10: Bend modifier: (a) original model, (b) bent model. (Around x-axis 90°)

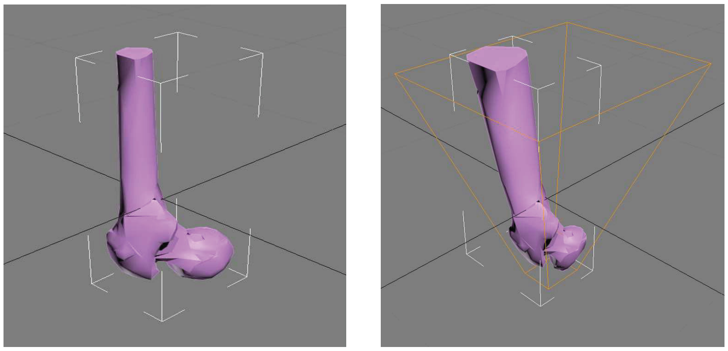
5. **Taper**: It scales one of the ends of a 3D mesh (Fig. 3.11).

6. **Twist**: It twists a 3D mesh in the x, y, or z direction by k degrees. (Fig. 3.12).

7. **Stretch**: It stretches a 3D mesh in the x, y, or z direction by k percent. It creates a scale in one direction and an opposing scale effect to the other two minor axes. (Fig. 3.13).

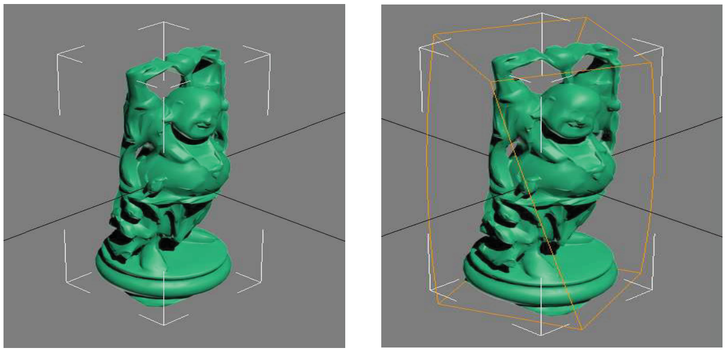
8. **Rotate**: It rotates a 3D mesh in the x, y, or z direction by k degrees.

9. **Scale**: It scales a 3D mesh in three directions by k percent.



(a) (b)

FIGURE 3.11: Taper modifier: (a) original model, (b) taped model. (Around z-axis, Amount = 5)



(a) (b)

FIGURE 3.12: Twist modifier: (a) original model, (b) twisted model. (Around z-axis, Amount = 45°)

10. **Squeeze**: It squeezes a 3D mesh by closing the vertices of a mesh to an axis (Fig. 3.14).

11. **Noise**: It creates a random noise based on 3D mesh shape and adds it to the vertices. In other words, the noise modifier modulates the position of a 3D mesh vertex along any combination of three axes (Fig. 3.15).

12. **Push**: It is used to create a bulging or shrunken appearance. Push modifier let us to change the vertices of an object inward or outward along average vertex normals. Fig. 3.16 depicts a 3D mesh and the pushed variations of it.

13. **Skew**: It skews a 3D mesh on x, y, or z direction n degree (Fig. 3.17).

14. **Spherify**: It distorts an 3D mesh in a spherical shape n percent (Fig. 3.18).

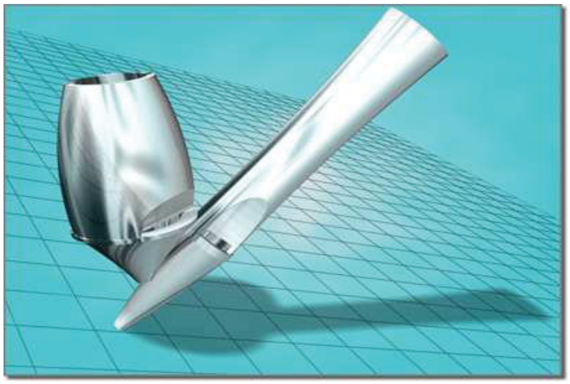
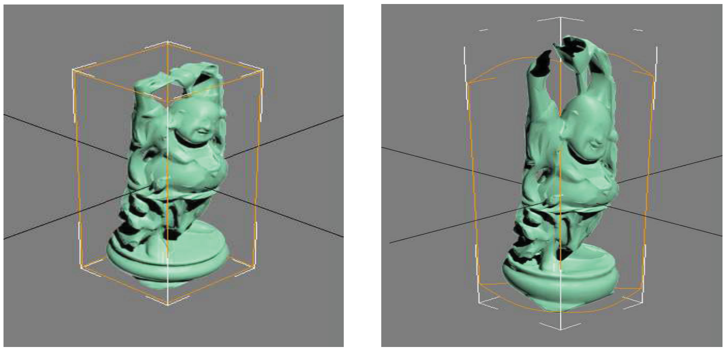


FIGURE 3.13: Stretch modifier: left: original model, right: stretched model [43].



(a) (b)

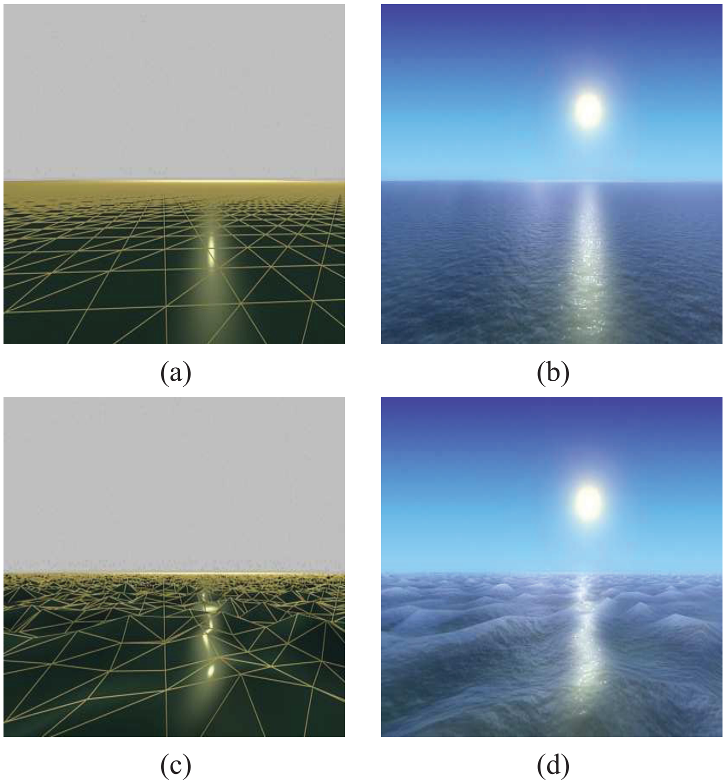
FIGURE 3.14: Squeeze modifier: (a) original model, (b) squeezed model. (Amount = 1)

15. **Relax**: It separates the vertices that are close to an average distance and smooths the overall geometry of a 3D mesh (Fig. 3.19).

