ROBUST DIGITAL WATERMARKING TECHNIQUES FOR MULTIMEDIA PROTECTION

Emad Eddien Awad Abdallah

A thesis

in

The Department

of

Computer Science

Presented in Partial Fulfillment of the Requirements

For the Degree of Doctor of Philosophy

Concordia University

Montréal, Québec, Canada

March 2009

Your file Votre référence
ISBN: 978-0-494-63393-9
Our file Notre référence
ISBN: 978-0-494-63393-9

Abstract

ROBUST DIGITAL WATERMARKING TECHNIQUES FOR MULTIMEDIA PROTECTION

Emad Eddien Awad Abdallah, Ph.D.
Concordia University, 2009

The growing problem of the unauthorized reproduction of digital multimedia data such as movies, television broadcasts, and similar digital products has triggered worldwide efforts to identify and protect multimedia contents. Digital watermarking technology provides law enforcement officials with a forensic tool for tracing and catching pirates. Watermarking refers to the process of adding a structure called a watermark to an original data object, which includes digital images, video, audio, maps, text messages, and 3D graphics. Such a watermark can be used for several purposes including copyright protection, fingerprinting, copy protection, broadcast monitoring, data authentication, indexing, and medical safety.

The proposed thesis addresses the problem of multimedia protection and consists of three parts. In the first part, we propose new image watermarking algorithms that are robust against a wide range of intentional and geometric attacks, flexible in data embedding, and computationally fast. The core idea behind our proposed watermarking schemes is to use transforms that have different properties which can effectively match various aspects of the signal's frequencies. We embed the watermark many times in all the frequencies to provide better robustness against attacks and increase the difficulty of destroying the watermark.

The second part of the thesis is devoted to a joint exploitation of the geometry and topology of 3D objects and its subsequent application to 3D watermarking. The key idea consists of capturing the geometric structure of a 3D mesh in the spectral domain by computing the eigen-decomposition of the mesh Laplacian matrix. We also use the fact that the global shape features of a 3D model may be reconstructed using small

low-frequency spectral coefficients. The eigen-analysis of the mesh Laplacian matrix is, however, prohibitively expensive. To lift this limitation, we first partition the 3D mesh into smaller 3D sub-meshes, and then we repeat the watermark embedding process as much as possible in the spectral coefficients of the compressed 3D sub-meshes. The visual error of the watermarked 3D model is evaluated by computing a nonlinear visual error metric between the original 3D model and the watermarked model obtained by our proposed algorithm.

The third part of the thesis is devoted to video watermarking. We propose robust, hybrid scene-based MPEG video watermarking techniques based on a high-order tensor singular value decomposition of the video image sequences. The key idea behind our approaches is to use the scene change analysis to embed the watermark repeatedly in a fixed number of the intra-frames. These intra-frames are represented as 3D tensors with two dimensions in space and one dimension in time. We embed the watermark information in the singular values of these high-order tensors, which have good stability and represent the video properties. Illustration of numerical experiments with synthetic and real data are provided to demonstrate the potential and the much improved performance of the proposed algorithms in multimedia watermarking.

Acknowledgments

I would like to express my appreciation and gratitude to my advisers, Professor A. Ben Hamza and Professor Prabir Bhattacharya, for their support and guidance during my graduate studies. I owe them a lot for their feedback during the development of this work. Their confidence in my abilities as a researcher is greatly appreciated. They polished my technical writing through numerous revisions. Most importantly, they taught me to question my intuition and search for better answers.

I wish to acknowledge all my committee members for giving me the privilege of having them on my doctoral committee and for valuable suggestions during my Ph.D seminar and my proposal defence and this dissertation. Special thanks to laboratory mates Sohail, Kaushik, Mahdi, Lilybert.

I would like to thank my friends back in Jordan, Majdi, Shamsi, and Amjad, who constantly tried to make me feel not so away from home. I would also like to thank my friends here in Montreal who gave me very beautiful memories: Alaa, Edi, Tarek, George, Eva, Eslam, Adnan, Faraz.

Finally, I wish to express my deepest gratitude to my father, my mother, my brothers Alaa and Mohammad, and my sister Hana'a for their encouragement and love. This achievement is as much theirs as it is mine.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Currently, multimedia files are represented in digital form because the digital representation offers the advantages of portability, efficiency, and perfect reproduction. With the increased use of digital multimedia on the web and the proliferation of inexpensive storage devices, it has become easy to make perfect counterfeit copies. The increase of such piracy triggered the need to protect digital multimedia from unauthorized duplication. Traditionally, encryption and control access techniques were employed to protect the ownership of media. These techniques, however, do not protect against unauthorized copying after the media have been successfully transmitted and decrypted. Digital watermarking provides a solution to the problem of pirating digital material by hiding the owner's information in the multimedia data [21, 38].

Here is a typical scenario of the creation and detection of a watermark: An owner starts with an original document and a secret watermark, which is just a vector of m random numbers, for example. The owner embeds the watermark in the document data. The owner hides the original document and the original watermark and publishes the watermarked document. A pirate, takes a copy of the watermarked document, modifies it, or "attacks" it, and publishes the attacked document claiming that it is his or her own. The owner obtains a copy of the attacked document and

applies his or her watermark extraction algorithm. He or she then gets a new vector of numbers to compare with the one he or she embedded. If the two vectors are very similar, they would give clear evidence that the suspected document came from the one which the owner watermarked. An example of such a scenario is provided in Figure 1.

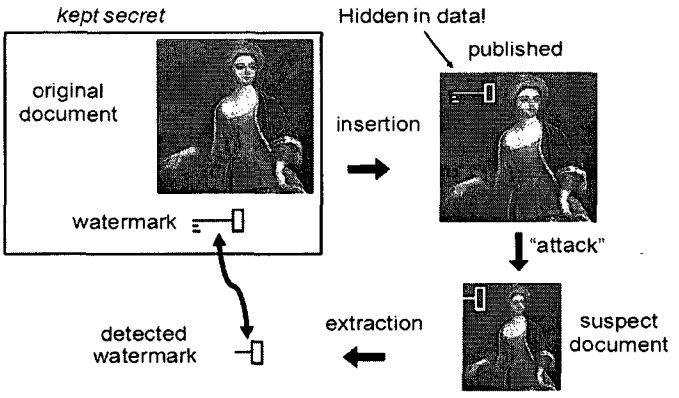


Figure 1: Watermark embedding and extraction model

Watermarking and steganography are often mentioned together, and indeed, robust watermarking techniques can be used in some steganographic applications. There are three fundamental differences between watermarking and steganography: 1) Robustness may be a concern in some steganography applications, but generally steganographic methods give priority to the capacity of information-carrying mediums, while watermarking techniques give priority to robustness against attacks [21, 52, 65, 73]. 2) The information hidden by a watermarking system is always associated to the digital object being protected or to its owner, while steganographic systems just hide any digital information [43]. 3) Another important difference is that the watermark requires the preservation of the quality of the cover, while in steganography the cover signal only serves as a carrier. If the quality of the cover is affected, it is not a major concern (unlike in watermarking). Cryptographic technology can be used to prevent unauthorized access to the digital content. However, cryptography is limited with

regards to protecting copyright because once the digital content gets decrypted there is nothing to prevent unauthorized user from illegally duplicating it.

In this thesis, we propose robust watermarking schemes for still images, 3D models and video image sequences. Our proposed schemes modify original multimedia documents in the transform domain. The core idea behind our proposed image approach is to use the fast Hadamard transform (FHT) and the discrete wavelet transform (DWT). Different transforms have different properties that can effectively match various aspects of a signal's frequencies. The majority of this thesis is devoted to 3D mesh watermarking. The key idea consists of using the mesh Laplacian matrix to embed a watermark in the spectral coefficients of a compressed 3D mesh. We also propose a new hybrid methodology for MPEG video watermarking. The central idea is based on a high-order tensor singular value decomposition (TSVD) and DWT.

1.1 Framework and Motivation

The primary objectives of this dissertation are to design watermarking techniques for images, 3D models, and video image sequences. These techniques should be computationally inexpensive, imperceptible, and robust against attacks.

1.1.1 Image watermarking

Image watermarking refers to the process of adding a hidden structure, called a "watermark", that carries information about either the owner of the cover or the recipient of the original image [21, 38, 58]. The challenge lies in providing good robustness against attacks, increasing the difficulty of destroying a watermark from an image, and providing a high visual quality of the watermarked image.

The problem of image watermarking has been addressed using a number of different techniques. These techniques can be divided into two main categories according

to the embedding domain of the cover image: spatial domain methods and transform domain methods. The spatial domain methods are the earliest and simplest watermarking techniques. However, they have a low information-hiding capacity, and the watermark can be easily erased by lossy image compression. On the other hand, the transform domain approaches [20, 56] insert the watermark into the transform coefficients of the image cover, embedding more information and yielding more robustness against watermarking attacks.

Most image watermarking methods use one transform to embed the watermark. Usually these methods have either good robustness against attacks or good watermark imperceptibility. Motivated by the need for more robustness against attacks, better visual imperceptibility, and the computational simplicity of the FHT, we propose robust, imperceptible image watermarking methods.

1.1.2 3D mesh watermarking

In 3D games, animated movies, virtual museums, medical imaging, computer aided-design (CAD), physical simulations, and other interactive environments, 3D models are commonly used interactively. Many of these models are valuable and require protection from misuse such as unlawful exhibition. The solution is 3D watermarking.

3D mesh watermarking is a relatively new area compared to 2D watermarking [21]. It has received less attention partly because the technology used for the image and video analysis cannot be easily adapted to 3D objects, which can be represented in several ways, including voxels, NURBS, and polygonal meshes. Early algorithms on 3D watermarking [12, 36, 68] consist of embedding the watermark information directly by modifying either the 3D mesh geometry or the topology of the triangles. These methods are usually simple and require low computational costs. However, they are not robust sufficiently to counter different types of attacks. Recently, several watermarking algorithms in the frequency domain have been proposed for 3D meshes

[70, 74, 75]. They are mainly based on multi-resolution mesh analysis (spectral decomposition and wavelet transform) and show good resistance against attacks.

Motivated by the need for more robustness against attacks, we use the spectral compression of mesh geometry introduced in [41]. We also use the fact that the low-frequency coefficients represent the global shape features of a 3D mesh. The rough approximation of the model can be reconstructed using small, low-frequency spectral coefficients. The goal of our proposed scheme is to design a robust, invisible, and non-removable 3D watermarking scheme by embedding the watermark in the low-frequency components. We carry this out by repeating the watermark embedding process as much as possible in the spectral coefficients of the compressed 3D mesh. The number of spectral coefficients of the compressed mesh is exactly the same as the number of spectral coefficients of the original 3D mesh without compression. As a result, we obtain the maximum number of watermark repetitions (maximum robustness against noise addition attack). Moreover, all the watermarks are embedded in the low frequency components that are used during the mesh compression stage. Embedding the watermarks in the low-frequency components guarantees the most robustness against smoothing and compression attacks.

1.1.3 Video watermarking

The use of digital video applications (such as video-conferencing, digital television, digital cinema, distance learning, videophone, and video-on-demand) has grown rapidly over the last few years. Today, it is much easier for digital data owners to transfer their videos over the internet, and hence the data could be perfectly duplicated and rapidly redistributed. Thus the importance of copyright protection for videos has became more critical.

Video watermarking schemes need to meet some other challenges not present in

image watermarking. These challenges include the large volume of inherently redundant data between frames, the unbalance between motion and motionless regions [78], and the real-time requirements in video broadcasting. These make video signals highly susceptible to pirate attacks, including frame averaging, frame dropping, frame swapping, and statistical analysis [14].

Motivated by the good performance of 2D image watermarking techniques proposed in [27, 56] and the multilinear generalization of the singular value decomposition, we present a robust, hybrid scene-based MPEG video watermarking technique based on a high-order tensor singular value decomposition and discrete wavelet transform. The key idea is to apply the TSVD to the four wavelet sub-bands of the video frames viewed as a 3D tensor with two dimensions in space and one dimension in time.

1.2 Thesis Overview and Contributions

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

- ❑ We performed a complete survey of the current watermarking technologies for images, 3D models and videos. We noticed that none of the current watermarking schemes can resist all common attacks.

- ❑ We designed and implemented two hybrid invisible and robust image watermarking techniques [1, 2] using two transforms that have different properties.

- ❑ We proposed two improved schemes for 3D mesh watermarking [3, 4] using spectral mesh analysis to embed the watermark repeatedly in the low-frequency components.

- ❑ We proposed two video watermarking schemes [6] using high order tensor singular value decomposition. These new schemes are robust against all the common

attacks in video processing.

The organization of this thesis is as follows:

- The **Background** chapter contains a brief review of essential concepts and definitions that we will refer to throughout the thesis. We present a short summary of the basic watermarking principles, schemes, types, attacks, and applications.

- **Robust and efficient image watermarking schemes**: In chapter 3 we will propose two fast image watermarking techniques using FHT, DWT, and singular value decomposition (SVD). In the first technique, we introduce a high-rate embedding of watermarks into digital images using FHT and SVD. The idea is to embed the singular values (SVs) of the watermark image in the direct current (DC) components of the FHT blocks of the cover image. In the second algorithm, we propose an improved technique using FHT and DWT. The key idea is to encode the SVs of the watermark image after applying the FHT to small blocks computed from the four DWT sub-bands. The proposed techniques improve the data embedding system, watermark imperceptibility, and are resistant to a wide range of intentional attacks.

- **Spectral graph-theoretic approach to 3D mesh watermarking**: In chapter 4 we propose a robust and imperceptible spectral watermarking method for high-rate embedding of a watermark into 3D polygonal meshes. The key idea is to encode a watermark vector repeatedly into the spectral coefficients of the compressed 3D mesh. The main attractive features of this approach are simplicity, flexibility in data embedding capacity, and fast implementation. We demonstrate, through experimental results, the power of the proposed technique in improving robustness and also in preserving 3D mesh quality.

- **3D watermarking technique using nonnegative matrix factorization**: In chapter 5 we propose a simple and robust 3D mesh watermarking methodology for embedding a watermark in the transform domain. The core idea behind our technique is to encode a sequence of random numbers after applying the nonnegative matrix factorization (NMF) to small blocks computed from the spectral matrix of the 3D mesh.

- **Robust video watermarking techniques using tensor singular value decomposition**: In chapter 6 we present robust, hybrid MPEG video watermarking techniques based on high-order tensor singular value decomposition (TSVD). Unlike previous methods where each video frame is marked separately, our proposed techniques use high-order tensor decomposition of videos. The key idea behind our approaches is to use the scene-change analysis to embed the watermark repeatedly in a fixed number of the intra-frames. These intra-frames are represented as a 3D tensor. Then we modify the singular values of the 3D tensor. The main attractive features of these approaches are simplicity and robustness. The experimental results show the robustness of the proposed schemes against the most common 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 work performed in this thesis.

Chapter 2

Background

This thesis presents new copyright protection algorithms for images, three-dimensional models, and video image sequences. The following background material is presented to provide context for this work. First, basic watermarking principles along with watermarking requirements are presented. Then, we present a discussion of the various watermarking applications.

2.1 Basic Watermarking Principles

Digital watermarking refers to the process of embedding imperceptible information called a digital watermark into a cover multimedia object so that the information may be detected or extracted later for security purposes. A digital watermark carries information about either the owner of the cover or the recipient, and can be used to identify the owner of a digital media and also to prevent unauthorized distribution [21]. The owner of the special key is the only one who can extract the watermark. Ultimately, a good watermark should satisfy some watermarking requirements, such as invisibility and robustness against attacks [9, 21].

2.1.1 Watermarking requirements

In general, good watermarking techniques require a few characteristics. These characteristics apply to watermarking schemes for all kinds of data that can be watermarked, such as audio, image, video, formatted text, and 3D models. The priority of these characteristics differs with the purpose of the watermarking [9, 21]. For example, in copyright protection, the watermarking system should be as robust as possible (See P11), but for authentication applications, robustness is not required. Unfortunately, the invisibility and robustness requirements create a contradictory situation for watermarking development [9, 20]. The main watermarking characteristics are:

Imperceptibility or invisibility

The modifications caused by watermark embedding should not affect or degrade the perceptual quality of the host media. This implies that some sort of perceptibility criteria should be used, not only in the process of designing the watermark but also in quantifying the distortion. In other words, the difference between the watermarked and the original documents should be unnoticeable to a human observer [43].

One way to quantify distortion is the mean-square error. The mean-square error between any signals S and $\widetilde{S}$ is defined as

$$MSE(S, \widetilde{S}) = \frac{1}{m^2} \sum_{i=1}^{m} \sum_{j=1}^{m} \|S_{ij} - \widetilde{S}_{ij}\|^2 \tag{1}$$

when S and $\widetilde{S}$ are identical, then $MSE(S, \widetilde{S}) = 0$. A related distortion measure is the peak signal-to-noise ratio (PSNR), measured in decibels (dB). The PSNR [64] is defined as follows:

$$PSNR(S, \widetilde{S}) = 20 \log_{10}\Big(\frac{MAX_i}{\sqrt{MSE(S, \widetilde{S})}}\Big) \tag{2}$$

where $MAX_i = max\{\widetilde{S}_{ij}, 1 \leq i, j \leq m\}$. The higher the $PSNR(S, \widetilde{S})$, the less distortion between S and $\widetilde{S}$. If the signals are identical, then $PSNR(S, \widetilde{S}) = \infty$.

The MSE and PSNR provide a way of quantifying the true distortion between two signals. However, true distortion is not as significant in watermarking as perceptual distortion.

When examining the perceptibility of distortion, the spatial and temporal distribution of the distortion is as significant as the total power of the distortion. For example, distortion that alternates rapidly in successive frames of video may appear to flicker or shimmer to an observer. This can be extremely distracting, even when the total amount of the distortion is small. On the other hand, relatively large amounts of distortion may be unnoticed in textured or busy areas of an image. In general, many factor affects affect human perception of distortion [52, 81], including the signal characteristics, distortion characteristic, and the environmental or viewing conditions.

Robustness

When given a watermarked document, an unauthorized party should not be able to destroy the watermark without at the same time rendering the document useless. To ensure robustness, the watermark information is usually redundantly distributed over many samples (or features) of the cover media, thus providing a global robustness. This means that the watermark can be recovered from a small fraction of the watermarked media element. Obviously, the watermark recovery is more robust if more of the watermarked data is available through the recovery process. For practical reasons, systems must find a compromise between robustness and the competing requirements of invisibility and information rate.

Attacks: The watermark signal may be attacked [52, 72, 83, 84] before examination by the watermark detector. An attack is any process that may remove the embedded watermark, increase the difficulty in detecting the watermark, or weaken the security of the watermark. The watermarked signal may be attacked multiple times. Removal attacks are attacks that damage or destroy an embedded watermark.

Removal attacks include filtering, lossy compression, noise addition, and noise removal. Some removal attacks such as lossy compression do not specifically target the embedded watermark but may incidentally destroy or damage it [52]. Another method of attack is to increase the difficulty of detecting the watermark. These detection-disabling attacks do not remove the watermark, but obscure it from the watermark detector. One type of detection-disabling attack is the mosaic attack [72]. In the mosaic attack, a large watermarked signal is partitioned into smaller signals, each of which is sufficiently small such that the watermark detector will fail to detect the watermark when examining the small signals alone.

The context of an application is an important consideration when discussing attacks [52]. While many attacks against watermarks have been described in [43], not all of the attacks are necessarily significant to a particular application. For example, the application may anticipate certain kinds of attacks and require an embedded watermark to be robust against these attacks. Robustness against other attacks may have less or no importance, especially if the attacks are not likely to occur in the application.

Depending on the application and watermarking requirements, the list of distortions and attacks to be considered includes [43]: signal enhancement (sharpening, contrast enhancement, color correction, gamma correction), additive and multiplicative noise (Gaussian, uniform, speckle, mosquito), linear filtering (low-pass, high-pass filtering), nonlinear filtering (median filtering, morphological filtering), lossy compression (images: JPEG, video: MPEG-2, MPEG-4), affine transforms (translation, rotation, scaling, shearing), data reduction (cropping, clipping, histogram modification), data composition (logo insertion, scene composition), multiple watermarking, collusion attacks, statistical averaging, and mosaic attacks.