Fig. 3.22 depicts some sample 3D models, and Fig. 3.23 shows the 3D vase model and the results of the above attacks. Figs. 3.24-3.25 depicts 3D dragon and rabbit models and corresponding changes after different attacks [44]. To evaluate the robustness of our method, we define the normalized correlation between the hash vectors as follows

$$\rho = \frac{|\boldsymbol{h}_1.\boldsymbol{h}_2|}{||\boldsymbol{h}_1||\,||\boldsymbol{h}_2||} \tag{3.27}$$

where $\boldsymbol{h}_1$ and $\boldsymbol{h}_2$ are the hash values before and after the attack respectively. Table 3.1 depicts the performance of the proposed method. As we can see, the performance of the proposed hashing method is pretty good and the correlation of the original and attacked 3D models are quite high.



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

FIGURE 3.15: Noise modifier: (a) plane with no noise applied, (b) adding texture to the plane creates a calm sea, (c) plane with fractal noise applied, (d) textured plane with noise creates a stormy sea [44].

3.5 CONCLUSIONS

In this chapter, we introduced a 3D hashing method that partitions a 3D mesh into many sub-meshes. We then generate the Laplace-Beltrami matrix for each sub-mesh, and eigen-decompose the matrix to extract the eigenvectors and eigenvalues. Finally, we use the Tsallis entropy of each sub-mesh, as well as the eigenvalues and eigenvectors to calculate the hash vector of the 3D mesh. In the experimental results section, we evaluated the performance and robustness of our algorithm under different well-known attacks.

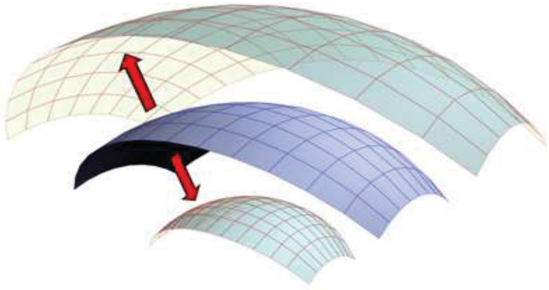
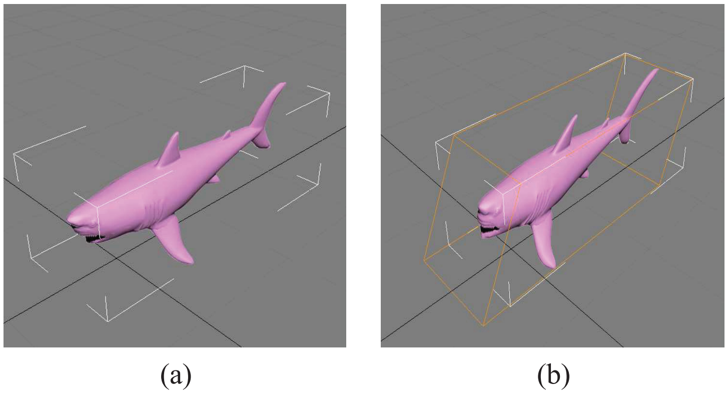


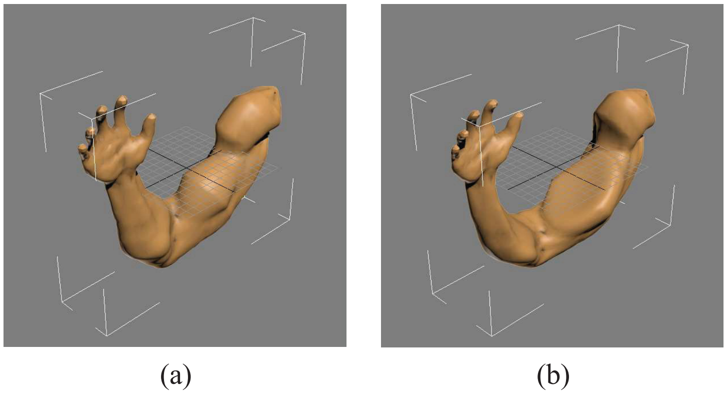
FIGURE 3.16: Push modifier: middle: original mesh object, top: push outward, down: push inward [44].



(a)

(b)

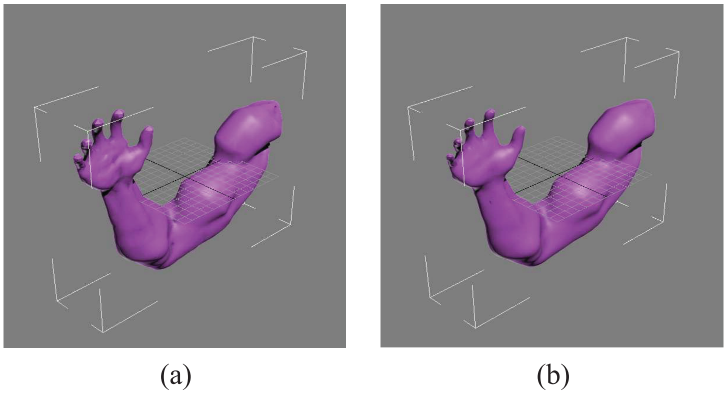
FIGURE 3.17: Skew modifier: (a) original model, (b) skewed model. (Z-axis $45°$)



(a)

(b)

FIGURE 3.18: Spherify modifier: (a) original model, (b) spherified model. (Value = 30%)

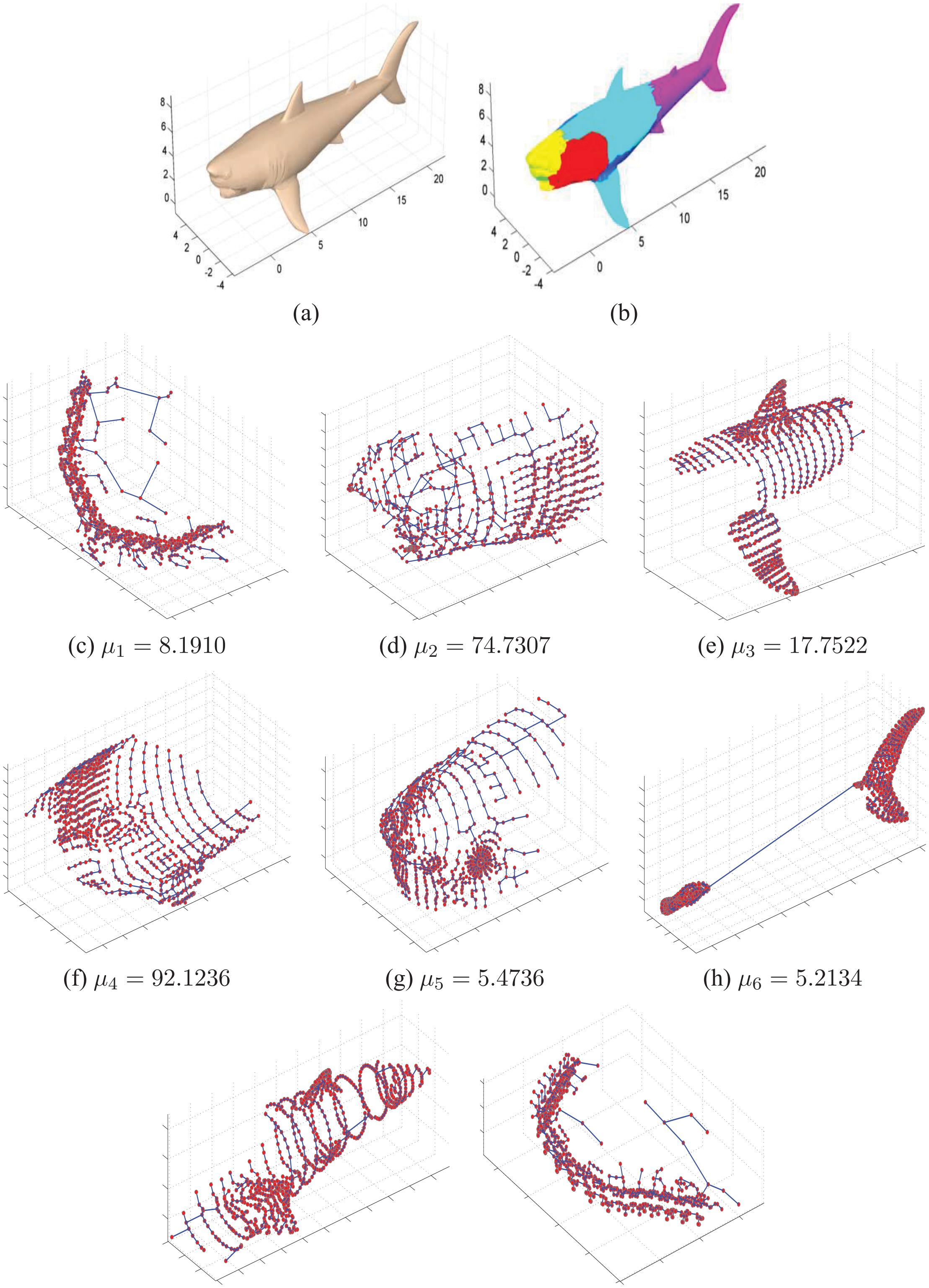


(a) (b)

FIGURE 3.19: Relax modifier: (a) original model, (b) relaxed model. (Value = 3)

TABLE 3.1: Normalized hash correlation results with different 3D models.

Attacks	3D Models					
	Cactus	Dragon	Female	Happy small	Femur	Rabbit
Mesh Scaling with X*2	0.98	0.99	0.99	0.99	0.99	0.99
Mesh Scaling with Y*2	0.99	0.99	0.98	0.99	0.99	0.99
Mesh Scaling with Z*2	0.99	0.99	0.99	0.99	0.99	0.99
Rotation around X 45°	1.0	0.99	1.0	1.0	1.0	1.0
Rotation around Y 45°	1.0	0.99	1.0	1.0	1.0	1.0
Rotation around Z 45°	1.0	0.99	1.0	1.0	1.0	1.0
Mesh Smoothing (10 iterations)	0.99	0.99	0.99	0.99	1.0	0.99
Gaussian Noise ($\sigma = 0.25$)	0.99	0.99	0.99	0.99	0.99	0.99
Tessellate (Tension = 25)	0.74	0.90	0.68	0.95	0.74	0.94
Welder (Threshold = 0.1)	1.0	0.72	1.0	1.0	1.0	1.0
Optimize (Face threshold = 4, Edge threshold = 1)	0.47	0.97	0.67	0.93	0.75	0.99
Bend (Z-axis 20°)	0.57	0.92	0.68	0.96	0.78	0.99
Taper (Amount = 0.2)	0.77	0.91	0.77	0.92	0.86	0.99
Twist (Z-axis 45°)	0.54	0.94	0.68	0.84	0.99	0.99
Stretch (Stretch = 0.2)	0.90	0.91	0.59	0.98	0.99	0.99
Squeeze (Axial bulge = 0.3)	0.98	0.88	0.99	0.99	0.97	0.99
Push (Value = 0.2)	0.92	0.91	0.48	0.90	0.71	1.0
Skew (Amount = 0.1)	0.91	0.95	0.64	0.95	0.94	0.99
Spherify (Percent = 30%)	0.64	0.89	0.74	0.95	0.78	0.98
Relax (Value = 1, Iteration = 1)	0.70	0.94	0.58	0.91	0.93	0.99



(a)

(b)