Security

In general, watermarking systems use one or more cryptographically secure keys to ensure security against manipulation and erasure of the watermark. As a result, knowing the algorithms for embedding and extracting does not help unauthorized parties detect or remove the watermark.

Capacity

Capacity is defined as the data payload, or the number of information bits embedded into the original media. For applications like copy control, the presence or the absence of one bit watermark is generally enough, but for other applications, like fingerprinting or copyright protection, much more data needs to be added.

2.1.2 General watermarking scheme

All watermarking methods consist of two main components: the embedding system and the watermark extraction or recovery system. Simple watermark embedding and extraction models are shown in Figure 2. The watermark embedding system has as inputs the cover media $\mathbf{C}$, the key, and the watermark symbol $\mathbf{W}$. The output of the embedding algorithm is the watermarked media $\widehat{\mathbf{C}}$. The watermark extraction process has as inputs the watermarked media, the secret key or the public key, the original cover or the original watermark or both. The output of the extraction algorithm is either the suspect watermark or some kind of confidence measure. The watermark detector decides whether a watermark is present. If so, the extracted watermark is then compared with the original watermark to measure their differences.

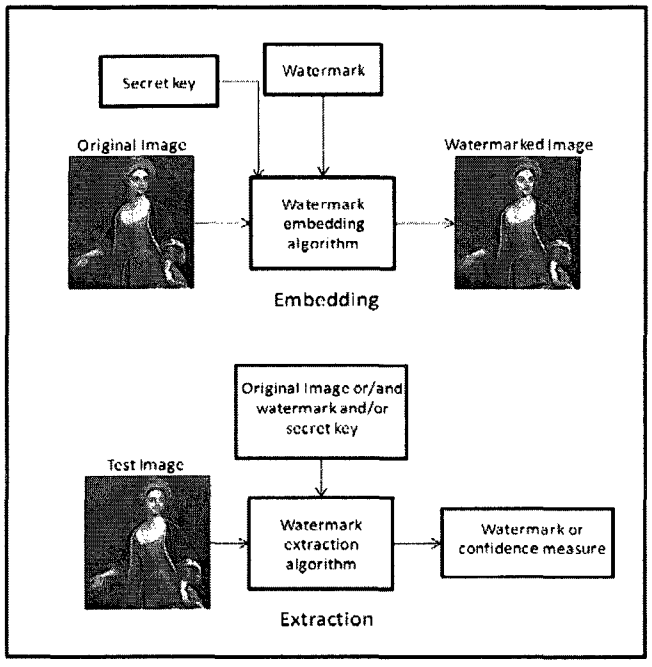


Figure 2: Watermarking embedding and extraction models

2.1.3 Types of watermarking systems

A watermarking system may be classified according to whether or not the original media and the secret key information are needed in the watermark extraction algorithm for private, semi-private, and public watermarking systems. According to the visibility of the watermark, watermarking schemes may be classified in two types: visible techniques and invisible techniques. According to their robustness against attacks, watermarking techniques may be classified as fragile, semi-fragile, and robust. Finally, watermarking systems may be classified according to the embedding domain of the cover media to the spatial domain methods and the transform domain methods [9, 21, 38, 65]. Table 1 shows an overview of the watermarking classification according to several criteria.

Visible watermark

In visible watermarking techniques, the watermark data is embedded in the cover in a way that the watermark is intentionally perceptible to a human observer [62]. The primary purpose of this technique is to prevent unauthorized commercial use of the media. At the same time, the observer knows the institution that owns the material. These watermarks commonly take the form of a logo.

Invisible watermark

In contrast to the visible watermark, the invisible watermark is embedded into the host contents such that the watermark is not perceptible to an observer, but may be extracted or detected by a computer program [56, 58]. The invisible watermark is used to make an assertion about the ownership of an object.

Private watermarking system

This is also called non-blind watermarking [20, 56]. For these types of systems, the cover, the secret key, and the watermarked media are needed for watermark detection. In some private watermarking systems, a copy of the embedded watermark symbols are also needed.

Public watermarking system

This is also referred to as blind watermarking [60, 86]. This type of watermarking technique requires only the secret key during the detection of the watermark information. The secret information is usually a pseudo-random sequence generated using a key as the seed.

Semi-private watermarking system

This is also called semi-blind watermarking [24]. This type of watermarking technique does not use the original cover; it requires the secret keys and a copy of the embedded watermark data.

Watermarking in the spatial domain

The watermark is encoded by modifying pixels directly [15, 38, 39]. The spatial domain methods are the earliest and simplest watermarking techniques, and often require low computational costs. However, they have a low information-hiding capacity, and the watermark can also be easily erased by lossy image compression.

Watermarking in the transform domain

The transform domain approach maps the multimedia data onto different mathematical space via transformation equations. Most discrete transforms map the cover data from the spatial domain to the frequency domain (spectral domain). The idea is to insert the watermark into the transform coefficients of the cover data, yielding more embedded information and providing more robustness against watermarking attacks. Many different ideas have been presented, most of them originating from [20].

2.2 Watermarking Applications

Watermarking applications can be found in different areas [9, 21, 43, 53, 65, 81], and may be categorized as follows:

Watermarking for copy protection or usage control

A desirable feature in multimedia distribution systems is the existence of a copy protection mechanism that disallows unauthorized copying of the media. A device

Table 1: Classification of watermarking schemes according to several criteria

Classification	**Contents**	**Brief description**
Domain type	Spatial	Pixels are modified directly to embed the watermark.
	Transform domain	Transform coefficients are modified to embed the watermark.
Perceptibility of watermark	Visible	Watermark is intentionally perceptible to the human observer.
	Invisible	Watermark is not perceptible to the observer.
Information type	Non-blind	Both original image and secret key are needed for extraction process.
	Semi-blind	Watermark and the secret key are needed for extraction process.
	Blind	Only secret key is needed for extraction process.
Robustness of watermark	Robust	Copyright information survives after manipulations that intended to remove the embedded information.
	Semi-fragile	Insensitive to compression but sensitive to attacks that alter the media information. Used to illegally tamper with the image rather than to verify its ownership.
	Fragile	Embedded with very low robustness; used for authentication applications.

that obeys the copy protection protocol detects the watermark and then disallows the creation of copies. Some copy protection schemes allow a user to create a single copy but restrict the user from making additional copies from a copy. An example is the DVD system in which the data contains copy information embedded as a watermark. A compliant DVD player is not allowed to playback or copy data that carries a "copy never" watermark. Data that carries a "copy once" watermark may be copied, but no consecutive copies are allowed to be made from that copy. Using watermarks in this manner requires cooperation from the recording devices to detect the watermark and prevent unauthorized copying.

Fingerprinting for transaction tracing

The owner personalizes each copy of the content by embedding a watermark into the copy. The embedded watermark identifies the user who has custody of that copy. Any subsequent digital copies made of the watermarked content will also be watermarked. This is useful in monitoring or tracing illegally produced copies of the data that may circulate. It enables the owner of the intellectual property to trace customers who have broken their license agreement. This type of application is sometimes referred to as fingerprinting. Watermarking for fingerprinting applications requires high robustness against standard data processing attacks.

An example of fingerprinting occurs in a digital cinema environment where films are distributed to cinemas in digital format. Even though digital distribution of films could be more flexible, more efficient, and less expensive, film producers and distributers are slow to adopt it because of their concern over the potential loss of revenues caused by the illegal copying and redistribution of films. Now, if each movie theater receives a unique identifiable copy of a film, then it should be possible to associate any illegal copies with the associated cinema, and initiate appropriate legal action.

Watermarking for copyright protection

One of the traditional applications of watermarking is copyright protection. In copyright watermarking, the embedded watermark encodes ownership information such as the identity of the owner and the copyright date. Detecting the invisible watermark provides the content owner with additional evidence of ownership. This might help prevent other parties from claiming the copyright in the data. Copyright protection requires a high level of robustness. The driving force for this application is the World Wide Web, which contains millions of freely available media that the rightful owners want to protect.

Watermarking for image authentication

In authentication applications, the objective is not to protect the contents from being copied or stolen, but rather to authenticate the cover media and ensure its integrity. This can be achieved through fragile watermarks [43] that have a low robustness against certain modifications, like compression. If the watermark is detected perfectly, the data is genuine; otherwise, the data may have been corrupted and cannot be considered. Among all possible watermarking applications, authentication watermarks require the lowest level of robustness.

Medical safety

Embedding the watermark and the patient's information in medical images has the potential to increase the confidentiality of medical information as well as the security.

Smart content

An embedded watermark may be used in conjunction with devices to provide additional functionalities or services that benefit the user [53]. For example, a watermark

embedded invisibly in a music video may display a link to the artist's Web site, which allows a user to purchase the artist's other work.

Broadcast monitoring

Watermarks can be embedded in any kind of data to be widely broadcast on a network and help the automated identification of broadcasted programs. For example, by embedding watermarks in commercial advertisements and television broadcasts, an automated monitoring system can verify whether the advertisements are broadcast as contracted.

Each application has its own special requirements with regard to robustness, security, imperceptibility, and capacity. For example, when digital watermarks are used for copyright protection, the need for robustness and imperceptibility is obvious, while the amount of data to be embedded is of only marginal interest. The specific requirements of each watermarking technique vary with the application [61, 90]. There is no universal watermarking technique that satisfies all requirements of all applications.

2.3 Discrete Image Transformation and Matrix Decomposition

The transform domain has been extensively studied in the context of image coding and compression, and much research can be applied to digital watermarking [63]. The theory of image coding maintains that the color-neighboring pixels are highly correlated. Mapping into a specific transform domain, such as discrete wavelet transform, serves two purposes. It de-correlates the original sample values, and it concentrates the energy of the original signal into just a few coefficients. For example, when an image is mapped into the frequency domain, the energy is concentrated in the low-index terms, which are very large compared to the high-index terms. This means

that an image is dominated by the low-frequency components. These low-frequency components represent the overall shapes and outlines of features in the image, as well as their luminance and contrast characteristics. As an example, a typical image might contain 95% of the energy of the lowest 5% of the frequency components.

2.3.1 Discrete wavelet transform

The DWT [76] provides a number of powerful image-processing algorithms, including noise reduction, edge detection, and compression. Recently, with the standardization of JPEG-2000 and the decision to use wavelet-based image compression, watermarking techniques operating DWTs have become more attractive to the watermarking research community [24, 27].

The DWT is computed by successive low-pass and high-pass filtering of the discrete time-domain signal. Its significance is in the manner that it connects the continuous-time multiresolution to the discrete-time filters. At each level, the high-pass filter produces detailed information, while the low-pass filter associated with scaling function produces coarse approximations. To use DWT for image processing, we use a 2D version of the analysis and synthesis filter banks. In the 2D case, the 1D analysis filter bank is first applied to the columns of the image and then applied to the rows. If the image has m rows and m columns, then, after applying the 2D analysis filter bank, four sub-band images are obtained (LL, LH, HL, and HH), each having $m/2$ rows and $m/2$ columns.

2.3.2 Fast Hadamard transform

The 2D Hadamard transform has been used with great success for image compression and image watermarking. Unlike the other well-known transforms, such as the discrete Fourier transform (DFT) and the discrete cosine transform (DCT), the elements of the basis vectors of the Hadamard transform take only the binary values +1 and

-1. Hence, the FHT is well suited for digital image-processing applications where computational simplicity is required.

Let $\boldsymbol{C}$ be the original image of size $m \times m$. The 2D Hadamard transform of $\boldsymbol{C}$ is given by

$$\widehat{\boldsymbol{C}} = (1/m)\boldsymbol{H}\boldsymbol{C}\boldsymbol{H} \tag{3}$$

where $\boldsymbol{H}$ is a Hadamard matrix of order $m = 2^k$ (k is an integer), and with entries $\{-1, +1\}$. The Hadamard matrix $\boldsymbol{H}$ has mutually orthogonal rows or columns, and satisfies $\boldsymbol{H}\boldsymbol{H}^T = m\boldsymbol{I}_m$, where $\boldsymbol{I}_m$ is the identity matrix. Hence, the original image may be recovered using

$$\boldsymbol{C} = (1/m)\boldsymbol{H}\widehat{\boldsymbol{C}}\boldsymbol{H} \tag{4}$$

Figure 3 shows an example of a 2D image with its FHT result. Furthermore, the Hadamard matrix of order m may be generated from the Hadamard matrix of order $m/2$ using the Kronecker product property $\boldsymbol{H}_m = \boldsymbol{H}_2 \otimes \boldsymbol{H}_{m/2}$, where

$$\boldsymbol{H}_2 = \begin{bmatrix} +1 & +1 \\ +1 & -1 \end{bmatrix}$$

is the Hadamard matrix of order $m = 2$. Consequently $\boldsymbol{H}_4$ becomes

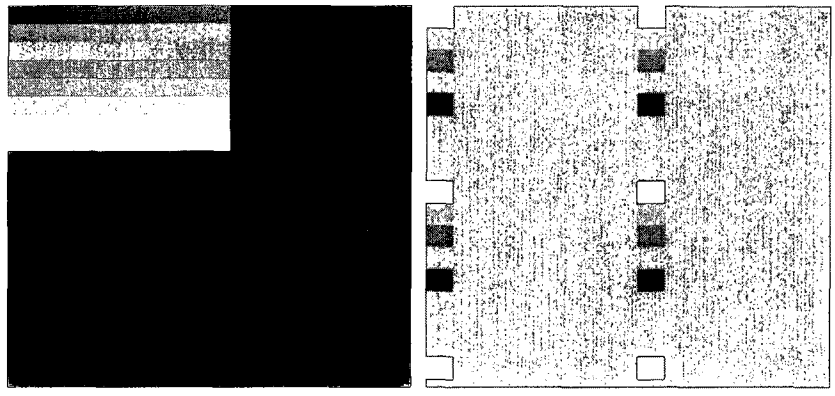
$$\boldsymbol{H}_4 = \begin{bmatrix} +H_2 & +H_2 \\ +H_2 & -H_2 \end{bmatrix} = \begin{bmatrix} +1 & +1 & +1 & 1 \\ +1 & -1 & +1 & -1 \\ +1 & +1 & -1 & -1 \\ +1 & -1 & -1 & +1 \end{bmatrix}$$

2.3.3 Singular value decomposition

The SVD of a 2D image $\boldsymbol{C}$ of size $m \times n$ is given by

$$\boldsymbol{C} = \boldsymbol{U}\boldsymbol{\Sigma}\boldsymbol{V}^T \tag{5}$$

where $\boldsymbol{U}$ is an orthogonal matrix ($\boldsymbol{U}^T\boldsymbol{U} = \boldsymbol{I}$), $\boldsymbol{\Sigma}$ =diag(λ_i) is a diagonal matrix of singular values λ_i, $1 \leq i \leq r$, arranged in decreasing order, and $\boldsymbol{V}$ is an orthogonal



(a) (b)

Figure 3: An example of the fast Hadamard transform: (a) original image, (b) transformed image

matrix ($\mathbb{V}^T\mathbb{V} = \mathbb{I}$) as depicted in Figure 4. The columns of $\mathbb{U}$ are the left singular vectors of the image, whereas the columns of $\mathbb{V}$ are the right singular vectors of the image $\mathbb{C}$.

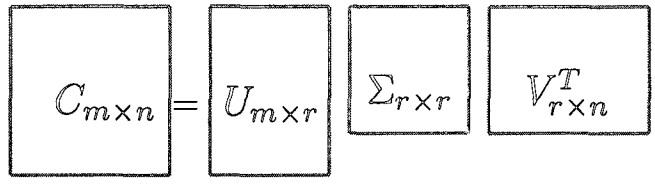


Figure 4: Illustration of the SVD approximation

Chapter 3

Image Watermarking Schemes using FHT

In this chapter, we present two robust image watermarking schemes using FHT, DWT, and SVD. Different transforms have different properties that can effectively match various aspects of the signals frequencies. The core idea of the first approach [2] is embedding the singular values of the watermark image in the DC components of the FHT blocks of the cover image. The second technique is an improved algorithm inspired by [2] that employs the DWT to modify the cover image. The algorithm consists of four main steps: (1) the original image is decomposed into four sub-bands, (2) the four sub-bands are divided into blocks; (3) FHT is applied to each block; and (4) the SVD is applied to the watermark image prior to distributing the singular values over the DC components of the transformed blocks.

3.1 Image Watermarking Overview

Recently, unitary transformations have been widely used for data embedding including the DCT [20, 46], DFT [23, 26], FHT [1, 2, 32] and DWT [27, 79, 85]. Watermarking using singular-value decomposition and its variants has been proposed

[17, 27, 28, 56]. The main idea of these approaches is to find the SVD of a cover image and then modify its singular values to embed the watermark. In [27], a hybrid non-blind watermarking scheme based on the SVD and the DWT was proposed. This method consists of decomposing the cover image into four transformed sub-bands, then the SVD is applied to each band, followed by modifying the singular values of the transformed sub-bands with the singular values of the visual watermark. This modification in all frequencies provides more robustness to different attacks. Another SVD-block-based watermarking scheme was proposed in [28], where the watermark embedding is done in two layers. In the first layer, the cover image is divided into small blocks and the singular values of the watermark are embedded in those blocks. In the second layer, the cover image is used as a single block to embed the whole watermark. In addition to the limited robustness to some attacks, the main weakness of the SVD-based techniques is that the SVD produces low-rank basis functions that do not respect the nonnegativity of the cover image.

Our proposed techniques use the DWT, FHT, and SVD. Extensive experiments are performed to demonstrate the potential and the much-improved performance of the proposed methods in comparison to other existing watermarking methods.

The remainder of this chapter is organized as follows. In Section 3.2, we briefly review some works that are closely related to our proposed schemes. In Section 3.3, we introduce the proposed methodologies, watermark embedding and extraction algorithms. In Section 3.4, we present some experimental results to demonstrate the much-improved performance of the proposed methods in comparison with existing techniques, and also to show its robustness against the most common attacks.

3.2 Image Watermarking: Related Work

In this section, we will review four representative methods for digital image watermarking that are closely related to our proposed approaches. We briefly discuss their mathematical foundations and algorithmic methodologies as well as their limitations.

3.2.1 Spread spectrum watermarking

The general idea of spread spectrum watermarking system is to spread a narrow-band signal as a watermark over much wider important frequency bands which are obtained from the transformed cover image. The watermark in each band is small and undetectable. On the other hand the watermark detector, with knowledge of spreading function, should be able to extract and sum up the watermark. In order to provide a high level of robustness to JPEG compression the watermark is embedded in the first n lowest frequency components or the first highest magnitude components $V = \{v_i\}_1^n$ of the full image DCT [20]. The watermark consists of a sequence of real numbers $W = \{w_i\}_1^n$ where each w_i is chosen according to $N(0,1)$ where $N(0,1)$ denotes the normal distribution with mean 0 and variance 1. It is embedded into the image using

$$\grave{v}_i = v_i(1 + \alpha w_i) \tag{6}$$

where α is the watermark strength factor. Watermark detection is performed using the following similarity measure:

$$sim(W, \acute{W}) = \frac{W, \acute{W}}{\sqrt{\acute{W}, \acute{W}}} \tag{7}$$

The $\acute{W}$ is the extracted watermark, which is calculated as:

$$\{\acute{w}_i\}_1^n = \{(\frac{\acute{v}_i}{v_i} - 1) \diagup \alpha\}_1^n \tag{8}$$

Where $\acute{v}_i$ components are extracted from the received watermarked image, and v_i components are extracted from the original cover image. The watermark is present if

the extracted $sim(W, \acute{W})$ is greater than a threshold.

3.2.2 SVD watermarking scheme

Let $\boldsymbol{C}$ be a cover image of size $m \times m$ and $\boldsymbol{W}$ be a watermark image of size $n \times n$ with $n \leq m$. The SVD of the cover image and the watermark image are given by $\boldsymbol{C} = \boldsymbol{U\Sigma V'}$ and $\boldsymbol{W} = \boldsymbol{U}_w\boldsymbol{\Sigma}_w\boldsymbol{V}'_w$ respectively, where $\boldsymbol{\Sigma}_w$ =diag(λ_{wi}) is a diagonal matrix of singular values (SVs) of the visual watermark. The SVD watermark embedding algorithm is given by

$$\lambda_i^d = \lambda_i + \alpha\lambda_{wi}, \quad (1 \leq i \leq n) \tag{9}$$

where λ_i^d denotes the distorted SVs of the watermarked image, and α is a constant scaling factor. Hence the watermarked image is given by

$$\boldsymbol{M} = \boldsymbol{U\Sigma}^d\boldsymbol{V'}$$

where $\boldsymbol{\Sigma}^d$ =diag(λ_i^d). We may extract the SVs of the visual watermark using

$$\hat{\lambda}_{wi} = (\lambda_i^d - \lambda_i)/\alpha \tag{10}$$

Consequently, the extracted watermark $\widehat{\boldsymbol{W}}$ is given by $\widehat{\boldsymbol{W}} = \boldsymbol{U}_w\widehat{\boldsymbol{\Sigma}}_w\boldsymbol{V}'_w$, where $\widehat{\boldsymbol{\Sigma}}_w$ =diag($\widehat{\lambda}_{wi}$).