(c) $\mu_1 = 8.1910$

(d) $\mu_2 = 74.7307$

(e) $\mu_3 = 17.7522$

(f) $\mu_4 = 92.1236$

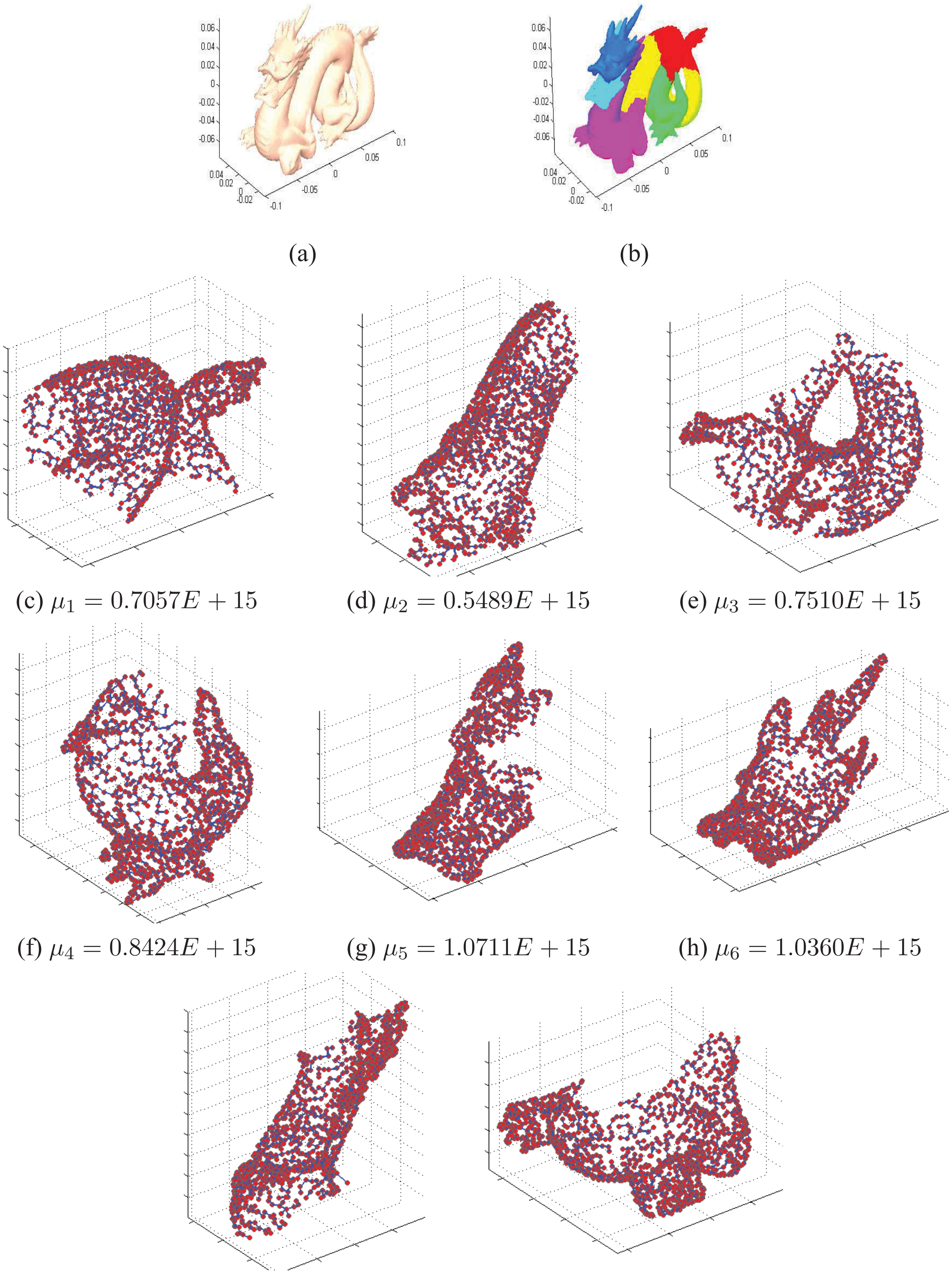
(g) $\mu_5 = 5.4736$

(h) $\mu_6 = 5.2134$

(i) $\mu_7 = 5.9408$

(j) $\mu_8 = 18.0399$

FIGURE 3.20: 3D shark model: (a) original mesh, (b) partitioned mesh. The partitions are colored randomly, (c)-(j) minimal spanning trees for each partition and the corresponding hash.



(a)

(b)

(c) $\mu_1 = 0.7057E + 15$

(d) $\mu_2 = 0.5489E + 15$

(e) $\mu_3 = 0.7510E + 15$

(f) $\mu_4 = 0.8424E + 15$

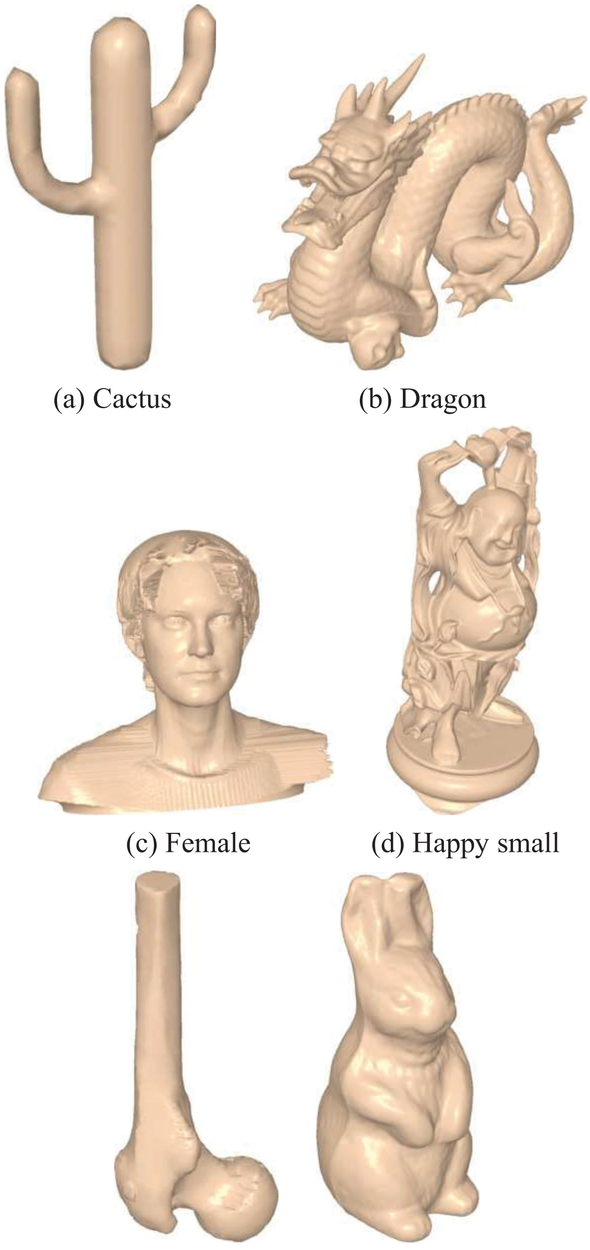
(g) $\mu_5 = 1.0711E + 15$

(h) $\mu_6 = 1.0360E + 15$

(i) $\mu_7 = 0.6606E + 15$

(j) $\mu_8 = 0.7398E + 15$

FIGURE 3.21: 3D dragon model: (a) original mesh, (b) partitioned mesh. The partitions are colored randomly, (c)-(j) minimal spanning trees for each partition and the corresponding hash.

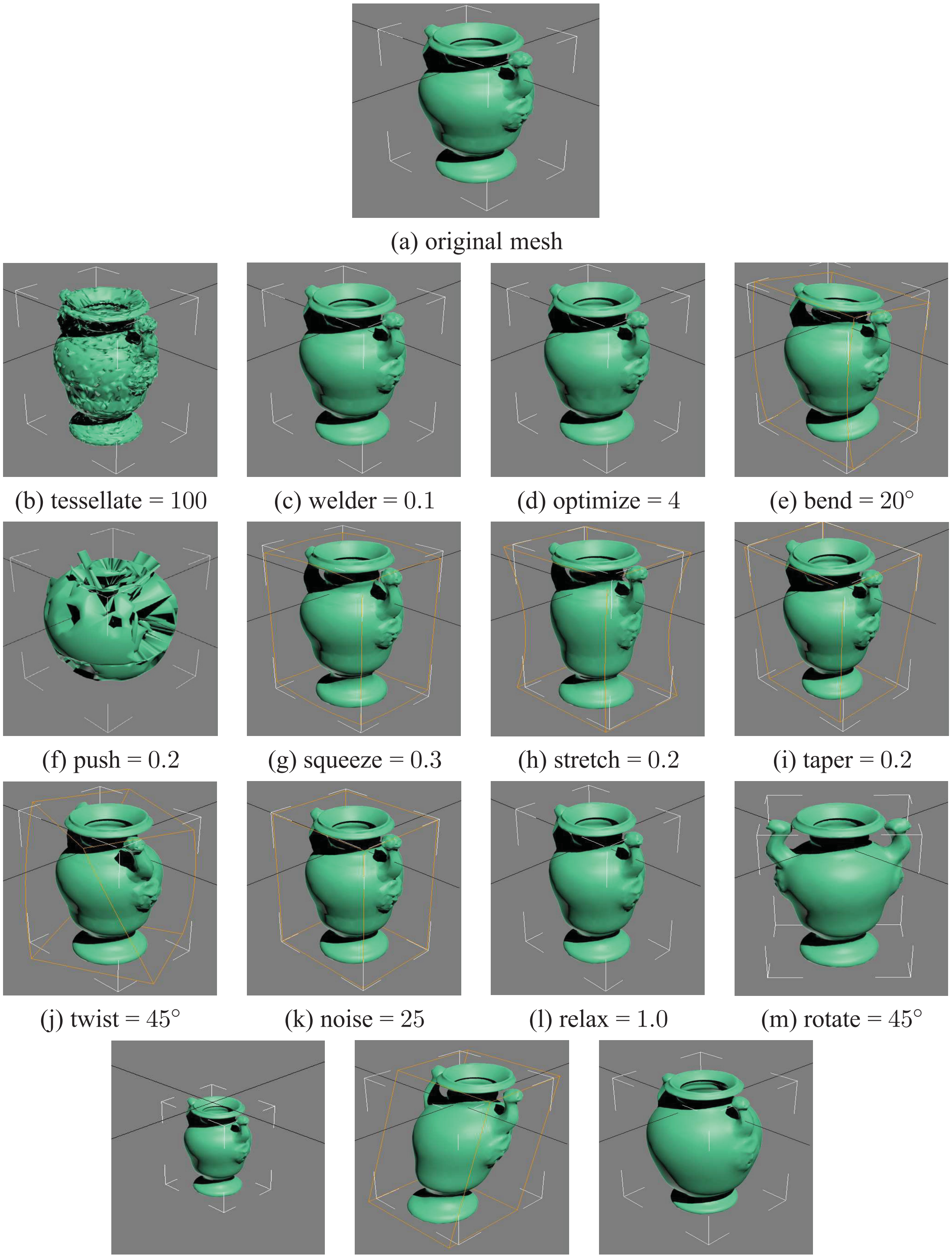


(a) Cactus (b) Dragon

(c) Female (d) Happy small

(e) Femur (f) Rabbit

FIGURE 3.22: Some of the 3D models used in our experiments.



(a) original mesh

(b) tessellate = 100

(c) welder = 0.1

(d) optimize = 4

(e) bend = 20°

(f) push = 0.2

(g) squeeze = 0.3

(h) stretch = 0.2

(i) taper = 0.2

(j) twist = 45°

(k) noise = 25

(l) relax = 1.0

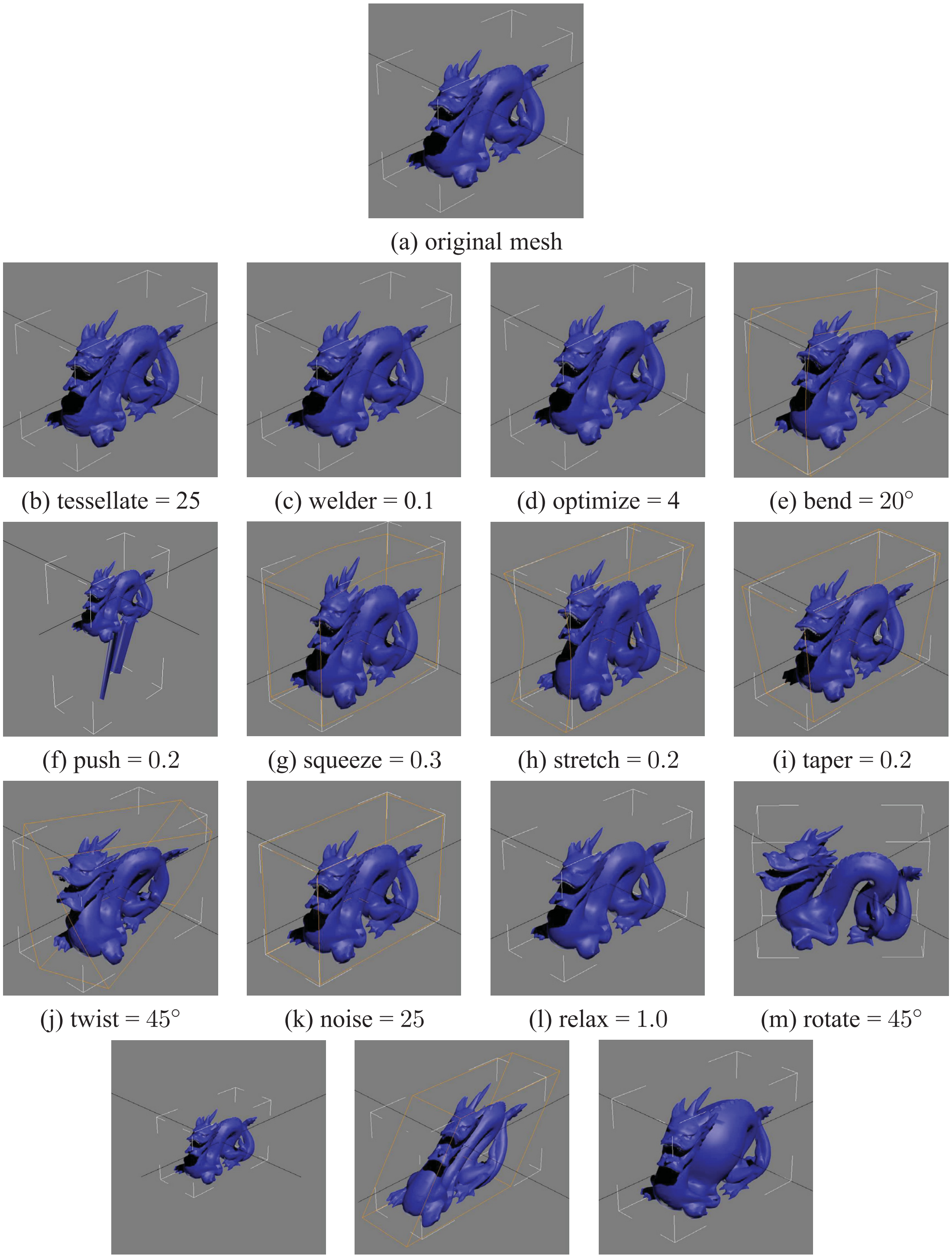
(m) rotate = 45°

(n) scale = 0.5

(o) skew = 0.1

(p) spherify = 30

FIGURE 3.23: 3D vase model: (a) original mesh, (b)-(p) mesh after different attacks.