3.2.3 Block-based SVD scheme

In [28] the cover image $\boldsymbol{C}$ is divided into blocks of size $\ell \times \ell$ and the SVD of each block $\boldsymbol{L}$ is given by $\boldsymbol{L} = \boldsymbol{U}_\ell\boldsymbol{\Sigma}_\ell\boldsymbol{V}'_\ell$. The SVs of the visual watermark $\boldsymbol{W}$ are embedded into each block of the cover image by modifying the largest singular value of each block. The scaling factor α used for block embedding is chosen relative to the SVs of the block and it is multiplied with a constant percentage c. The block-based SVD watermark embedding algorithm is based in the following linear transformation

$$\lambda_i^d = \lambda_{max} + \alpha_i\lambda_{wi} \tag{11}$$

where i ranges from 1 to the number of block, λ_{max} denotes the largest SV of a block in the cover image, λ_{wi} denotes the SVs of the visual watermark, and λ_i^d denotes the distorted SV of a given block of the watermarked image. Hence, we may extract the SVs of the visual watermark using

$$\hat{\lambda}_{wi} = (\lambda_i^d - \lambda_{max})/\alpha_i \tag{12}$$

Therefore, the extracted watermark $\widehat{\boldsymbol{W}}$ is given by $\widehat{\boldsymbol{W}} = \boldsymbol{U}_w \hat{\boldsymbol{\Sigma}}_w \boldsymbol{V}'_w$ where $\hat{\boldsymbol{\Sigma}}_w = \text{diag}(\widehat{\lambda}_{wi})$.

3.2.4 DWT-SVD watermarking scheme

In [27] the cover image $\boldsymbol{C}$ is decomposed into four sub-bands: the approximation coefficient LL, and the detailed coefficients HL, LH, and HH. The SVD is applied to each sub-band $C^k \in$ {LL,HL,LH,HH} of the cover image

$$\boldsymbol{C}^k = \boldsymbol{U}_c^k \boldsymbol{\Sigma}_c^k \boldsymbol{V}_c'^k, k = 1, 2, 3, 4 \tag{13}$$

where λ_i^k, $1 \leq i \leq m/2$, are the SVs of $\boldsymbol{\Sigma}_c^k$. The SVD of the watermark image $\boldsymbol{W}$ of size $m/2 \times m/2$ is applied, $\boldsymbol{W} = \boldsymbol{U}_w \boldsymbol{\Sigma}_w \boldsymbol{V}'_w$ where $\boldsymbol{\Sigma}_w$ =diag(λ_{wi}). The SVs of the cover image in each sub-band are modified with the SVs of the watermark as follows:

$$\lambda_i^{dk} = \lambda_i^k + \alpha_k \lambda_{wi}, \quad 1 \leq i \leq m/2, \quad 1 \leq k \leq 4 \tag{14}$$

where λ_i^{dk} denotes the distorted SVs of a sub-band of the watermarked image. Hence, the four modified sub-bands are obtained by $\boldsymbol{M}^k = \boldsymbol{U}_c^k \boldsymbol{\Sigma}_d^k \boldsymbol{V}_c'^k$ where $\boldsymbol{\Sigma}_d^k$ =diag(λ_i^{dk}), $1 \leq k \leq 4$. Then, the inverse DWT is applied using the four sets of the modified DWT coefficients to produce the watermarked image. The algorithm is invertible, and the watermark can be extracted from the watermarked image.

3.3 Proposed Methodologies

3.3.1 Block-based image watermarking scheme using FHT and SVD

In this section, we provide the main steps of our first image watermarking scheme. The embedding and the extraction algorithms are illustrated in the block diagrams shown in Figures 5 and 6. Denote by A the cover image of size $K \times K$, and W the watermark image of size $M \times M$ with $M \leq 2K$.

Watermark embedding algorithm:

1) Divide the cover image into blocks of size $N \times N$.

2) To each block B, apply the FHT:

$$B^{\star} = (HBH)/N \tag{15}$$

where H is the Hadamard matrix of order N.

3) Apply SVD to the watermark:

$$W = U_W \Sigma_W V_W^T \tag{16}$$

where Σ_W is a diagonal matrix of singular values (SVs).

4) Randomize the SVs, and keep the seed in a secret key.

5) Modify the DC components of each transformed block $B^{\star}$ using

$$d_w^i = d_0^i + (b/\ell)\alpha\lambda_W^i \tag{17}$$

where d_0^i and d_w^i are the original and the modified DC components respectively, α is a constant, b is the block number, ℓ is the width of the watermark, and λ_W^i are the SVs of W.

6) Apply the inverse fast Hadamard transform (IFHT) to all the blocks to produce the watermarked image.

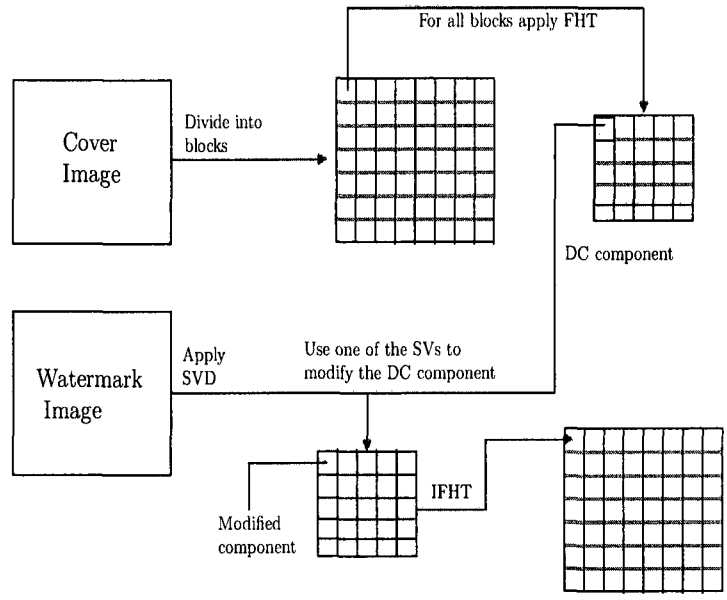


Figure 5: Block-based using FHT and SVD watermark embedding algorithm

Watermark extraction algorithm:

1) Divide the watermarked image into blocks of size $N \times N$

2) To each block C, apply the FHT: $C^{\star} = (HCH)/N$.

3) Extract the singular values from each block using

$$\lambda_w^i = (d_w^i - d_0^i)/(\alpha b/\ell) \tag{18}$$

where d_0^i and d_w^i are the original and the watermarked DC components respectively, α is a constant, b is the block number, and ℓ is the width of the watermark. Note that the d_0^i are saved in the secret key during the embedding stage.

4) Use the secret seed to arrange the SVs, then construct the watermark image using $W = U_W \Sigma_W^\star V_W^T$, where U_W and V_W are saved in the secret key during the embedding stage, and $\Sigma_W^\star$ is the extracted matrix of SVs.

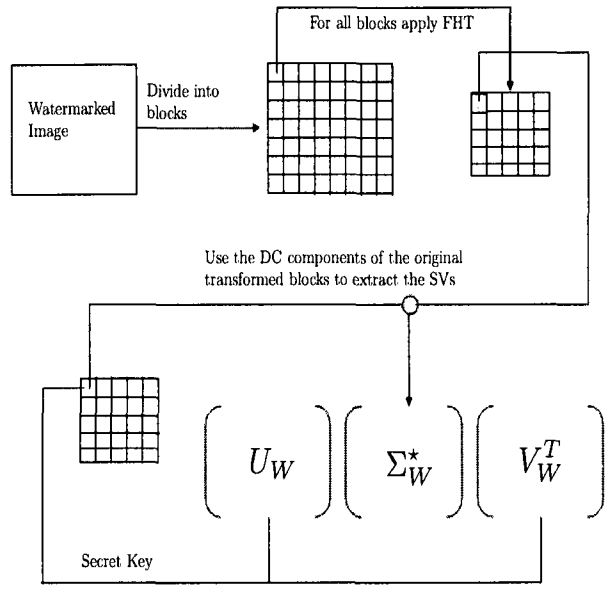


Figure 6: Block-based using FHT and SVD watermark extraction algorithm

3.3.2 Improved image watermarking scheme using FHT and DWT

In this section, we provide the main steps of the embedding and extraction algorithms of the second proposed image watermarking scheme. The algorithms are illustrated in the block diagrams shown in Figure 7 and 8.

In [27, 57], the authors prove experimentally that embedding the watermark in the low and high-frequency components increases the robustness against attacks. For example, embedding in low-frequency components increases the robustness to the attacks that have low-frequency characteristics like filtering, lossy compression, and

geometric distortions. However, embedding in the middle and high-frequency components is typically less robust against low-pass filtering and small geometric deformations of the image, but is extremely robust to noise addition, contrast adjustment, gamma correction, and histogram manipulations.

Therefore, our goal of the FHT/DWT approach may be described as applying multiple transforms to the cover image to embed the watermark many times in all the frequencies, which provides better robustness against attacks, amplifies the difficulty of destroying the watermark from all the frequencies, and provides a high visual quality of the watermarked image. The use of these multiple transforms is motivated by the facts that the DWT is a powerful analysis tool for multiresolution image representation in scalable lossless coding and the FHT has significant advantage in shorter processing time and ease of hardware implementation. Denote by $\boldsymbol{C}$ the cover image of size $m \times m$ and by $\boldsymbol{W}$ the watermark image of size $n \times n$ with $2n = m$.

Watermark embedding algorithm:

1) Apply DWT to the cover image $\boldsymbol{C}$ to obtain 4 sub-bands (LL, LH, HL, HH).

2) Divide each sub-band into blocks of size $\ell \times \ell$.

3) To each block $\boldsymbol{L}$, apply the FHT: $\widehat{\boldsymbol{L}} = (1/\ell)\boldsymbol{HBH}$, where $\boldsymbol{H}$ is the Hadamard matrix of order ℓ.

4) Apply SVD to the watermark image: $\boldsymbol{W} = \boldsymbol{U}_w \boldsymbol{\Sigma}_w \boldsymbol{V}'_w$, where $\boldsymbol{\Sigma}_w$ is a diagonal matrix of SVs.

5) Modify the DC components of the transformed blocks $\widehat{\boldsymbol{L}}$ using $d^i_w = d^i_0 + \beta\lambda^i_w$, where d^i_0 and d^i_w are the original and the modified DC components respectively, λ^i_w are the SVs of $\boldsymbol{W}$, and $\beta = \alpha b/n$ where α is a constant and b is the block number. Add the original DC components to the secret key if the original cover are not available for the extraction algorithm.

6) Apply the inverse fast Hadamard transform (IFHT) to all the blocks to produce the watermarked transformed sub-bands.

7) Apply the inverse discrete wavelet transform (IDWT) using the four watermarked sub-bands from Step 6 to produce the watermarked image.

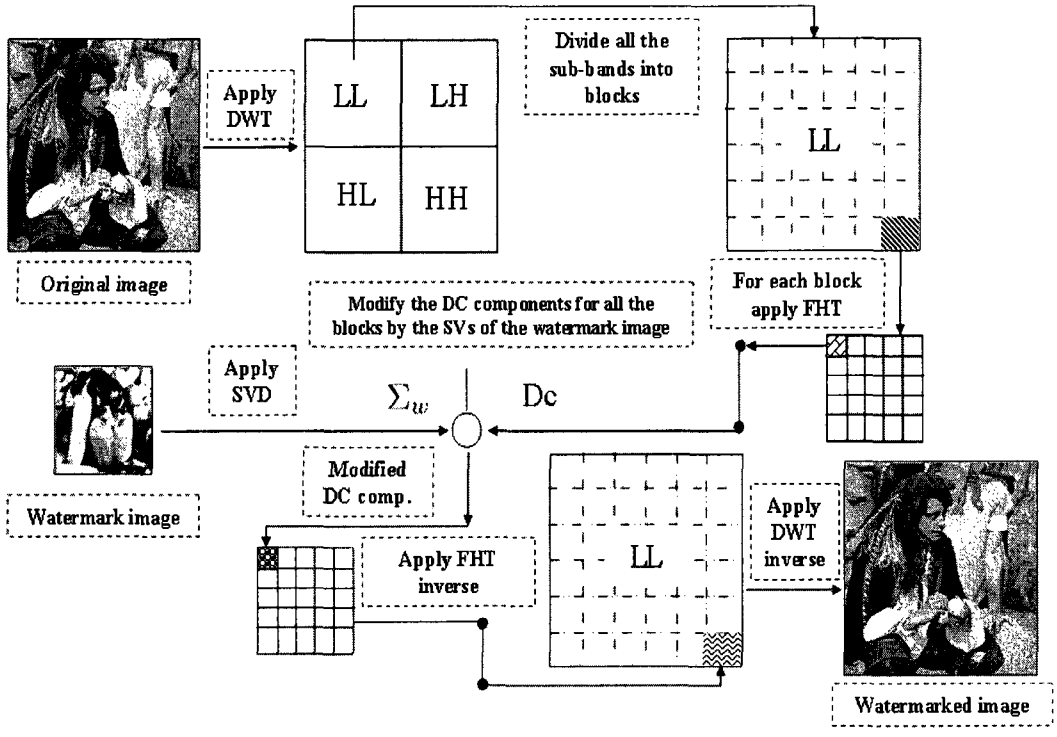


Figure 7: Improved image watermarking scheme using FHT and DWT. Watermark embedding algorithm

Watermark extraction algorithm:

1) Apply the first three steps of the embedding algorithm to the watermarked image.

2) For each sub-band, extract the SVs from each block using $\lambda_w^i = (d_w^i - d_0^i)/(\beta)$, where d_0^i and d_w^i are the original and the watermarked DC components respectively, and $\beta = \alpha b/n$ where α is a constant and b is the block number. Note that the d_0^i are saved in the secret key during the embedding stage.

3) Construct the four watermarks images using the SVs extracted from the four sub-bands: $\widehat{\boldsymbol{W}}_k = \boldsymbol{U}_w \widehat{\boldsymbol{\Sigma}}_k \boldsymbol{V}'_w$ where $\boldsymbol{U}_w$ and $\boldsymbol{V}_w$ are the left and right singular vectors of $\boldsymbol{W}$ respectively, and $\widehat{\boldsymbol{\Sigma}}_k$ is the extracted matrix of SVs for each sub-band k $(1 \leq k \leq 4)$.

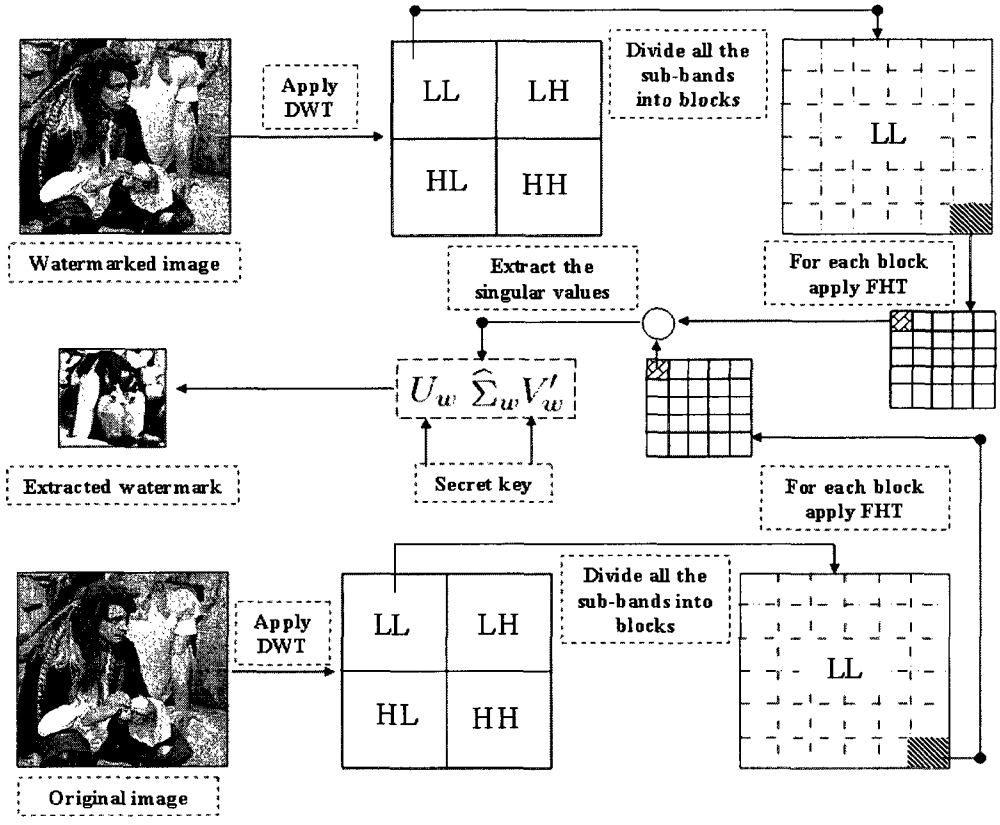


Figure 8: Improved image watermarking scheme using FHT and DWT. Watermark extraction algorithm

3.4 Image Watermarking Experimental Results

This section presents simulation results, we perform a number of experiments using a variety of gray-scale images to show the effectiveness of our proposed schemes. The experiments are basically divided into 2 types: tests on robustness and on the watermark invisibility of the proposed watermarking schemes.

3.4.1 Robustness of the block-based scheme using FHT and SVD

We conducted a number of experiments to test the robustness of the FHT/SVD watermarking algorithm in comparison with existing techniques, and in particular with the SVD watermarking technique [56] and the block-based method [28]. We performed our simulations on cover images and visual watermarks of size 512×512, and we set the scaling factor α to 0.7. First we divide the cover images into blocks of different sizes $2^i \times 2^i$, $i = 2, 3, 4, 5$. For $i = 4$ the resulting number of blocks is 1024, hence we may embed the watermark in two sub-bands of 512 blocks each. Then, we apply the watermark extraction algorithm to select the best extracted watermark that is the one having the highest correlation coefficient with the original watermark. Figures 9 (a) and (b) depict the cover image (peppers) and the watermark (letter-G) respectively. The watermarked peppers image and the extracted watermarks from the two sub-bands are shown in Figures 9 (c), (d), and (e) respectively.

We tested our FHT/SVD scheme against a wide range of attacks including JPEG compression, Gaussian noise, Gaussian blurring, Gamma correction, histogram equalization, cropping, rescaling, sharpening, mosaic, and rotation. Figures 10 and 11 show the watermarked peppers images with different kinds of attacks, and their corresponding best extracted watermarks respectively. For each attack, we select the extracted watermark that has the highest correlation coefficient with the original watermark. The correlation coefficient results are depicted in Figure 12, where the performance of FHT/SVD approach over SVD [56] and Block-based SVD [28] techniques is clearly demonstrated.