(a) original mesh

(b) tessellate = 25

(c) welder = 0.1

(d) optimize = 4

(e) bend = 20°

(f) push = 0.2

(g) squeeze = 0.3

(h) stretch = 0.2

(i) taper = 0.2

(j) twist = 45°

(k) noise = 25

(l) relax = 1.0

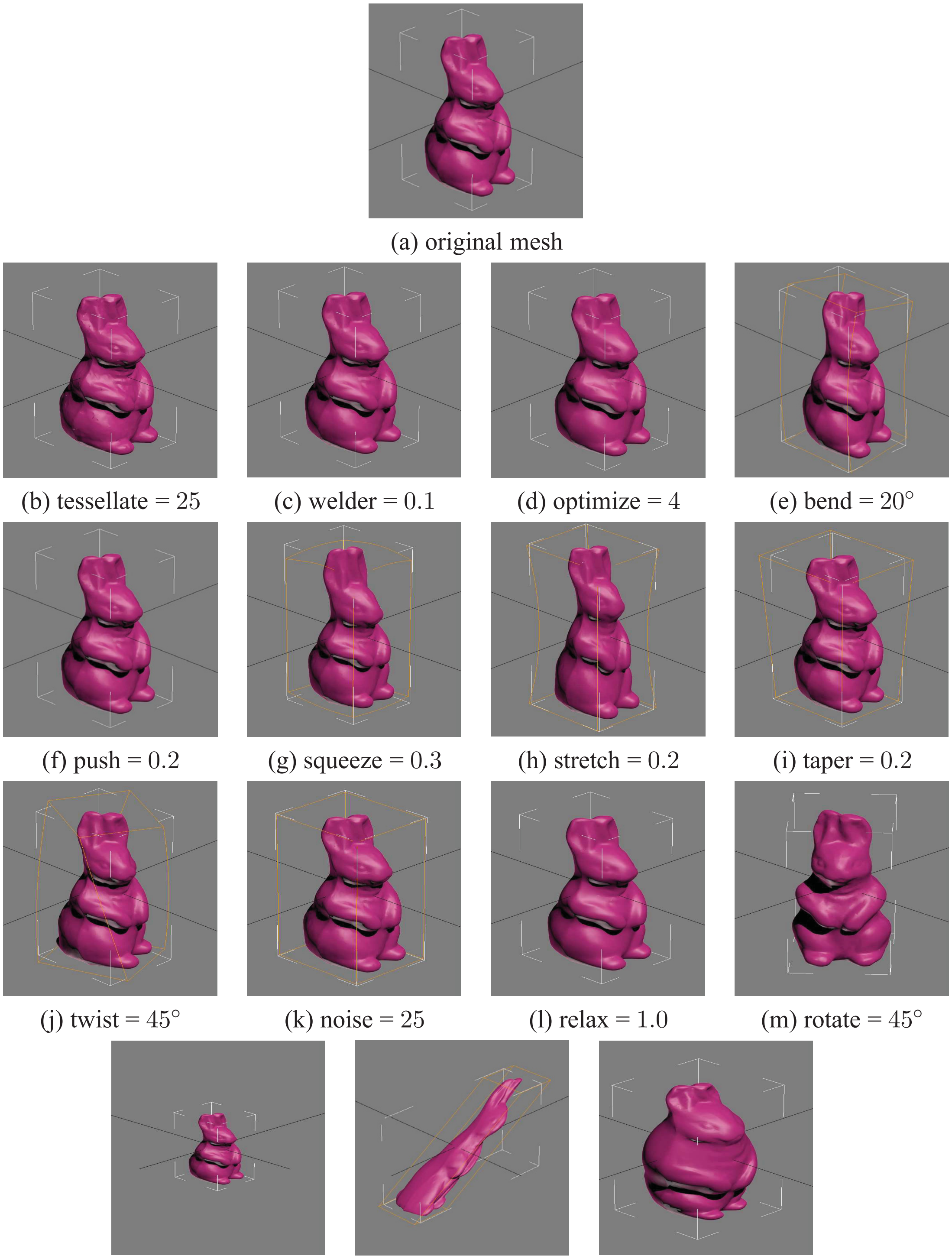
(m) rotate = 45°

(n) scale = 0.5

(o) skew = 0.1

(p) spherify = 30

FIGURE 3.24: 3D dragon model: (a) original mesh, (b)-(p) mesh after different attacks.



(a) original mesh

(b) tessellate = 25

(c) welder = 0.1

(d) optimize = 4

(e) bend = 20°

(f) push = 0.2

(g) squeeze = 0.3

(h) stretch = 0.2

(i) taper = 0.2

(j) twist = 45°

(k) noise = 25

(l) relax = 1.0

(m) rotate = 45°

(n) scale = 0.5

(o) skew = 0.1

(p) spherify = 30

FIGURE 3.25: 3D rabbit model: (a) original mesh, (b)-(p) mesh after different attacks.

Chapter 4

Conclusions and Future Work

This thesis has presented a robust approach for multimedia compression and encryption protection as well as 3D mesh hashing. We have demonstrated the performance of the proposed algorithms through extensive experiments, and compared our techniques to existing methods in the literature. A variety of images and 3D objects were used in the experiments to show the effectiveness of the proposed schemes.

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 also provided in Section 4.2.

4.1 Contributions of the Thesis

4.1.1 Image Compression using DST and Encryption with AES

We proposed a compression method using discrete shearlet transform, and an encryption using the Advanced Encryption Standard (AES) method. We tested our proposed compression/encryption technique through extensive experiments. The experimental results from various tests clearly demonstrate the improved performance of the proposed compression/encryption scheme in terms of bandwidth saving and security enhancement.

4.1.2 3D Mesh Hashing

We proposed a efficient methodology for 3D object hashing. The idea is to partition a 3D mesh into s sub-meshes ($s \in \mathbb{N}$), followed by the application of eigen-decomposition to each sub-mesh and derivation of hash values for all sub-meshes. The hash vector of the mesh is a vector containing all hashes of the sub-meshes. The performance of the proposed hashing method was evaluated through extensive experiments which clearly showed excellent resiliency against a slew of attacks.

4.2 Future Research Directions

Several interesting research directions, motivated by this thesis, are discussed below:

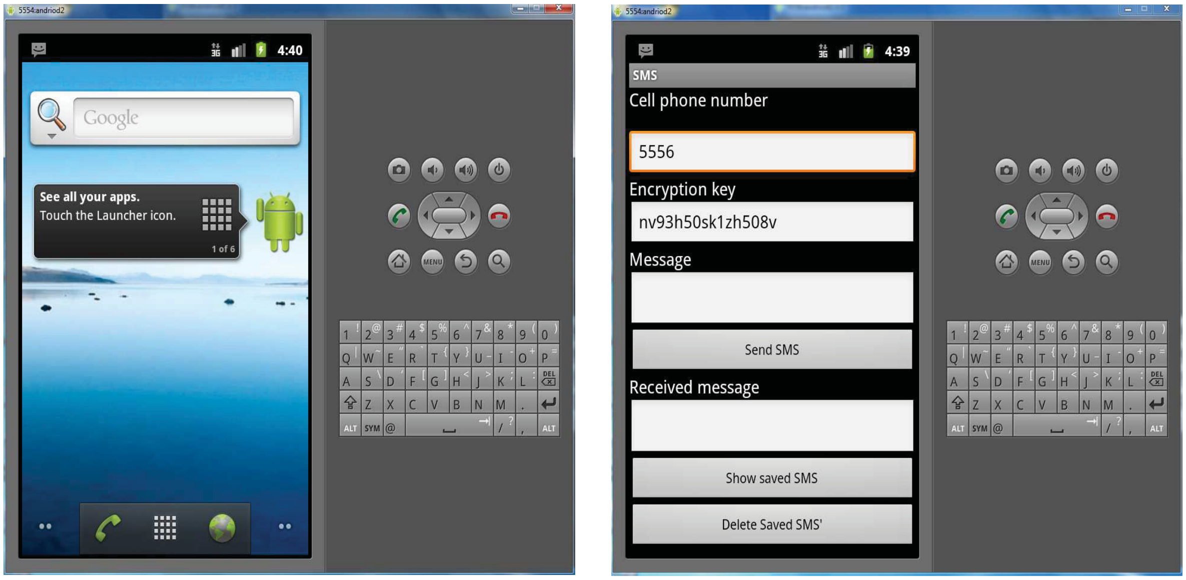
4.2.1 Image Approximation using Shearlets

The approximation and compression scheme introduced in Chapter 2 uses the shearlet transform as its building block. The implementation of the shearlet transform with current algorithms is, however, slower than in the case of wavelets. Recent algorithms proposed in [6] may be used to increase the DST calculation speed. In addition, efficient compression algorithms, such as EZW, may also be employed in the shearlet framework. We would like to extend the compression method of the shearlet transform in order to obtain a higher compression ratio.

4.2.2 Chaotic Approach to Image Encryption

In addition to the satisfactory experimental results obtained by applying the AES encryption method, we can still make use of new image encryption methods, which are specifically designed for image encryption. Recently, the use of chaotic maps for image encryption has gained a lot of interest [14]. These maps can generate a high quality random sequence of numbers that are extremely sensitive to the initial parameter. We can use this sequence of random numbers to create an encryption algorithm with potential applications to mobile computing.

Smartphones and tablets are becoming increasingly popular. They have powerful hardware which can handle multi-tasking processes. Thus, there is an increasing demand for security algorithms and compression methods. We have developed a secure short message service (SMS) software tool for Android devices, which uses AES encryption protocol to send and receive SMS. The program screenshots are shown in Fig. 4.1. This program can be extended to send and receive medical images in a secure way to other devices or PCs via email or multimedia messaging service (MMS). In addition, other encryption methods, such as the chaotic map combined with compression methods, such as the EZW and the shearlet approximation can be implemented in Android devices.



See all your apps.
Touch the Launcher icon.

Cell phone number: 5556

Encryption key: nv93h50sk1zh508v

Message:

Received message:

(a) (b)

FIGURE 4.1: (a) Android emulator main screen, (b) secure SMS program screenshot.

REFERENCES

[1] A. Uhl and A. Pommer, *Image and Video Encryption From Digital Rights Management to Secured Personal Communication*, Springer Science and Business Media, 2005.

[2] A.J. Menezes, P.C. van Oorschot, and S.A. Vanstone, *Handbook of applied cryptography*, CRC Press, 1997.

[3] Y.Q. Shi and H. Sun, *Image and video compression for multimedia engineering, Fundamentals, Algorithms, and Standards*, CRC Press, 2008.

[4] J. Shukla, M. Alwani, and A.K Tiwari, “A survey on lossless image compression methods,” *2nd International Conference on Computer Engineering and Technology (ICCET)*, vol. 6, pp. V6-136-V6-141, 16-18, 2010.

[5] G.M. Davis, “A wavelet-based analysis of fractal image compression,” *IEEE Trans. on Image Processing*, vol. 7, pp. 141-154, 1998.

[6] S. Häuser, “Fast finite shearlet transform: a tutorial,” *arXiv:1202.1773v1*, 2012.

[7] G.R. Easley, D. Labate, and F. Colonna, “Shearlet-based total variation diffusion for denoising,” *IEEE Trans. Image Processing*, vol. 18, no. 2, pp. 260-268, 2009.

[8] P.P. Dang and P.M. Chau, “Image encryption for secure internet multimedia applications,” *IEEE Trans. Consumer Electronics*, vol. 46, no. 3, pp. 395-403, 2002.

[9] L. Wang, C. Laih, H. Tsai, and Nern-Min Huang, “On the hardware design for DES cipher in tamper resistant devices against differential fault analysis,” *Proc. IEEE Int. Symposium on Circuits and Systems*, vol. 2, pp. 697-700, 2000.

[10] Z. Cao and L. Liu, “An attack against DES based on the relationship $L_{i+1} = R_i$,” *Proc. Int. Symposium on Electronic Commerce and Security (ISECS)*, pp. 280-283, 2010.

[11] L. Sauvage, S. Guilley, J.-L. Danger, Y. Mathieu, and M. Nassar, “Successful attack on an FPGA-based WDDL DES cryptoprocessor without place and route constraints,” *Proc. Design, Automation and Test in Europe Conference and Exhibition*, pp. 640-645, 2009.