To gain further insight into the robustness of the proposed FHT/SVD method, we tested the performance of our algorithm using different block sizes as illustrated in Figure 13 which shows that we achieve more robustness against most of the attacks by using 8×8 block size. However, the best results for contrast, cropping and histogram

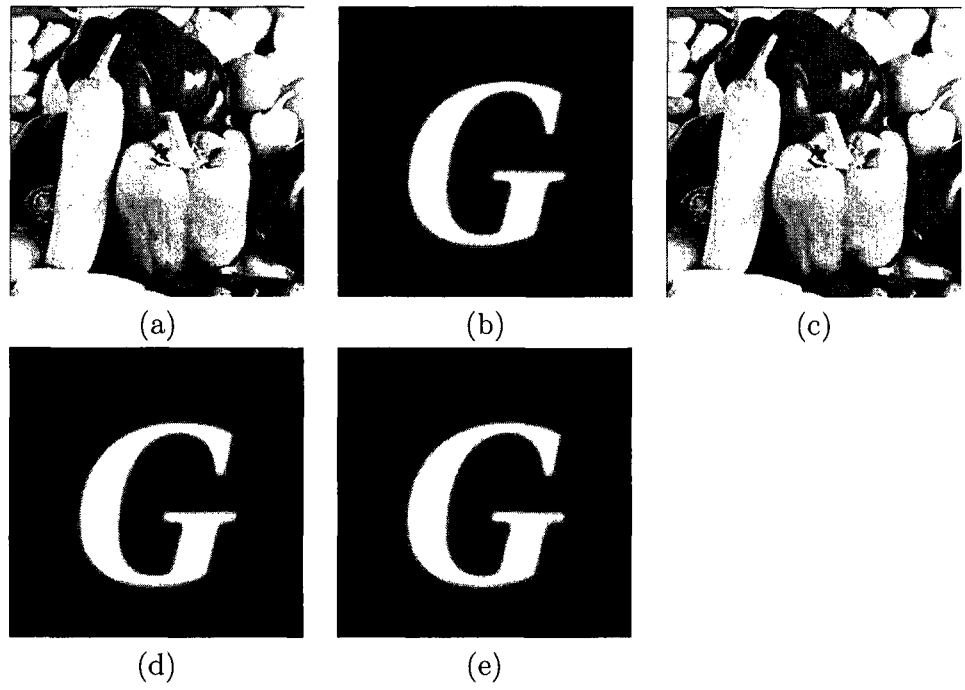


Figure 9: Block-based using FHT and SVD (a) original image, (b) visual watermark, (c) watermarked image, and (d)-(e) are extracted watermarks from the two sub-bands

equalization attacks achieved using block size of 4×4. Note that for a block of size 32×32, we can only embed watermark images of size 256×256. Figure 14 (resp. Figure 15) displays the correlation coefficient results with the same cover image Peppers (resp. with the same watermark Letter-G) but with different watermarks (resp. with different cover images). Note that the correlation coefficient is higher than 0.9 for most of the attacks. These results are consistent with various images and watermarks used for experimentation.

3.4.2 Robustness of the watermarking scheme using FHT and DWT

In this section, we conducted the experiments to test the robustness of the FHT/DWT scheme against attacks. Comparisons between our proposed FHT/DWT scheme and other transformed watermarking techniques were also conducted. The images used

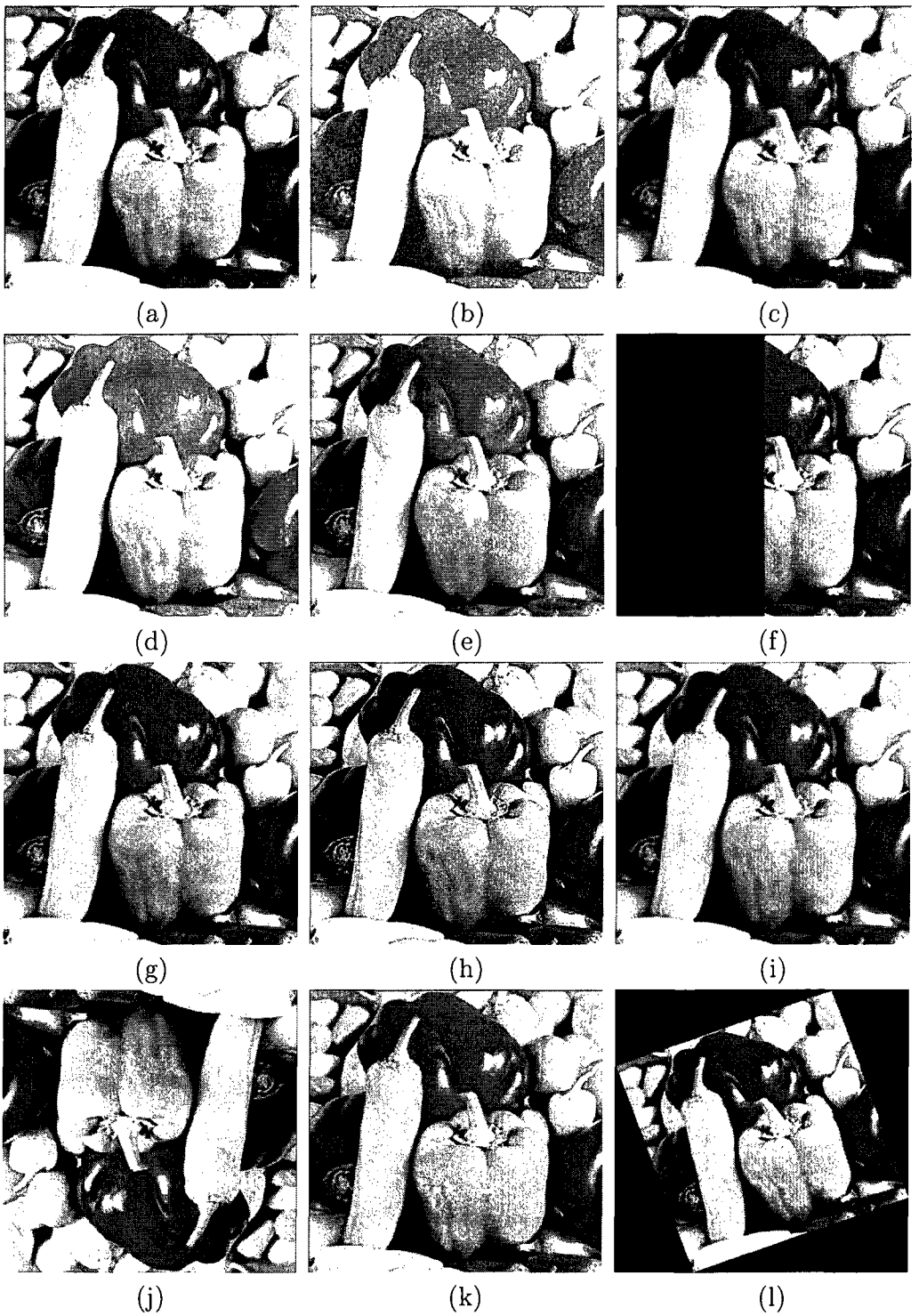


Figure 10: Block-based using FHT and SVD, watermarked image with different attacks: (a) JPEG, (b) Gaussian noise 0.3, (c) Gaussian blurred 5×5, (d) Gamma correction 0.6 , (e) histogram equalization, (f) cropping 50%, (g) resize 512-256-512, (h) sharpen 80, (i) contrast-20, (j) rotation 180^0, (k) mosaic 2, (l) rotation 20^0+ scaling

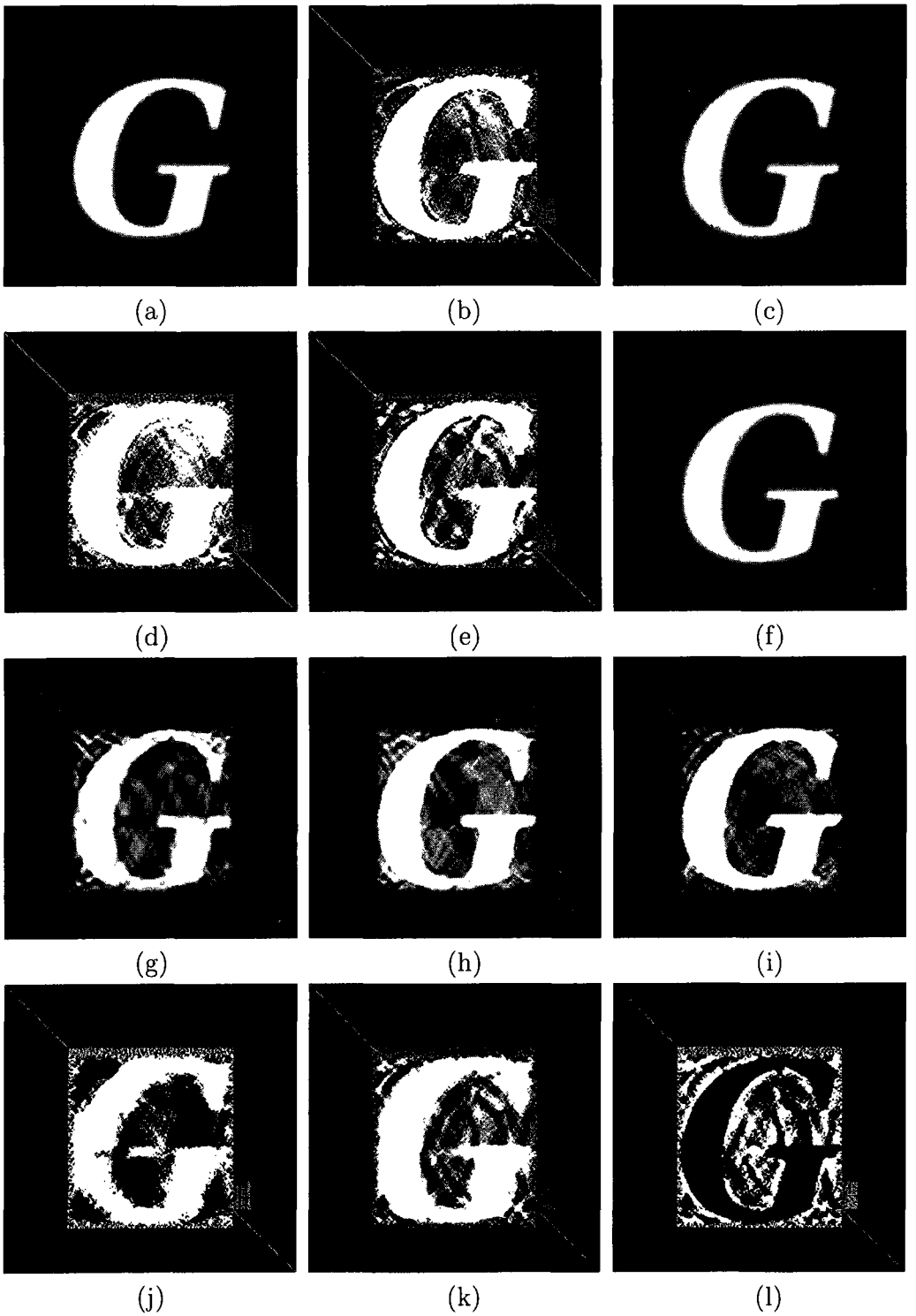


Figure 11: Block-based using FHT and SVD, best extracted watermarks under different attacks: (a) JPEG, (b) Gaussian noise 0.3, (c) Gaussian blurred 5×5, (d) Gamma correction 0.6 , (e) histogram equalization, (f) cropping 50%, (g) resize 512-256-512, (h) sharpen 80, (i) contrast-20, (j) rotation 180^0, (k) mosaic 2, (l) rotation 20^0+ scaling

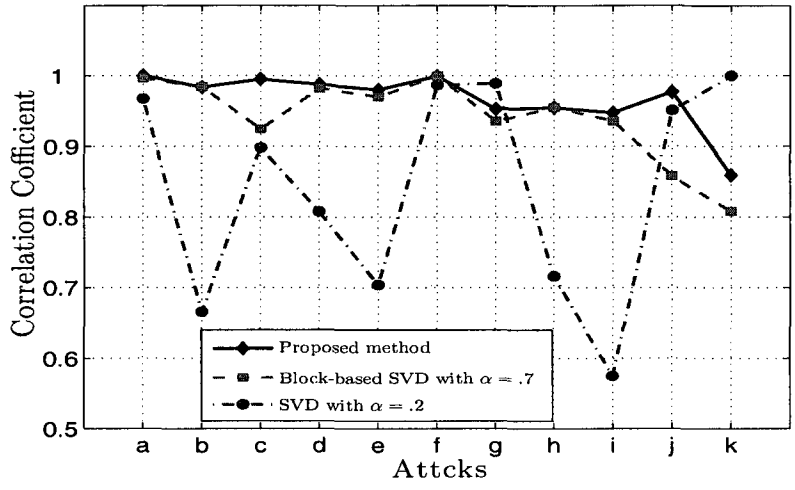


Figure 12: Block-based using FHT and SVD, correlation coefficient comparison results between the proposed FHT/SVD approach and other methods. (a) JPEG , (b) Gaussian noise 0.3, (c) Gaussian blurred 5×5, (d) Gamma correction 0.6 , (e) histogram equalization, (f) cropping 50%, (g) resize $512 - 256 - 512$, (h) sharpen 80, (i) contrast -20, (j) mosaic 2, (k) rotation 180^0

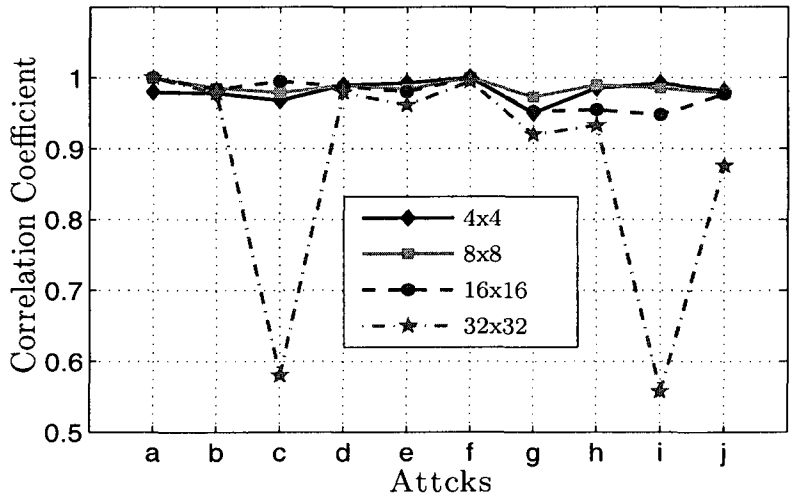


Figure 13: Block-based using FHT and SVD, correlation coefficient results using four different block sizes. (a) JPEG , (b) Gaussian noise 0.3, (c) Gaussian blurred 5×5, (d) Gamma correction 0.6 , (e) histogram equalization, (f) cropping 50%, (g) resize $512 - 256 - 512$, (h) sharpen 80, (i) contrast -20, (j) mosaic 2

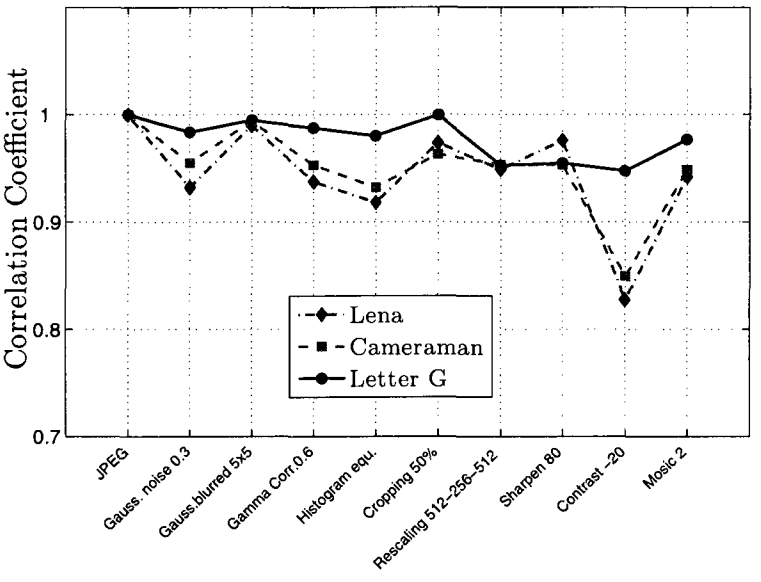


Figure 14: Block-based using FHT and SVD, correlation coefficient results using the same cover image Peppers and three different watermark images

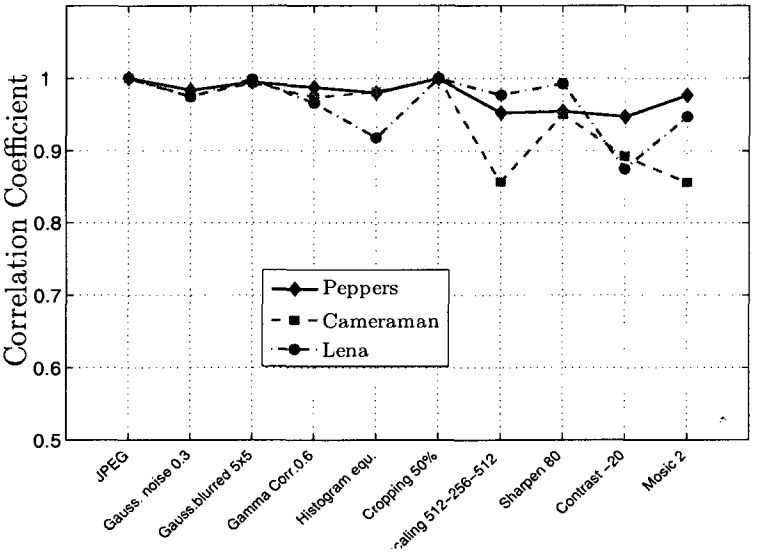


Figure 15: Block-based using FHT and SVD, correlation coefficient results using the same watermark Letter-G and three different cover images

in the experiments are of size 512×512 for the cover images and of size 256×256 for the watermark images. The LL sub-band has the lowest frequency components of the cover image and the highest wavelet coefficients (highest magnitude). The scaling factor is chosen according to the wavelet coefficients of each sub-band. The HL, LH, and HH sub-bands have very similar wavelet coefficients values; for simplicity, we used one scaling factor for all middle and high-frequency sub-bands.

Figure 16 shows the wavelet coefficients values for all four sub-bands. In particular, the wavelet coefficients of the LL sub-band are the highest among all the coefficients of the other bands.

We used a strength factor $\beta = \alpha b/n$ defined in terms of the block position b, the sub-band width n, and the constant scaling factor α set to 0.7 for the LL sub-band and 0.2 for all the other sub-bands. The scaling factors α are chosen experimentally to repel as much attack as possible and to also obtain a watermarked image with invisible degradation to the human observer.

We divide the sub-bands of size 256×256 into blocks of size 16×16. In this case the resulting number of blocks is 256. Hence, we may embed the watermark image once in each sub-band. Figures 17 (a) and 17 (c) depict an example of the cover image Guy and the watermark image Peppers respectively. The watermarked image Guy and one of the extracted watermarks from the four sub-bands are shown in Figures 17 (b), and 17 (d), respectively.

To verify the robustness, we applied different attacks to the watermarked image. The attacks include JPEG compression, Gaussian noise, multiplicative noise, Gaussian filter, deblurring with undersized point-spread function (PSF), deblurring with oversized PSF, gamma correction, histogram equalization, cropping, rescaling, sharpening, contrast adjustment, brightness change, motion blurring, and foreground. Figures 18-22 show the watermarked images with different kinds of attacks, and their

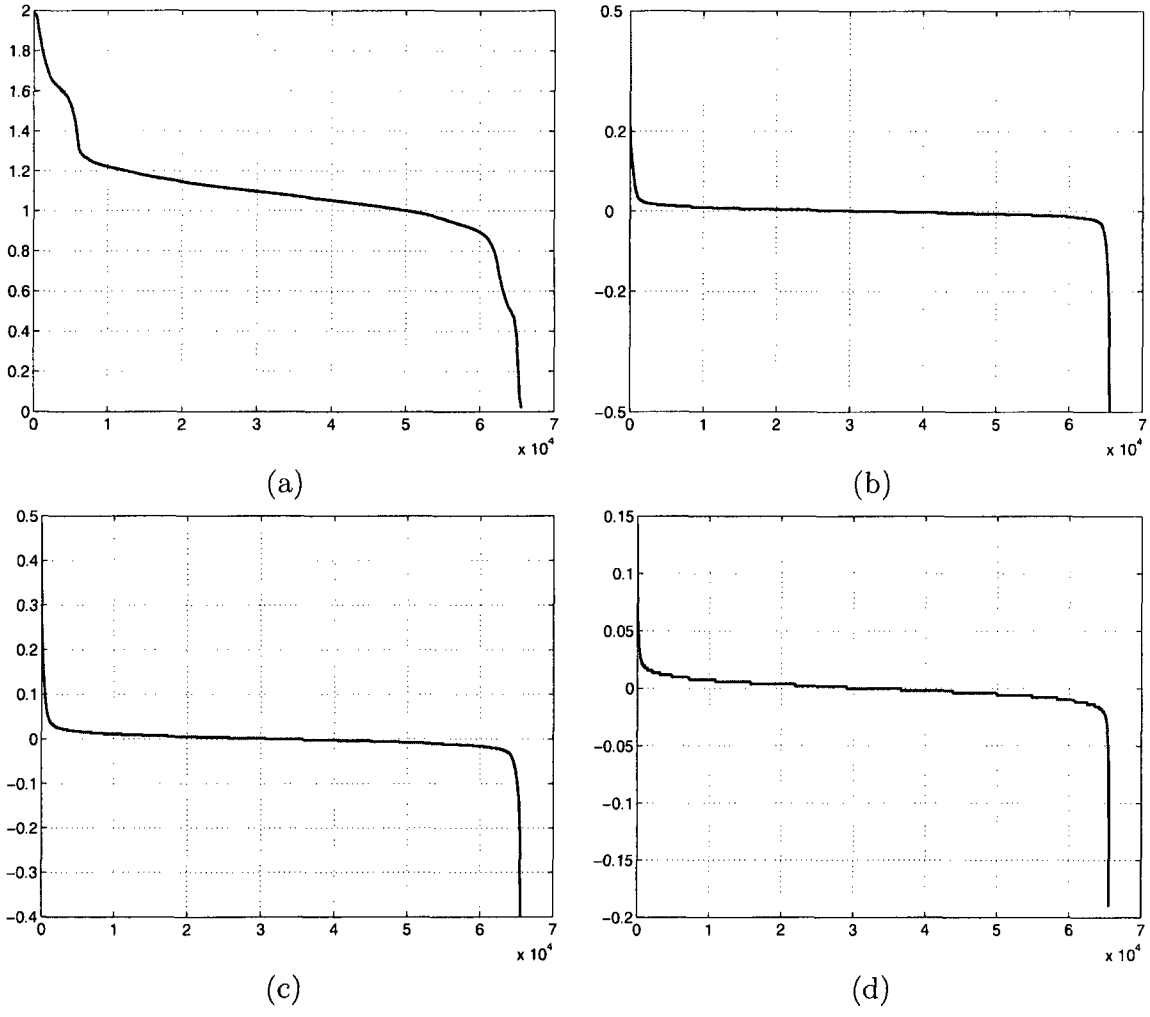


Figure 16: DWT coefficients of all the four sub-bands of the guy image (a) LL sub-band, (b) LH sub-band, (c) HL sub-band, and (d) HH sub-band

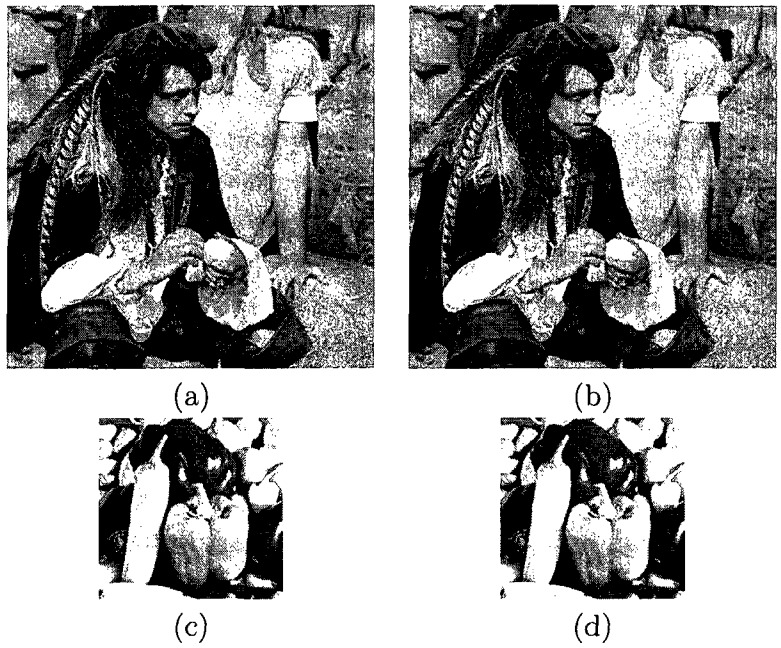


Figure 17: Image watermarking scheme using FHT and DWT. (a) original image, (b) watermarked image, (c) visual watermark, and (d) one of the four extracted watermarks from the four sub-bands

corresponding best extracted watermarks. For each attack, we extracted four watermarks from the four sub-bands, and then we selected the best watermark that had the highest correlation coefficient with the original watermark. The correlation coefficient ρ between the original watermark image W and the extracted watermark $\widehat{W}$ is defined as

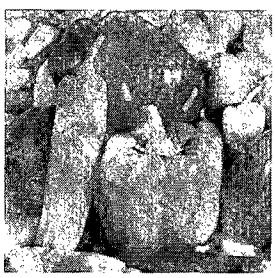
$$\rho = \frac{\sum_{i,j=1}^{n}(W_{ij} - \overline{W})(\widehat{W}_{ij} - \overline{\widehat{W}})}{\sqrt{\left(\sum_{i,j=1}^{n}(W_{ij} - \overline{W})^2\right)\left(\sum_{i,j=1}^{n}(\widehat{W}_{ij} - \overline{\widehat{W}})^2\right)}} \tag{19}$$

where $\overline{W}$ and $\overline{\widehat{W}}$ are the mean values of W and $\widehat{W}$ respectively. The label below each extracted image in Figures 18-22 show the best extracted watermark sub-band and the correlation coefficient between the original and the best extracted watermark.

Gauss.noise $\sigma = 0.3$

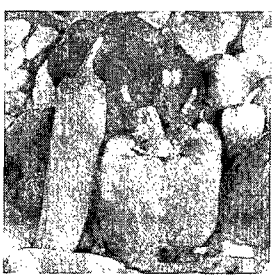
LL $\rho = 0.9921$

M.unifo.noise $\sigma = 0.05$



LH $\rho = 0.9390$

Salt+Pep.noise 4%



LH $\rho = 0.9287$

Figure 18: Illustration of the first group of the watermarked Guy images with different attacks and their best extracted watermarks

Low-pass filter [5x5]

LL $\rho = 0.9937$

Deblur. over PSF

LL $\rho = 0.8636$

Deblur. under PSF

LL $\rho = 0.9868$

Figure 19: Illustration of the second group of the watermarked Guy images with different attacks and their best extracted watermarks

Cropping 50%

HH $\rho = 0.9248$

Brightness -128

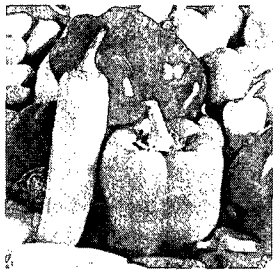
HL $\rho = 0.9987$

Histogram equalization

HH $\rho = 0.9885$

Figure 20: Illustration of the third group of the watermarked Guy images with different attacks and their best extracted watermarks

Motion 45°



HH $\rho = 0.9905$

Morphological opening

HH $\rho = 0.9289$

Sharpening

LL $\rho = 0.9943$

Figure 21: Illustration of the fourth group of the watermarked Guy images with different attacks and their best extracted watermarks

Rescaling $512 - 256 - 512$

LL $\rho = 0.8905$

JPEG $Q = 30\%$

LL $\rho = 0.9985$

Gamma correction 0.6

HL $\rho = 0.9895$

Figure 22: Illustration of the fifth group of the watermarked Guy images with different attacks and their best extracted watermarks

3.4.3 Invisibility

To measure the perceptual quality of the watermarked images, we calculate the PSNR which is used to estimate the quality of the watermarked images in comparison with the original ones. The PSNR experimental results as shown in Figure 23 and 24 and Table 2 clearly indicate that the FHT/DWT method gives high visual quality of the reconstructed image, and hence it guarantees the watermarks imperceptibility.

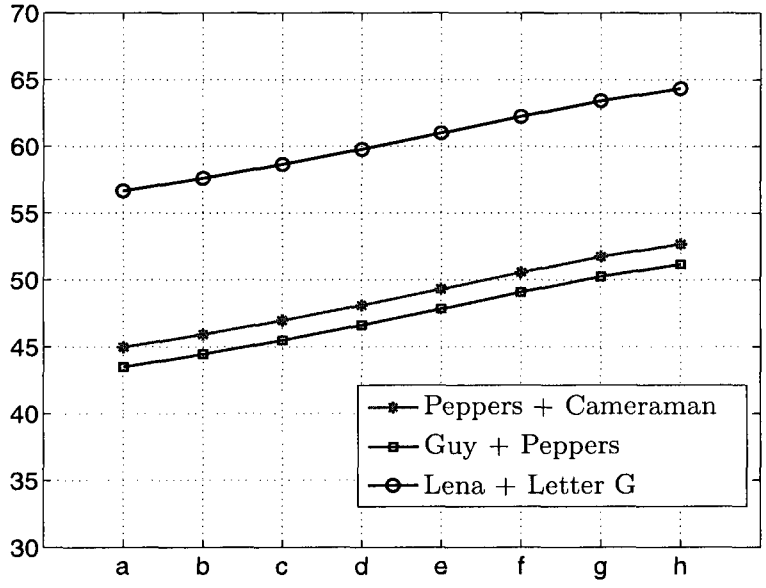


Figure 23: PSNR between different cover images and their corresponding watermarked images with different strength factors. For the LH, HL and HH sub-bands α is fixed = 0.2, and for the LL sub-band we used the following values (a) $\alpha = 0.8$, (b) $\alpha = 0.7$, (c) $\alpha = 0.6$, (d) $\alpha = 0.5$, (e) $\alpha = 0.4$, (f) $\alpha = 0.3$, (g) $\alpha = 0.2$, (h) $\alpha = 0.1$

3.4.4 Comparisons with existing techniques

We conducted several experiments to compare the robustness of the proposed FHT/DWT method with related existing techniques, in particular with the pure SVD watermarking scheme [56], DWT with SVD scheme [27], block-based FHT-SVD [2], and block-based SVD [28]. Figures 25-28 depict the correlation coefficient comparisons between

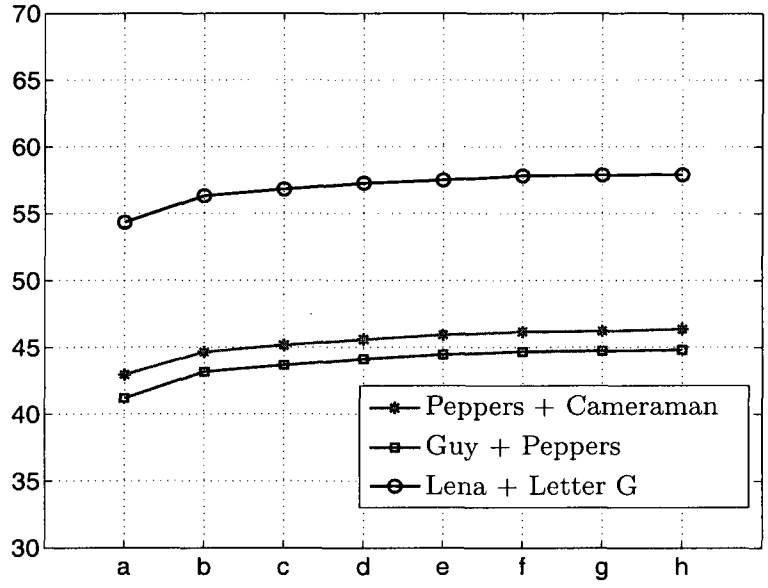


Figure 24: PSNR between different cover images and their corresponding watermarked images with different strength factors. For the LL sub-band α is fixed = 0.7, and for the LH, HL, and HH sub-bands we used the following values (a) $\alpha = 0.85$ (b) $\alpha = 0.5$, (c) $\alpha = 0.4$, (d) $\alpha = 0.3$, (e) $\alpha = 0.2$, (f) $\alpha = 0.1$, (g) $\alpha = 0.05$, (h) $\alpha = 0.01$

our proposed method and four different schemes with different attacks. In these comparisons, we used the Peppers, Guy, Lena and Liftingbody as a cover images and the Cameraman, Peppers, Letter-G, and MRI as watermark images. The results obtained for all the attacks clearly indicate that our proposed FHT/DWT method performs the best in terms of robustness against the attacks.

The negative correlation coefficients indicate that the extracted watermark image looks like the picture in the negative film (the lighter areas appear dark, and vice versa). For some watermark images like the letter-G, the extracted watermark could be considered as detected in both cases: a white letter-G on a uniform black background (high positive correlation) or a black letter-G on a white background (high negative correlation).

Table 2 lists the PSNRs of the proposed method and the other existing methods for the same test images. Our algorithm clearly outperforms all other methods in

terms of the visual quality of the reconstructed watermarked image. This better performance is, in fact, consistent with a variety of images used for experimentation.

Table 2: PSNR comparison results. The boldface number indicate the best PSNR for each example.

Watermarking Technique	**Peppers Cameraman**	**Guy Peppers**	**Lena Letter-G**	**Liftingbody MRI**
Proposed method	**46.93**	**44.45**	**57.61**	**48.06**
DWT+SVD[27]	25.55	25.29	29.63	30.25
Pure SVD[56]	25.58	25.33	29.66	30.28
Block SVD[28]	43.78	42.31	55.47	45.92
Block FHT[2]	44.44	42.42	56.58	47.2

3.4.5 Computational complexity

The computational complexity of embedding a watermark of size $m/2 \times m/2$ into a cover image of size $m \times m$ is obtained by the calculation of the DWT applied to the entire cover image and also by the calculation of the FHT to small blocks of size $(\ell \times \ell)$ where $\ell << m$. The DWT and FHT costs are given by $\mathcal{O}(m \log m)$ and $\mathcal{O}(\ell \log \ell)$ respectively. Therefore the most expensive process of our proposed scheme is the computation of the SVD of the watermark image which is given by $\mathcal{O}((m/2)^3)$.

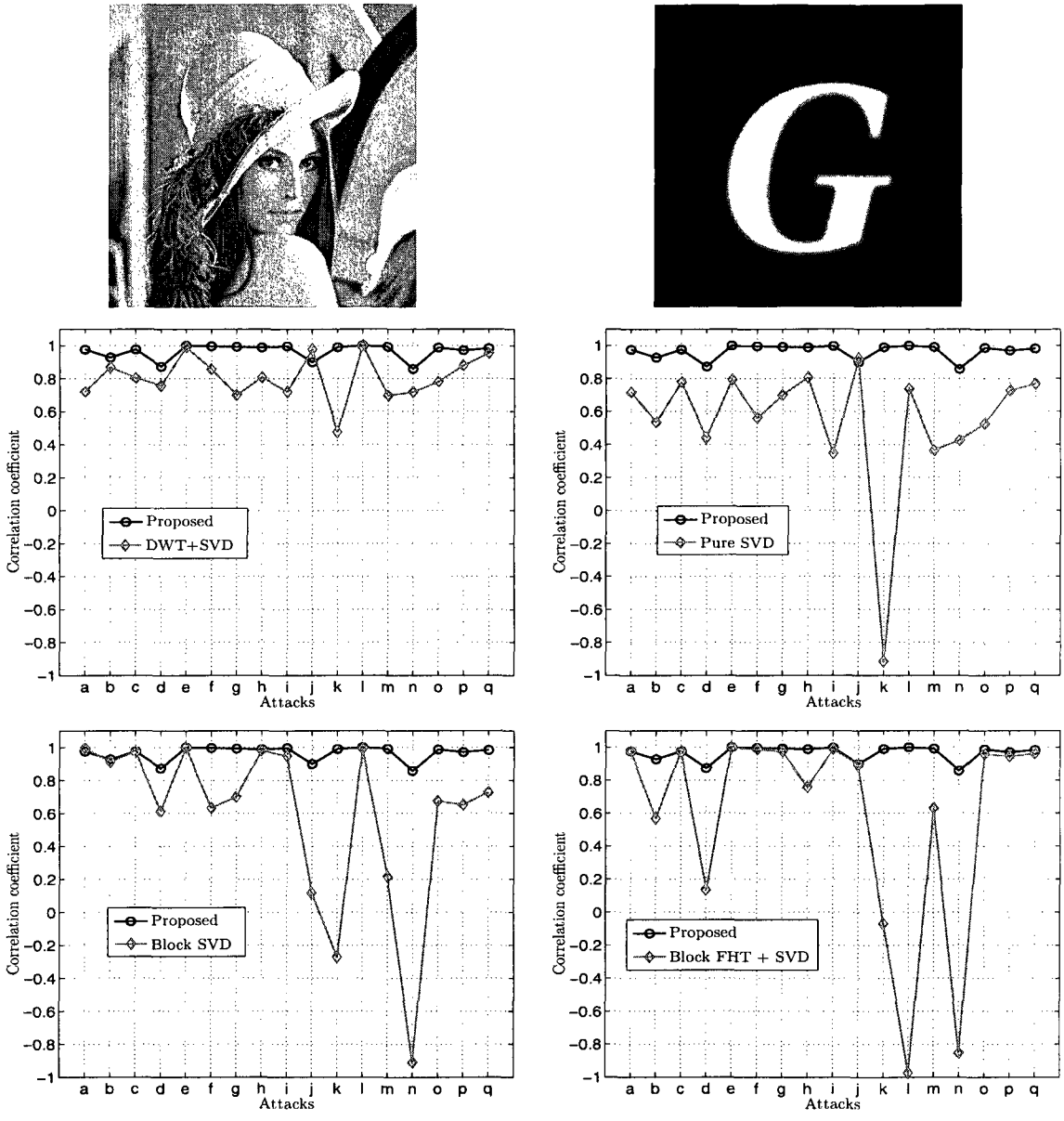


Figure 25: Correlation coefficient comparison results between the proposed approach and other methods. Lena and Letter-G are used as a cover image and a watermark image respectively. The scaling factor α used in this experiment is 0.7 for the LL band and 0.2 for all the other sub-bands. (a) Gaussian noise $\sigma = 0.3$, (b) multiplicative uniform noise $\sigma = 0.04$, (c) additive uniform noise $\sigma = 0.4$, (d) salt and peppers 4%, (e) JPEG compression $Q = 30\%$, (f) low-pass filter $[5x5]$, (g) Gamma correction 0.6, (h) histogram equalization, (i) sharpening, (j) rescaling $512 - 256 - 512$, (K) corroding 50% right, (l) brightness -128, (m) motion blurring 45°, (n) foreground image, (o) deblurring with undersized PSF (point-spread function), (p) deblurring with oversized PSF, and (q) deblurring with initial PSF.