[12] Z. Jun, L. Jinping, and W. Luqian, “A new compound chaos encryption algorithm for digital images”, *Proc. International Forum on Information Technology and Applications*, vol. 1, pp. 277-279, 2010.

[13] X. Wang, L. Ma, and X. Du, “An encryption method based on dual-chaos system,” *Proc. Int. Conference on Intelligent Networks and Intelligent Systems*, pp. 217-220, 2009.

[14] A. Awad, “A new chaos-based cryptosystem for secure transmitted images,” *IEEE Trans. on Computers*, vol. PP, no. 99, 2011.

[15] Q. Shen-En, M. Bergeron, I. Cunningham, L. Gagnon, and A. Hollinger, “Near lossless data compression onboard a hyperspectral satellite,” *IEEE Trans. on Aerospace and Electronic Systems*, vol. 42, no. 3, pp. 851-866, 2006.

[16] W.-Q Lim, “The discrete shearlet transform: A new directional transform and compactly supported shearlet frames,” *IEEE Trans. Image Processing*, vol. 19, no. 5, pp. 1166-1180, 2010.

[17] D. Labate, W. Lim, G. Kutyniok, and G. Weiss, “Sparse multidimensional representation using shearlets,” *Proc. Wavelets XI*, pp. 254-262, 2005.

[18] G. Kutyniok and T. Saue, “From Wavelets to Shearlets and back again,” *Proc. Approximation Theory XII*, Nashville, TN, pp. 201-209, 2008.

[19] National Institute of Standards and Technology (NIST), *Advanced Encryption Standard (AES)*, Federal Information Processing Standards (FIPS), Publication 197, 2001.

[20] K. Nadehara, M. Ikekawa, and I. Kuroda, “Extended instructions for the AES cryptography and their efficient implementation,” *Proc. 18th IEEE Workshop on Signal Processing Systems*, pp. 152-157, 2004.

[21] M.I. Sobhy and A.E.R. Shehata , “Methods of attacking chaotic encryption and countermeasures,” *Proc. IEEE Int. Conference on Acoustics, Speech, and Signal Processing*, vol. 2, pp. 1001-1004, 2001.

[22] Y. Wang, X. Zheng, and H. Liu, “Robust 3D watermarking based on geometry image, ” *Proc. 4th Int. Conference on Wireless Communications, Networking and Mobile Computing*, pp. 1-4, 2008.

[23] S. Wang, Y. Zhang, Y. Zou, and J. Sun, “A new hash algorithm based on MQ problem and polymorphic cipher,” *Proc. Int. Conference on Information Science and Technology*, pp. 193-198, 2011.

[24] B. Chen and V. Chandran, “Robust image hashing using higher order spectral features,” *Proc. Int. Conference on Digital Image Computing: Techniques and Applications*, pp. 100-104, 2010.

[25] V. Monga and M.K. Mihcak, “Robust image hashing via non-negative matrix factorizations,” *Proc. IEEE Int. Conference on Acoustics, Speech and Signal Processing*, vol. 2, pp. II, 2006.

[26] Y. Hu and X. Niu, “DWT based robust image hashing algorithm,” *Proc. 6th Int. Conference on Networked Computing*, pp. 1-4, 2010.

[27] K. Senel and M.K. Mihak, “A learning framework for robust hashing of face images,” *Proc. 17th IEEE Int. Conference on Image Processing*, pp. 197-200, 2010.

[28] V. Monga and B.L. Evans, “Perceptual image hashing via feature points: performance evaluation and tradeoffs,” *IEEE Trans. on Image Processing*, vol. 15, no. 11, pp. 3452-3465, 2006.

[29] A. Quarteroni, R. Sacco, and F. Saleri, *Numerical Mathematics*, Springer, 2007.

[30] H. Yingbo, X. Yong, C. Tiangping, K. Abed-Meraim, and M. Yongfeng , “Natural power method for fast subspace tracking,” *Eurographics STAR report*, pp. 176-185, 1999.

[31] K. Dookhitram, R. Boojhawon, and M. Bhuruth, “A new method for accelerating Arnoldi algorithms for large scale eigenproblems,” *Mathematics and Computers in Simulation*, vol. 80, no. 2, pp. 387-401, 2009.

[32] P.J. Olver, “Orthogonal bases and the QR algorithm,” *University of Minnesota*, http://www.math.umn.edu/∼olver/aims_/qr.pdf, 2010.

[33] K. Zhang, O. Kaick, and R. Dyer, “Spectral methods for mesh processing and analysis,” *Proc. IEEE Signal Processing Society Workshop*, pp. 122, 2007.

[34] B.Y. wu and K.M. Chao, *Spanning Trees and Optimization problems*, Chapman & Hall/CRC, 2004.

[35] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem,” *Proc. American Mathematical Society*, vol 7, no. 1, pp. 4850, 1956.

[36] F. Nielsen and R. Nock, “On Rényi and Tsallis entropies and divergences for exponential families,” *arXiv:1105.3259*, 2011.

[37] C Tsallis, “Possible Generalization of Boltzman-Gibbs statistics,” *J. Statistical Physics*, vol. 52, pp. 479-487, 1998.

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

[39] M. Ghaderpanah, A. Abbas, and A. Ben Hamza , “Entropic hashing of 3D objects using Laplace-Beltrami operator,” *Proc. 15th IEEE International Conference on Image Processing*, pp. 3104-3107, 2008.

[40] G. Karypis and V. Kumar,“MeTiS: A software package for partitioning unstructured graphs, partitioning meshes, and computing fill-reducing orderings of sparse matrices,” *Version 4.0, University of Minnesota*, 1998.

[41] G.L. Miller, S.H. Teng, W. Thurston, and S.A. Vavasis, “Automatic mesh partitioning,” *IMA Volumes in Mathematics and its Application*, vol. 56, 1993.

[42] G.L. Miller, S.H. Teng, W. Thurston, and S.A. Vavasis, “Geometric separators for finite-element meshes,” *SIAM J. Scientific Computing,* vol. 19, no. 2, pp. 364-386, 1998.

[43] “3DS MAX Tutorials website,” *http://www.3dmax-tutorials.com*.

[44] “Autodesk website,” *http://www.autodesk.com*.

[45] S. Chen, T. Hwang, and W. Lin , “Randomness Enhancement Using Digitalized Modified Logistic Map,” *IEEE Trans. Circuits and Systems II*, vol. 57, no. 12, pp. 996-1000, 2010.

[46] T. Podoba, J. Giesl, and K. Vlcek, “Image encryption in wavelet domain based on chaotic maps,” *Proc. Int. Congress on Image and Signal Processing*, 2009.

[47] S. Jianbo and J. Malik, “Normalized cuts and image segmentation,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.