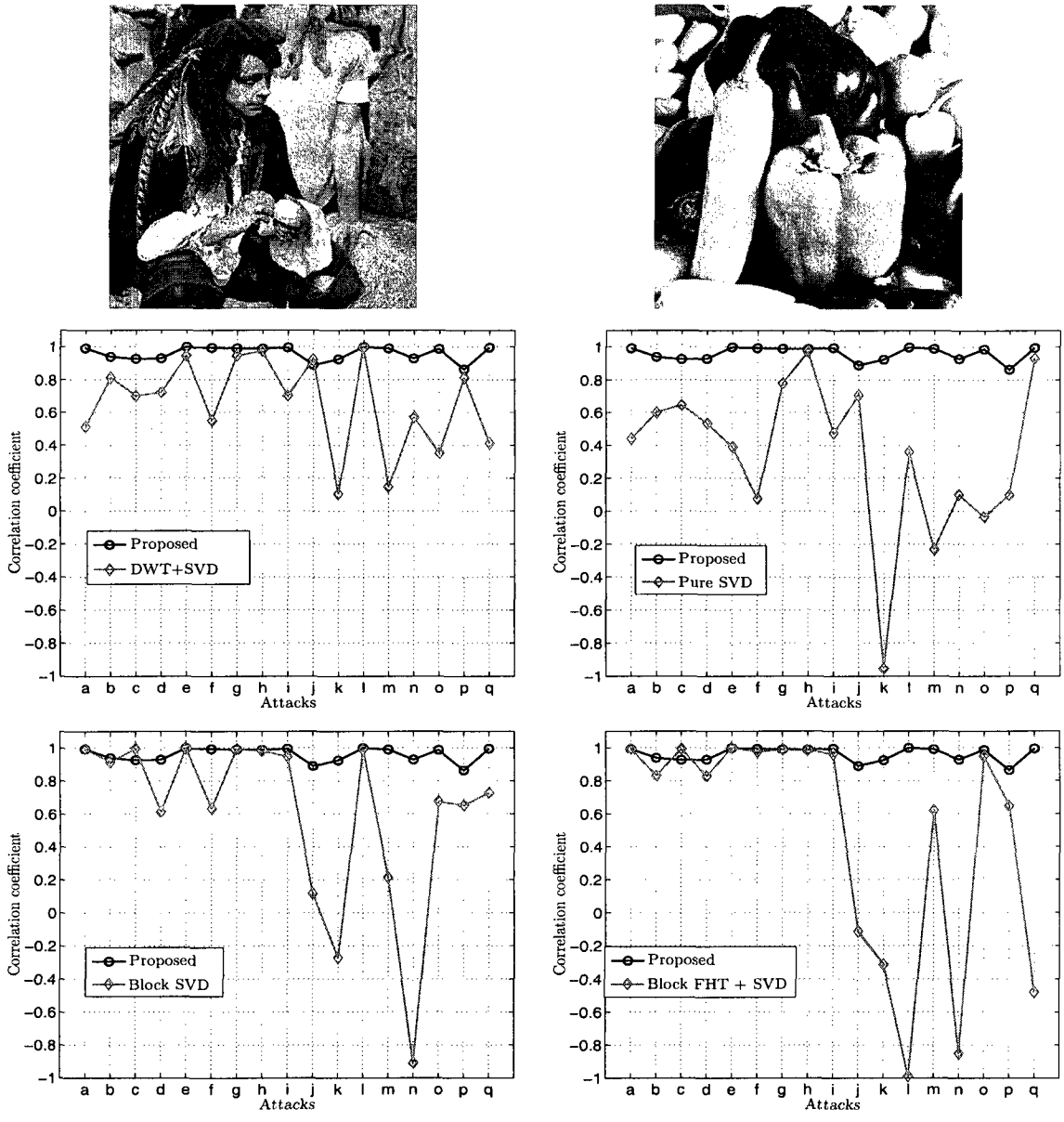


Figure 26: Correlation coefficient comparison results between the proposed approach and other methods. Guy and Peppers are used as a cover image and a watermark image respectively. The scaling factor α used in this experiment is 0.7 for the LL band and 0.2 for all the other sub-bands. (a) Gaussian noise $\sigma = 0.3$, (b) multiplicative uniform noise $\sigma = 0.04$, (c) additive uniform noise $\sigma = 0.4$, (d) salt and peppers 4%, (e) JPEG compression $Q = 30\%$, (f) low-pass filter [5x5], (g) Gamma correction 0.6, (h) histogram equalization, (i) sharpening, (j) rescaling $512 - 256 - 512$, (K) corroding 50% right, (l) brightness -128, (m) motion blurring 45^o, (n) foreground image, (o) deblurring with undersized PSF (point-spread function), (p) deblurring with oversized PSF, and (q) deblurring with initial PSF.

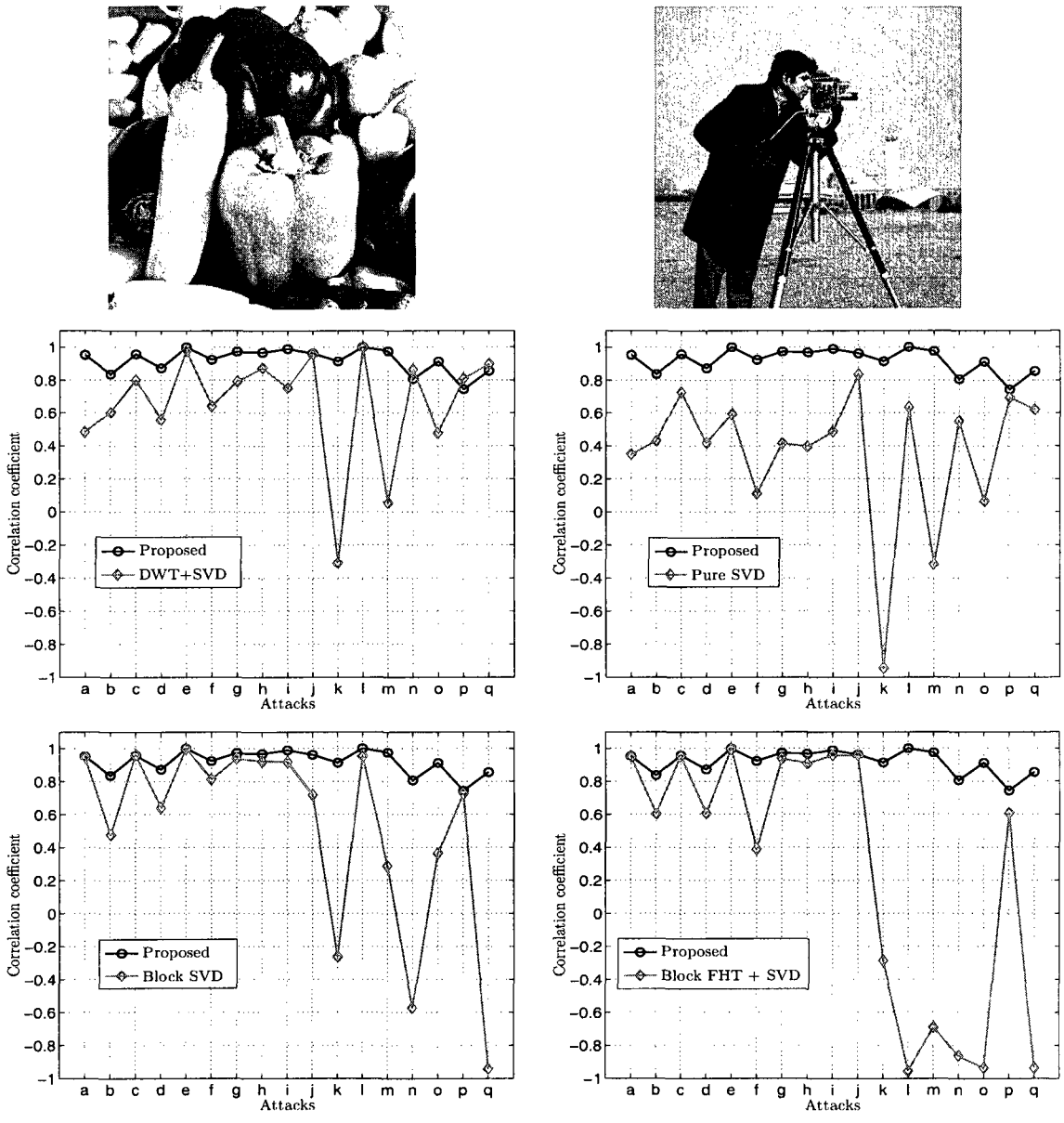


Figure 27: Correlation coefficient comparison results between the proposed approach and other methods. Peppers and Cameraman are used as a cover image and a watermark image respectively. The scaling factor α used in this experiment is 0.7 for the LL band and 0.2 for all the other sub-bands. (a) Gaussian noise $\sigma = 0.3$, (b) multiplicative uniform noise $\sigma = 0.04$, (c) additive uniform noise $\sigma = 0.4$, (d) salt and peppers 4%, (e) JPEG compression $Q = 30\%$, (f) low-pass filter $[5x5]$, (g) Gamma correction 0.6, (h) histogram equalization, (i) sharpening, (j) rescaling $512 - 256 - 512$, (K) corroding 50% right, (l) brightness -128, (m) motion blurring 45^o, (n) foreground image, (o) deblurring with undersized PSF (point-spread function), (p) deblurring with oversized PSF, and (q) deblurring with initial PSF.

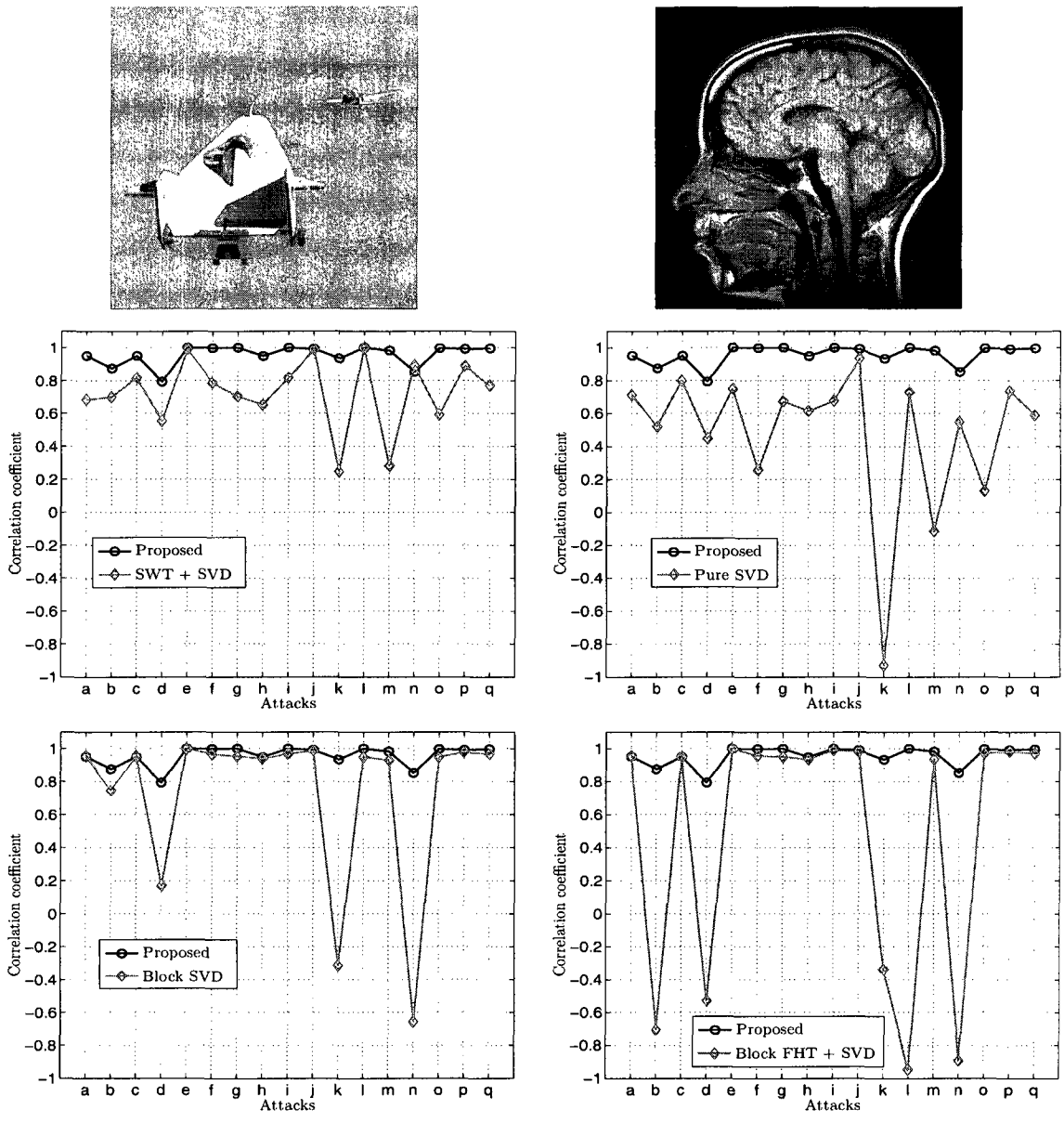


Figure 28: Correlation coefficient comparison results between the proposed approach and other methods. Liftingbody and MRI are used as a cover image and a watermark image respectively. The scaling factor α used in this experiment is 0.7 for the LL band and 0.2 for all the other sub-bands. (a) Gaussian noise $\sigma = 0.3$, (b) multiplicative uniform noise $\sigma = 0.04$, (c) additive uniform noise $\sigma = 0.4$, (d) salt and peppers 4%, (e) JPEG compression $Q = 30\%$, (f) low-pass filter $[5x5]$, (g) Gamma correction 0.6, (h) histogram equalization, (i) sharpening, (j) rescaling $512 - 256 - 512$, (K) corroding 50% right, (l) brightness -128, (m) motion blurring 45^o, (n) foreground image, (o) deblurring with undersized PSF (point-spread function), (p) deblurring with oversized PSF, and (q) deblurring with initial PSF.

Chapter 4

Spectral graph-theoretic approach to 3D mesh watermarking

In this chapter, we present a robust and imperceptible spectral watermarking method for high-rate embedding of a watermark into 3D polygonal meshes. Our approach consists of four main steps: (1) the mesh is partitioned into smaller sub-meshes, and then the watermark embedding and extraction algorithms are applied to each sub-mesh, (2) the mesh Laplacian spectral compression is applied to the sub-meshes, (3) the watermark data is distributed over the spectral coefficients of the compressed sub-meshes, (4) the modified spectral coefficients with some other basis functions are used to obtain uncompressed watermarked 3D mesh. The main attractive features of this approach are simplicity, flexibility in data embedding capacity, and fast implementation.

4.1 3D Watermarking Overview

Early algorithms on 3D watermarking [12, 36, 45, 67, 68] consist of embedding the watermark information directly by modifying either the 3D mesh geometry or the topology of the triangles. Recently, several watermarking algorithms in the frequency

domain have been proposed for 3D meshes [70, 74, 75] and are mainly based on multi-resolution mesh analysis. In [74] a set of scalar basis functions has been constructed over the mesh vertices where the watermark perturbs the vertices of each mesh along the direction of the surface normal, weighted by the basis functions. In [75] the original mesh is decomposed into a series of details at different scales by using the spherical wavelet transform, and the watermark is then embedded more in the approximation part than in the detail part. In [51] a watermarking scheme for subdivision surfaces has been presented. In [70] a watermarking algorithm based on the mesh spectral matrix has been proposed. The watermark is embedded by modifying the spectral coefficients and this idea was generalized in [19] to watermark point-based 3D geometries. A blind watermarking scheme robust against affine transformation attacks was proposed in [91]. Watermarking of texture attributes has been proposed in [29]. Two blind watermarking schemes that are robust against distortionless as well as distortion attacks are proposed in [18], where the idea was to modify the vertex norms distribution according to a watermark bit sequence. Wavelet blind watermarking scheme has been proposed in [80] where it is assumed that the host meshes are semi-regular. A wavelet decomposition is applied to embed the watermark at a suitable resolution level. A robust and fast spectral watermarking scheme for large meshes using new orthogonal basis functions based on radial basis functions has been proposed in [89]. In [41] the mesh Laplacian matrix was used to encode the 3D shape into a more compact representation by retaining the smallest eigenvalues and associated eigenvectors that contain the highest concentration of the shape information.

We propose a robust imperceptible watermarking approach [3, 4] using the spectral mesh compression. Our approach uses the mesh Laplacian matrix to embed a watermark in the spectral coefficients of a compressed 3D mesh. Extensive numerical experiments are performed to demonstrate the much improved performance of the proposed method in comparison with other methods. The visual error is evaluated

by computing a non-linear visual error metric between the original 3D models and the watermarked models obtained by our proposed algorithm.

The remainder of this chapter is organized as follows. In section 4.2, we briefly review some background material and describe the spectral compression of the mesh geometry. In Section 4.3 we introduce the proposed approach and describe in detail the watermark embedding and extraction algorithms. In Section 4.4, we present some experimental results and comparisons with existing techniques, and we show the robustness against the most common attacks.

4.2 Mesh Compression

4.2.1 3D model representation

In computer graphics and computer-aided design, 3D objects are usually represented as polygonal or triangle meshes. A triangle mesh $\mathbb{M}$ is a triple $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$, where $\mathcal{V} = \{\boldsymbol{v}_1, \ldots, \boldsymbol{v}_m\}$ is the set of vertices, $\mathcal{E} = \{e_{ij}\}$ is the set of edges with cardinality $|\mathcal{E}|$, and $\mathcal{T} = \{\boldsymbol{t}_1, \ldots, \boldsymbol{t}_n\}$ is the set of triangles. Each edge $e_{ij} = [\boldsymbol{v}_i, \boldsymbol{v}_j]$ connects a pair of vertices $\{\boldsymbol{v}_i, \boldsymbol{v}_j\}$. Two distinct vertices $\boldsymbol{v}_i, \boldsymbol{v}_j \in \mathcal{V}$ are adjacent (written $\boldsymbol{v}_i \sim \boldsymbol{v}_j$) if they are connected by an edge $e_{ij} \in \mathcal{E}$. The neighborhood of a vertex $\boldsymbol{v}_i$ is the set $\boldsymbol{v}_i^\star = \{\boldsymbol{v}_j \in \mathcal{V} : \boldsymbol{v}_i \sim \boldsymbol{v}_j\}$. The degree d_i of a vertex $\boldsymbol{v}_i$ is the cardinality of $\boldsymbol{v}_i^\star$. Let $\boldsymbol{v}_i = (x_i, y_i, z_i) \in \mathcal{V}$, $1 \leq i \leq m$, then the mesh vertex matrix $\boldsymbol{V}$ is the $m \times 3$ matrix whose i^{th} row is the vector $\boldsymbol{v}_i$.

$$\boldsymbol{V} = (\boldsymbol{v}_x\ \boldsymbol{v}_y\ \boldsymbol{v}_z) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{pmatrix}$$

Figure 29 depicts an example of a neighborhood $\boldsymbol{v}_i^\star$, where the degree of the vertex $\boldsymbol{v}_i$ is $d_i = 6$, and the number of triangles of the set $\mathcal{T}(\boldsymbol{v}_i^\star)$ is also equal to 6.

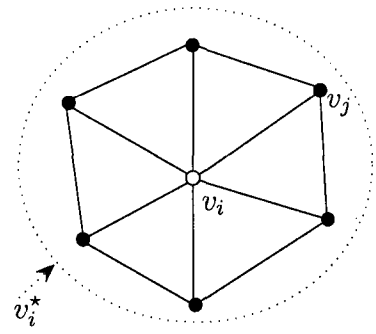


Figure 29: Vertex neighborhood $\boldsymbol{v}_i^\star$

4.2.2 Laplacian matrix of a triangle mesh

The mesh Laplacian matrix of a triangle mesh $\mathbb{M} = (\mathcal{V}, \mathcal{E}, \mathcal{T})$ is given by:

$$\boldsymbol{L} = \boldsymbol{D} - \boldsymbol{A} \tag{20}$$

where $\boldsymbol{A}$ is the adjacency matrix between the vertices, defined by:

$$\boldsymbol{A}_{ij} = \begin{cases} 1 & \text{if } \boldsymbol{v}_i \sim \boldsymbol{v}_j \\ 0 & \text{otherwise} \end{cases} \tag{21}$$

and $\boldsymbol{D}$ is the $m \times m$ diagonal matrix whose (i, i) entry is d_i. Figure 30 illustrates an example of a 3D triangle mesh and its sparse Laplacian matrix.

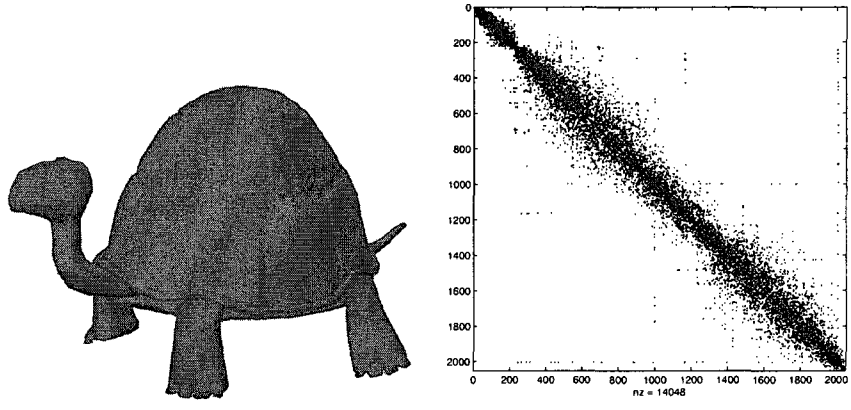


Figure 30: 3D triangle mesh and its Laplacian matrix

4.2.3 Spectral mesh compression

In [41] the 3D mesh geometry was represented as a linear combination of a few basis functions. The idea is to apply the eigen-decomposition to the mesh Laplacian matrix, and then discard the largest eigenvalues and their corresponding eigenvectors in order to reduce the dimensionality of the new spectral basis. A significant compression ratio with a very small loss in the mesh quality is obtained because this small number of basis functions contains the optimal concentration of the shape information. The eigen-decomposition of the Laplacian matrix $\boldsymbol{L}$ is given by

$$\boldsymbol{L} = \boldsymbol{B}\boldsymbol{\Lambda}\boldsymbol{B}^T \tag{22}$$

where $\boldsymbol{B} = (\boldsymbol{b}_1\ \boldsymbol{b}_2\ \ldots \boldsymbol{b}_m)$ is an orthogonal matrix whose columns $\boldsymbol{b}_i$ are the eigenvectors of $\boldsymbol{L}$ which we refer to as Laplacian basis functions, and $\boldsymbol{\Lambda} = diag\{\lambda_i : 1 \leq i \leq m\}$ is a diagonal matrix of the eigenvalues of $\boldsymbol{L}$ arranged in increasing order of magnitude. We express the mesh vertex matrix in the subspace spanned by the Laplacian matrix eigenvectors as follows:

$$\boldsymbol{V}^T = \boldsymbol{C}^T\boldsymbol{B}^T = \sum_{i=1}^{m} \boldsymbol{c}_i^T\boldsymbol{b}_i^T \tag{23}$$

where $\boldsymbol{C} = (\boldsymbol{c}_1\ \boldsymbol{c}_2\ \ldots \boldsymbol{c}_m)^T$ is an $m \times 3$ matrix of the spectral coefficient vectors, that is, $\boldsymbol{C} = \boldsymbol{B}^T\boldsymbol{V}$ where $\boldsymbol{C}$ is the projection of the mesh vertex matrix onto the Laplacian basis vectors. Moreover, Eq. (23) can be written as:

$$\boldsymbol{V}^T = \underbrace{\sum_{i=1}^{r} \boldsymbol{c}_i^T\boldsymbol{b}_i^T}_{\text{compressed}} + \sum_{i=r+1}^{m} \boldsymbol{c}_i^T\boldsymbol{b}_i^T = \boldsymbol{C}_r^T\boldsymbol{B}_r^T + \sum_{i=r+1}^{m} \boldsymbol{c}_i^T\boldsymbol{b}_i^T \tag{24}$$

where r is usually chosen to be smaller than m, and hence this yields a compressed mesh version $\mathbb{M}_r$ of the original mesh $\mathbb{M}$ with a very small loss in the mesh quality. The matrix $\boldsymbol{B}_r = (\boldsymbol{b}_1\ \boldsymbol{b}_2\ \ldots \boldsymbol{b}_r)$ contains the spectral basis vectors, and the matrix $\boldsymbol{C}_r = (\boldsymbol{c}_1\ \boldsymbol{c}_2\ \ldots \boldsymbol{c}_r)^T$ contains the spectral coefficient vectors. If we rewrite $\boldsymbol{V}$ and $\boldsymbol{C}$

in the form of 3-column matrices, that is

$$\boldsymbol{V} = (\boldsymbol{v}_x\ \boldsymbol{v}_y\ \boldsymbol{v}_z) = \begin{pmatrix} x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \\ \vdots & \vdots & \vdots \\ x_m & y_m & z_m \end{pmatrix}$$

and

$$\boldsymbol{C} = (\boldsymbol{c}_x \quad \boldsymbol{c}_y \quad \boldsymbol{c}_z) = \begin{pmatrix} c_{x_1} & c_{y_1} & c_{z_1} \\ c_{x_2} & c_{y_2} & c_{z_2} \\ \vdots & \vdots & \vdots \\ c_{x_m} & c_{y_m} & c_{z_m} \end{pmatrix},$$

The spectral coefficients in the x, y, and z-dimension are given by $\boldsymbol{c}_x = \boldsymbol{B}^T\boldsymbol{v}_x$, $\boldsymbol{c}_y = \boldsymbol{B}^T\boldsymbol{v}_y$, and $\boldsymbol{c}_z = \boldsymbol{B}^T\boldsymbol{v}_z$ respectively, see Figure 31 for an example. Figure 32 shows two examples of the mesh compression results using Laplacian-based method with 500 basis functions.

4.2.4 Mesh partitioning

The computation of the eigenvalues and the eigenvectors of a large $m \times m$ Laplacian matrix is prohibitively expensive = $\mathcal{O}(m^3)$. To circumvent this limitation, we partition a large 3D mesh into smaller sub-meshes. The embedding and extraction algorithms are then applied to each sub-mesh. In our approach we implemented a 3D mesh partitioning algorithm based on MeTiS software [42]. We used sub-meshes of 500 vertices on average as illustrated in Figure 33.

4.2.5 Watermarking in the mesh spectral domain

Watermarking schemes in the mesh spectral domain usually embed the watermark data into the mesh shape by modifying the spectral coefficients computed from the

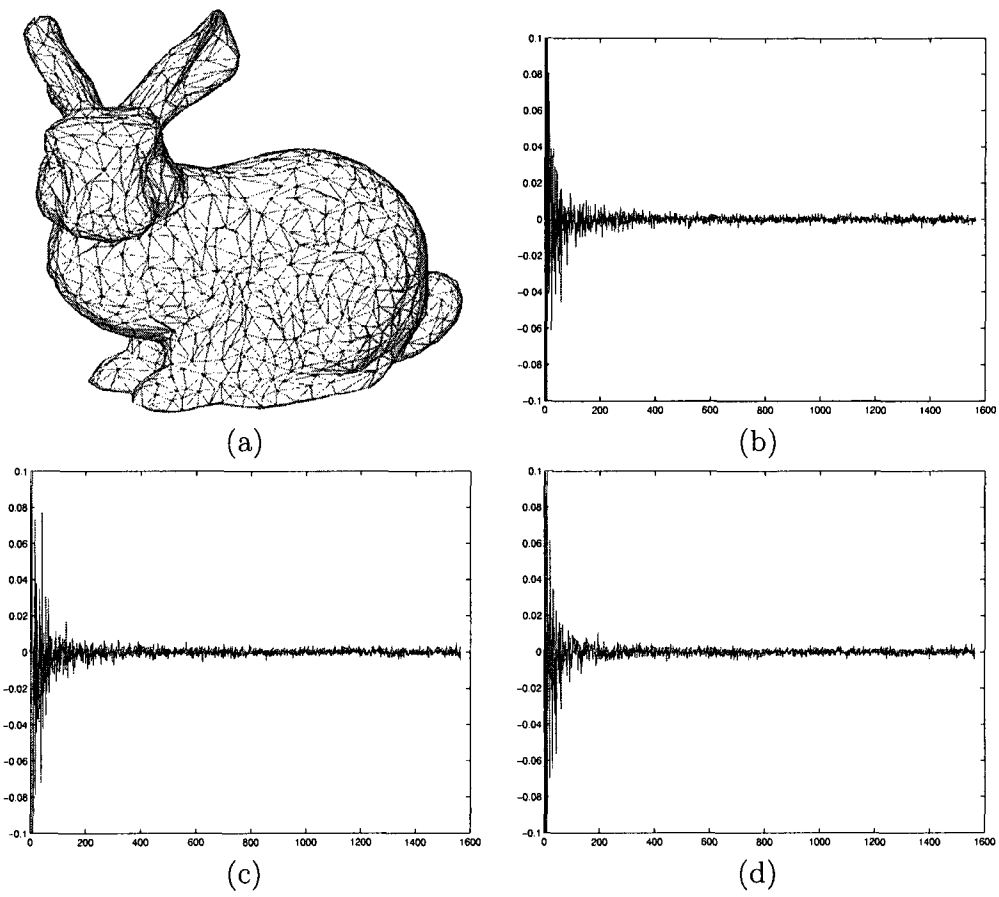


Figure 31: (a) 3D rabbit model, and its spectral coefficients in the (b) x-dimension, (c) y-dimension, and (d) z-dimension

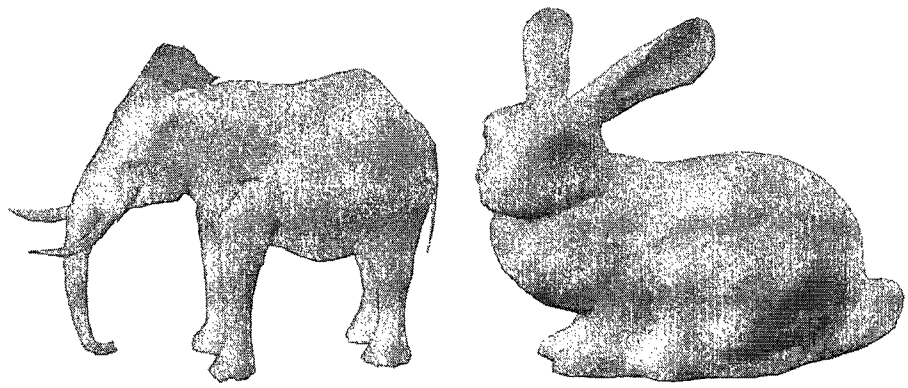


Figure 32: Spectral compression of the 3D models (a) Elephant model (b) Rabbit model

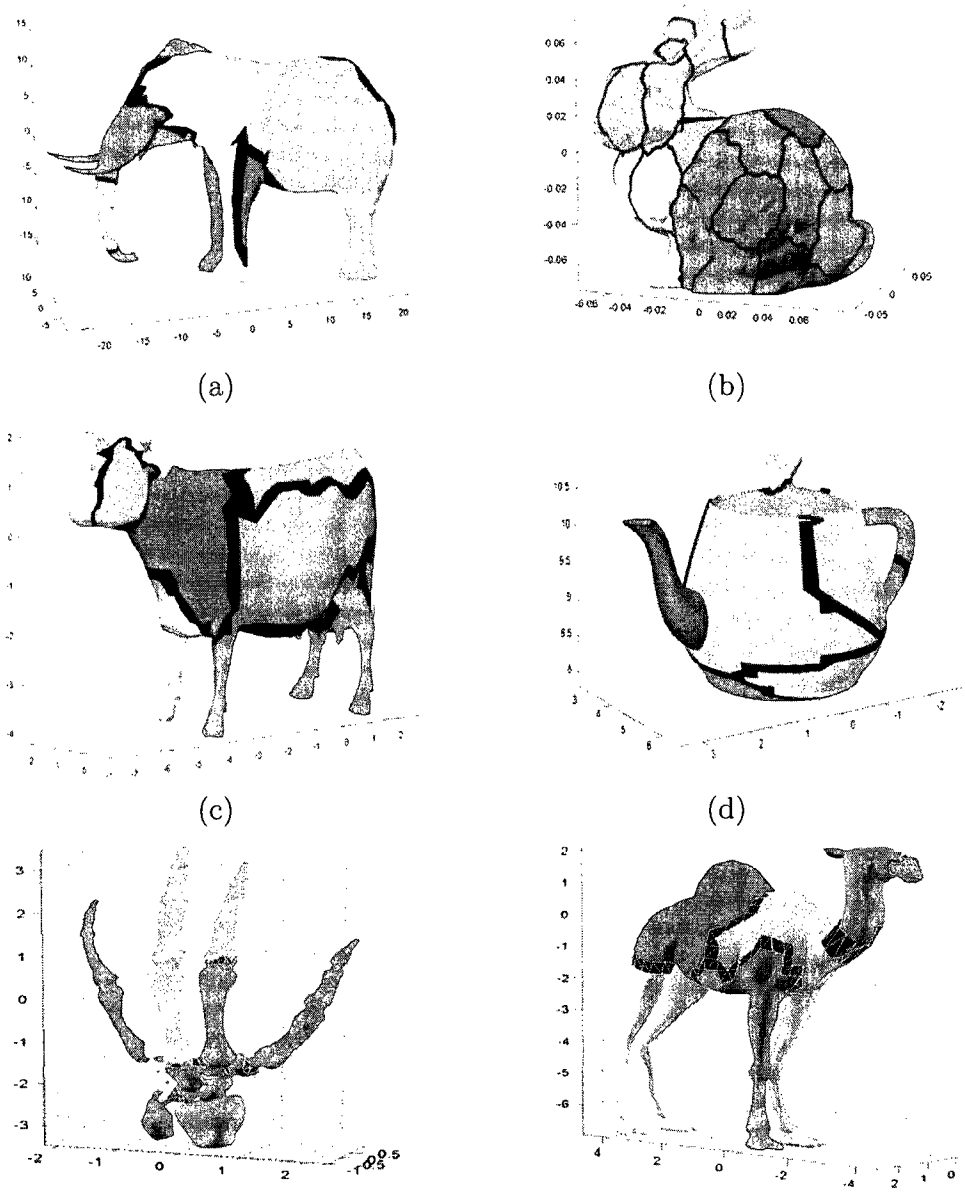


Figure 33: MeTis mesh partitioning. Each sub-mesh is colored by a random color. Black triangles represent edge cuts. (a) Elephant model: 4067 vertices, 8 sub-meshes, (b) Rabbit model: 20100 vertices, 40 sub-meshes, (c) Cow model: 2903 vertices, 7 sub-meshes, (d) Teapot model: 3241 vertices, 7 sub-meshes, (e) Hand model: 2600 vertices, 5 sub-meshes, and (f) Camel model: 2540, 5 sub-meshes

mesh topology. It is known that the smaller spectral coefficients correspond to the low-frequency components, and the high spectral coefficients correspond to the high-frequency components of the 3D mesh.

The two methods proposed in [69, 70] use a spectral approach by embedding the watermark directly in the spectral coefficients computed from the original 3D mesh.

Our experiments show that in order to increase the robustness against smoothing and compression attacks, the watermark should be embedded only in the spectral coefficients that represent the low-frequency components. However, in order to increase the robustness against additive random noise the same watermark should be embedded repeatedly as much as possible in the spectral coefficients vectors. So there is a trade-off between the robustness against random noise attack on the one hand and smoothing and compression attacks on the other hand.

In [70], the lowest five spectral vectors (low-frequency components) are not used in the embedding process due to their use in the realignment process before the extraction algorithm. In [69], the realignment algorithm is fixed, and the lowest frequency components are used to embed the watermark. However, the trade-off between smoothing, compression and noise attacks was not addressed.

Motivated by the need for more robustness against attacks we use the "Spectral compression of mesh geometry" introduced in [41]. We also use the fact that the low-frequency coefficients represent the global shape features of a 3D mesh (the rough approximation of the model may be reconstructed using small low-frequency spectral coefficients). The goal of our proposed scheme is to embed the watermark in the low-frequency components by repeating the watermark embedding process as much as possible. The number of the spectral coefficients of the compressed 3D mesh is exactly the same as the number of the spectral coefficients of the original 3D mesh without compression. As a result, we obtain the maximum number of the watermark repletion (maximum robustness against a noise attack). Moreover, all the

embedded watermarks are done in the low-frequency components that are used during the mesh compression stage. This guarantees the most robustness against smoothing and compression attacks.

4.3 Proposed Method: Watermarking in the Compressed Spectral Domain

In this section, we describe the main steps of the proposed watermark embedding and extraction algorithms, Figure 34 and Figure 36 show the flow diagrams. The goal of our proposed approach may be described as embedding the watermark in the global shape features which are represented by the low-frequency components of the 3D mesh. In this case we are not only increasing the robustness against attacks but also increasing the watermark imperceptibility. The proposed algorithm embeds the watermark information into the spectral coefficients of the compact representation of the 3D model.

4.3.1 Watermark embedding process

The 3D mesh is partitioned into smaller sub-meshes and the watermark embedding procedure is applied to each sub-mesh. Let $\mathbb{S}$ be a sub-mesh of n vertices and W be a pseudo-random vector of {-1,1} used as a watermark of size k such that $k << n$.

For all sub-meshes the watermark embedding process consists of the following steps:

1) Compute the Laplacian matrix $\boldsymbol{L}$ of size $n \times n$.

2) Compute the eigenvalues and the associated eigenvectors (basis functions) of $\boldsymbol{L}$.

3) Project the mesh vertices onto the basis functions to get the spectral coefficients matrix $\boldsymbol{C} = \boldsymbol{B}^T\boldsymbol{V}$ of the original sub-mesh $\mathbb{S}$.

4) Use the r basis functions ($r < n$) to obtain the compressed sub-mesh $\mathbb{S}_r$.

5) Repeat the steps (1-3) on the compressed 3D sub-mesh $\mathbb{S}_r$ to get the spectral coefficients matrix $\boldsymbol{C}_r$.

6) Duplicate the watermark d times, where $d = \lfloor n/k \rfloor$. Let the new watermark sequence be W_d. Modify the compressed spectral coefficients $\boldsymbol{C}_r$ by the watermark sequence W_d. So,

$$\widehat{\boldsymbol{C}}_r = \boldsymbol{C}_r + \alpha W_d \tag{25}$$

where $\widehat{\boldsymbol{C}}_r$ is the modified compressed spectral coefficients matrix, and α is a watermark strength.

7) Express the compressed watermarked sub-mesh vertices in the subspace using the modified spectral coefficients. Thus,

$$\boldsymbol{V}_{W_r}^T = \widehat{\boldsymbol{C}}_r^T \boldsymbol{B}^T = \sum_{i=1}^{r} \widehat{\boldsymbol{c}}_{ir}^T \boldsymbol{b}_i^T$$

where $\boldsymbol{V}_{W_r}^T$ is the compressed sub-mesh vertex matrix.

8) Use the remaining basis functions that are not used in step (4) to obtain the uncompressed watermarked sub-mesh with vertex matrix given by

$$\boldsymbol{V}_W^T = \boldsymbol{V}_{W_r}^T + \sum_{i=r+1}^{n} \bar{\boldsymbol{c}}_i^T \boldsymbol{b}_i^T \tag{26}$$

where $\bar{\boldsymbol{C}} = \{\bar{\boldsymbol{c}}_i\}_{r+1}^n$ is the spectral coefficients matrix of the high-frequency basis functions. Figure 35 shows two different 3D models with their corresponding watermarked meshes.

4.3.2 Watermark extraction process

We provide a private watermarking scheme, that is, the original unwatermarked object is necessary for the extraction process. An initial search step to find the right

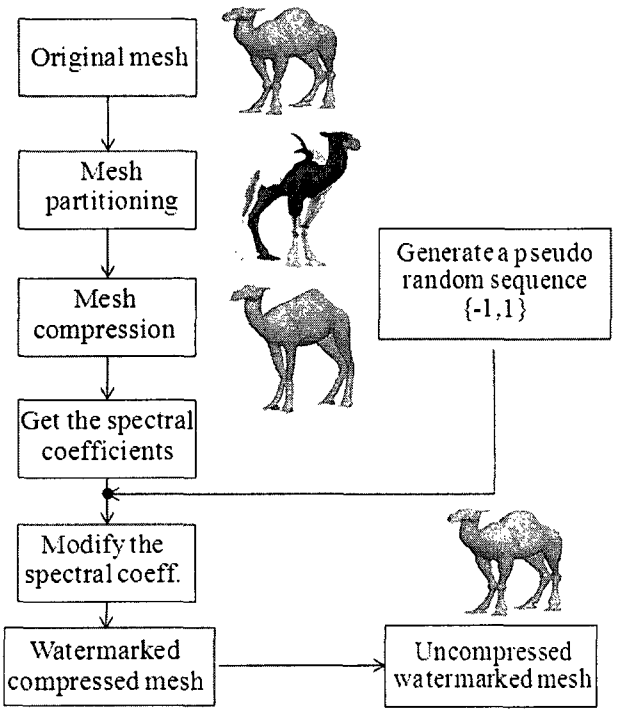


Figure 34: Watermark embedding process

original model from the owner database. We used the global geodesic measure proposed in [11]. The idea is to represent an object by a probabilistic shape descriptor that measures the global geodesic distance between two arbitrary points on the surface of an object. Unlike the Euclidean distance which is more suitable for linear spaces, the geodesic distance has the advantage of being able to capture the intrinsic geometric structure of the data. The matching task therefore becomes a one-dimensional comparison problem between probability distributions which is clearly much simpler than comparing 3D structures. The computational complexity of this method is $\mathcal{O}(m \log m)$, where m is the number of the centroids points of a 3D mesh model. Let the original unwatermarked mesh be $\mathbb{M}$ and the watermarked probably attacked mesh be $\widehat{\mathbb{M}}$.

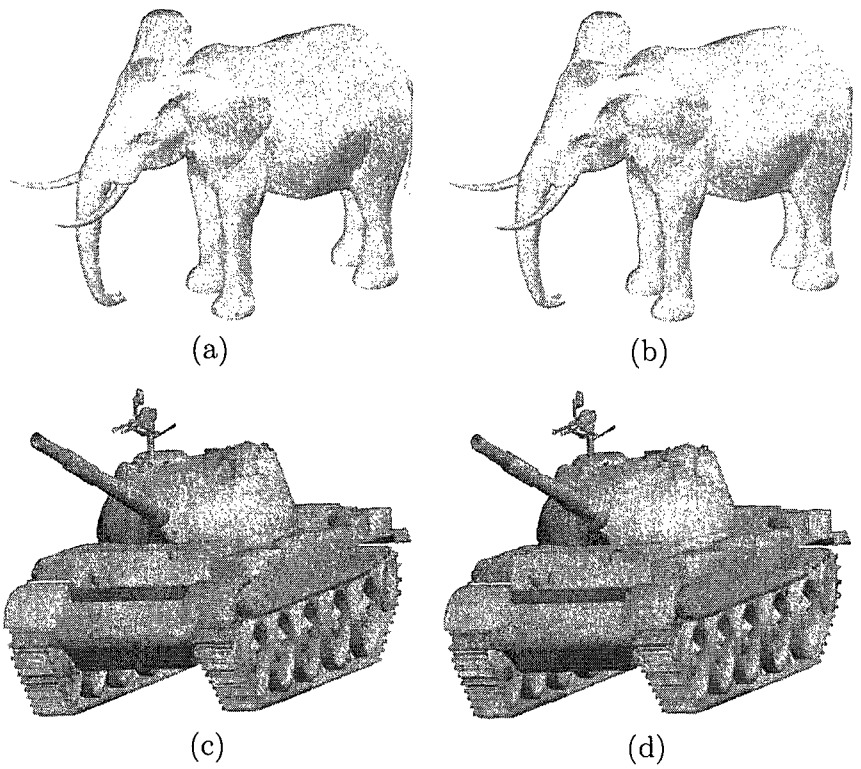


Figure 35: (a)-(c) Original 3D models and their corresponding watermarked models (b)-(d). Elephant model with (4076 vertices, 7999 faces) and Tank model with (15186 vertices, 13902 faces)

i) Mesh registration

We need to estimate the optimal rotation, scaling and translation to get $\widehat{\mathbb{M}}$ back to its initial scale and location if it is changed. This registration process is very important in order to extract the watermark successfully. We use the *iterative closest point* (ICP) method [13, 59] to select the optimal transformation (translation and rotation) to align two surfaces. Sometimes it is necessary to provide initial alignment, especially with the cropping attack. For the scaling transform, if both meshes represent the non-cropped objects or represent exactly the same surface patches of an object, we need to align both meshes to their initial position using ICP and then measure the ratio between the length of their corresponding axes.

ii) Remeshing

After registration, a remeshing is usually necessary to deal with the changes resulted by the attacks that may modify the mesh topology like simplification algorithms. To map the original topology we used the remeshing method [69] by tracing a ray through each vertex of the original mesh in the same direction of the normal vector of that vertex. If an intersection point is not found, create a vertex with the same coordinate as its reference in the original mesh. After applying the registration and remeshing processes to the watermarked and probably attacked mesh, we apply the watermark extraction algorithm which can be summarized as follows: The 3D mesh is partitioned into smaller sub-meshes using the same procedure as in the embedding process.

For each sub-mesh:

1) Apply the first four steps of the embedding process with the same number of basis functions to the initial and the watermarked sub-meshes to obtain compressed version.

2) Apply the steps $(1-3)$ of the watermarking algorithm on the compressed 3D sub-meshes. Then, to extract the watermark vector we compare the spectral coefficients of the initial compressed sub-mesh with the spectral coefficients of the watermarked and probably attacked compressed sub-mesh.

$$w_x^i = (\widehat{x}_i - x_i)/\alpha$$

$$w_y^i = (\widehat{y}_i - y_i)/\alpha$$

$$w_z^i = (\widehat{z}_i - z_i)/\alpha \tag{27}$$

where $(\widehat{X}, \widehat{Y}, \widehat{Z})$, (X, Y, Z) are the spectral vectors of the compressed watermarked and the compressed initial sub-meshes respectively, and α is a constant saved in the secret key during the embedding process.

3) Construct

$$\overline{W} = (W_x + W_y + W_z)/3 \tag{28}$$

where W_x, W_y, and W_z are the extracted watermark vectors in step (2).

4) Find the average watermark vector $\overline{W}_d$ from $\overline{W}$ which contains $d = \lfloor n/k \rfloor$ watermark copies. Finally the extracted vector is given by the decision rule:

$$\widehat{W}_d = \{\overline{w}_{d_i}\}_{i=1}^{k} = \begin{cases} -1 & \text{if } w_{d_i} < 0 \\ 1 & \text{otherwise} \end{cases} \tag{29}$$

5) If the correlation coefficient between $\widehat{W}_d$ and W is greater than a predefined threshold, then the watermark is present.

4.4 Experimental Results

Our experiments were performed using a variety of 3D models represented as triangle meshes. Table 3 shows the characteristics of the 3D models used in our experiments

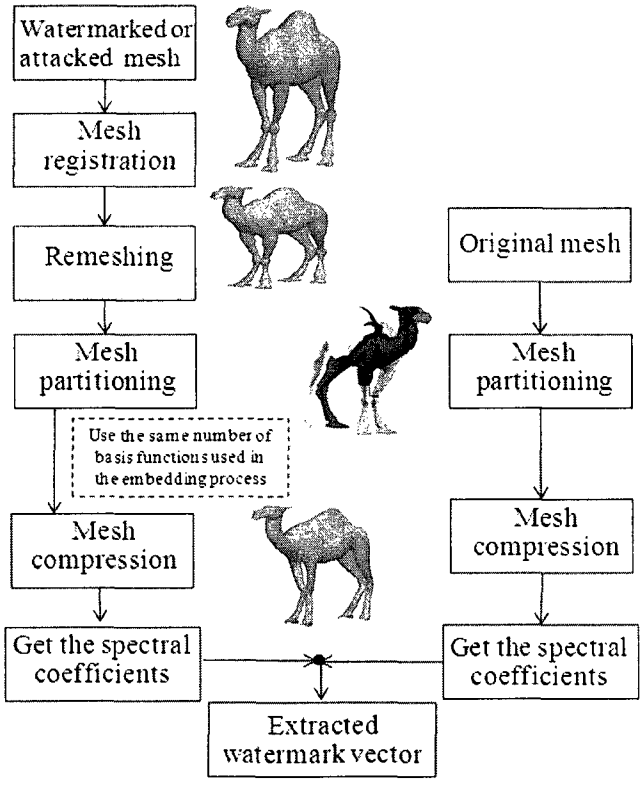


Figure 36: Watermark extraction process

Table 3: characteristics of the 3D models used in our experiments.

Model	# vertices	# faces	# patches	# watermarks
Camel	4001	8050	7	750
Rabbit	20100	39999	40	3768
Max Planck	5040	10067	10	945
Elephant	4067	7999	8	762
Tank	15186	13902	31	2847
Mesh part	2496	5000	4	468
Cow	2903	5804	4	543
Rocker arm	10000	20000	18	1875
Hand	10113	19801	18	1896

(collected by courtesies of the Stanford University, Avalon, Cyberware, and the Max Planck Institute). We conducted experiments to test the imperceptibility of the watermark and the robustness against attacks. A comparison between our proposed scheme and other spectral watermarking techniques were also conducted.

4.4.1 Imperceptibility

In order to achieve high visual quality of the watermarked model, the watermark strength factor α should be taken into consideration. The most common embedding rule is the additive one: $\overline{x}_i = x_i + \alpha w_i$ where x_i is the i^{th} component of the original vector, w_i the i^{th} sample of the watermark, and α is the watermark strength. The parameter α is chosen by the owner of the 3D model such that it is small enough to keep the watermark imperceptible to the human observer, and large enough to resist as many attacks as possible. Figures 37-38 depict the robustness of the camel model with noise and smooth attacks respectively. Different strength factors, different noise rates, and different smoothing iterations have been used. Clearly, a higher strength factor gives better correlation coefficients between the original watermark vector and the average vector of the extracted watermarks. Figure 59 shows an example of the

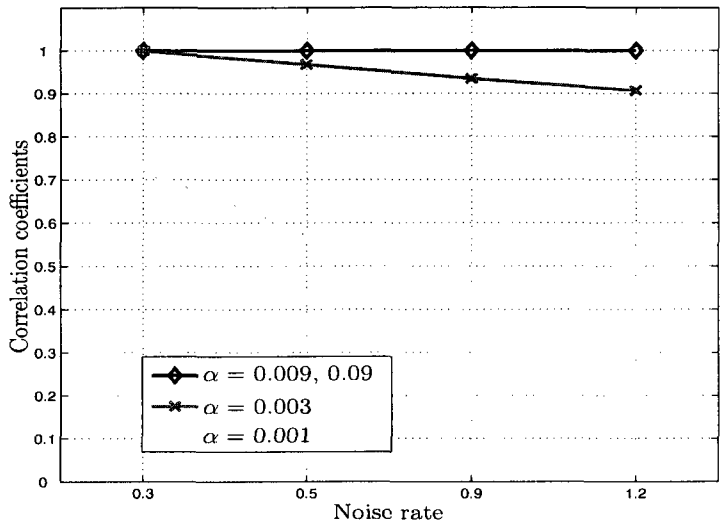


Figure 37: Correlation coefficient results for the camel model using four different strength factors and four noise rates attacks

influence of the strength factor on the watermark perceptibility. The watermark-embedded in the camel model with strength factor 0.09 is not only perceptible to the human observer but it may also destroy the overall geometric structure of the 3D model. So there is a trade-off between the robustness of the watermarked model against attacks, and the degradation of the original 3D mesh.

To quantify the imperceptibility of the proposed approach, we used the non-linear visual error [93] $D(\mathbb{M}, \widehat{\mathbb{M}})$ defined between the original model $\mathbb{M}$ and the watermarked model $\widehat{\mathbb{M}}$ as follows

$$D(\mathbb{M}, \widehat{\mathbb{M}}) = (\sum_{i=1}^{m} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\|^2 + \|\mathcal{A}(\boldsymbol{v}_i) - \mathcal{A}(\hat{\boldsymbol{v}}_i)\|^2)/(2m)$$

where $\{\boldsymbol{v}_i\}_{i=1}^m$ and $\{\hat{\boldsymbol{v}}_i\}_{i=1}^m$ are the mesh vertex sets of $\mathbb{M}$ and $\widehat{\mathbb{M}}$ respectively. $\mathcal{A}$ is a non-linear diffusion operator [92] defined as

$$\mathcal{A}(\boldsymbol{v}_i) = (1/d_i) \sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^*} (\boldsymbol{v}_i - \boldsymbol{v}_j)\Big(g(|\nabla \boldsymbol{v}_i|) + g(|\nabla \boldsymbol{v}_j|)\Big) \tag{30}$$

where the gradient magnitudes are given by

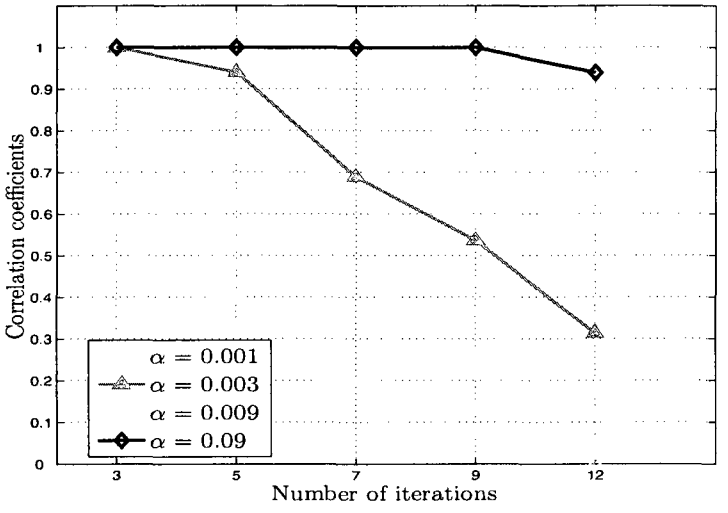
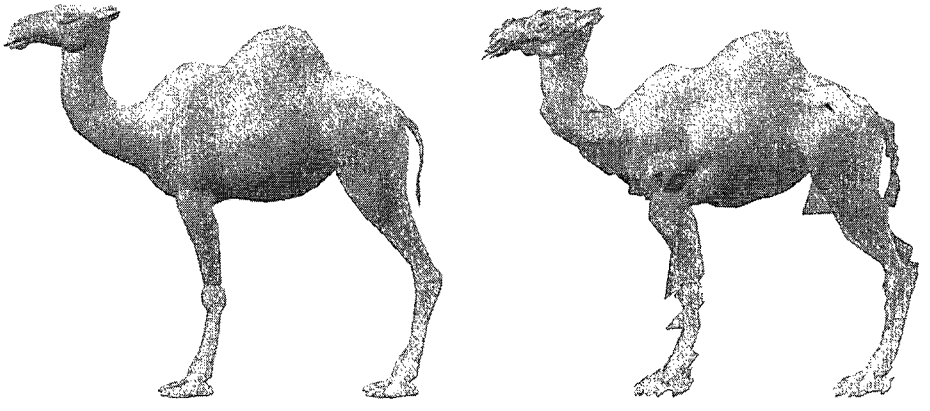


Figure 38: Correlation coefficient results for the camel model using four different strength factors and smoothing attacks with different number of iterations



(a) $\alpha = 0.003$ (b) $\alpha = 0.09$

Figure 39: Watermark perceptibility. A 16-bit watermark-embedded in the camel model with different strength factors

$$|\nabla \boldsymbol{v}_i| = \sqrt{\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \left\| (\boldsymbol{v}_i/\sqrt{d_i}) - (\boldsymbol{v}_j/\sqrt{d_j}) \right\|^2}, \tag{31}$$

$$|\nabla \boldsymbol{v}_j| = \sqrt{\sum_{\boldsymbol{v}_k \in \boldsymbol{v}_j^\star} \left\| (\boldsymbol{v}_j/\sqrt{d_j}) - (\boldsymbol{v}_k)/(\sqrt{d_k}) \right\|^2} \tag{32}$$

and $g(x) = 1/(1 + x^2/c^2)$ is the Cauchy weight function (see Figure 40) with a constant tuning parameter c that needs to be estimated. It can be shown (see [77]) that the 95% asymptotic efficiency on the standard Gaussian distribution is obtained with $c = 2.3849$ which is used in all the experimental results. Note that the visual error $D(\mathbb{M}, \widehat{\mathbb{M}})$ requires the use of two neighboring rings as depicted in Figure 41. Intuitively, the anisotropic operator $\mathcal{A}$ introduces some smoothing effect which may be explained as follows: around the sharp features of the 3D mesh where the vertex gradient magnitudes are large, the non-linear diffusion operator in Eq. (30) used to preserve the sharp feature of the 3D mesh. Moreover, in the flat regions of the 3D mesh where the vertex gradient magnitudes are relatively small, Eq. (30) is reduced to a linear operator which tends to checks the distortion of the watermarked model in these flat areas (smoothness of the watermarked model).

We used also the geometric Laplacian distance error [8, 41] defined as

$$G(\mathbb{M}, \widehat{\mathbb{M}}) = (\sum_{i=1}^{m} \|\boldsymbol{v}_i - \hat{\boldsymbol{v}}_i\| + \|\mathcal{I}(\boldsymbol{v}_i) - \mathcal{I}(\hat{\boldsymbol{v}}_i)\|^2)/(2m) \tag{33}$$

where $\mathcal{I}$ is the geometric Laplacian operator given by:

$$\mathcal{I}(\boldsymbol{v}_i) = \boldsymbol{v}_i - (\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \ell_{ij}^{-1} \boldsymbol{v}_j)/(\sum_{\boldsymbol{v}_j \in \boldsymbol{v}_i^\star} \ell_{ij}^{-1}) \tag{34}$$

where ℓ_{ij} is the Euclidian distance between $\boldsymbol{v}_i$ and $\boldsymbol{v}_j$. This visual metric consists of a sum of two error terms between the original and the watermarked vertex positions: the first term provides a measure of geometric closeness between the correct and the watermarked vertex locations, and the second term captures the smoothness properties of these vertices.

The non-linear metric error is an extension of the geometric distance error. The main difference between the two metric errors is that the second term of geometric distance error is defined in terms of a linear operator that tends to smooth more, whereas the second term of the non-linear diffusion operator that tends to smooth less and hence leads to a much better performance of the mesh geometric structures. Figure 42 shows the non-linear diffusion operator error and the geometric Laplacian distance error for three different watermarked models with different strength factors. Clearly our experimental results show that the proposed method gives low visual metric errors that guarantee the imperceptibility of the watermark.

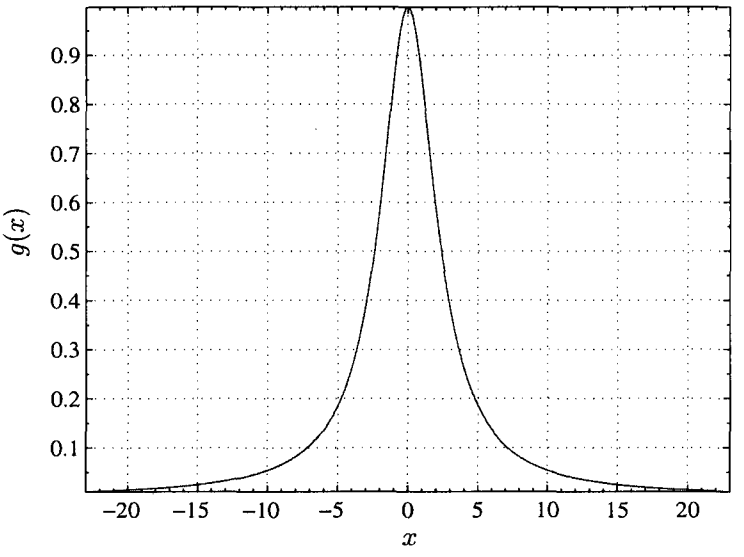


Figure 40: Cauchy weight function with $c = 2.3849$. Taken from [93]

4.4.2 Robustness

Robustness is an important factor that we need to consider when designing a watermark system for copyright protection. Attacks do not necessarily mean the removal of the watermark; they can be operations to make the watermark undetectable [72, 84]. We tested the robustness of the proposed algorithm with different 3D models (see Table 1) against various attacks including mesh transformation, mesh simplification,

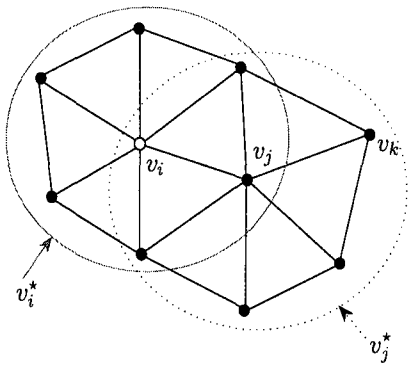


Figure 41: Illustration of two neighboring rings. Taken from [93]

additive random noise, mesh smoothing, compression, and cropping. A sequences of 16 binary digits $\{-1, 1\}$ are randomly generated and used as watermarks. In the experiments we display the attacked models with the detector response for the real watermark, and 99 randomly other generated watermarks. For all the detector response figures the correlation between the original watermark and the extracted watermark is located at 75 on the X-axis and the dotted line at 0.8 on the Y-axis represents the threshold. The threshold is chosen manually to decrease false-positive (presenting incorrectly the watermark in the model) and false-negative alarm (failing to detect the watermarked model). If the correlation is larger than 0.8, then the watermark is present. In all the experiments the strength factors that have been used are 0.01 for the Max Planck model and 0.02 for the elephant model.

Additive random noise

In order to test the robustness of the watermark, Gaussian noise was added to the watermarked mesh by summing a random vector to each vertex in the model. See Figure 43(a) for the attacked Max Planck model by Gaussian random noise ($\sigma^2 = 0.0035$). The watermark could be extracted without any loss. The detector response is illustrated in Figure 43(b). The watermark is lost when we increase the noise

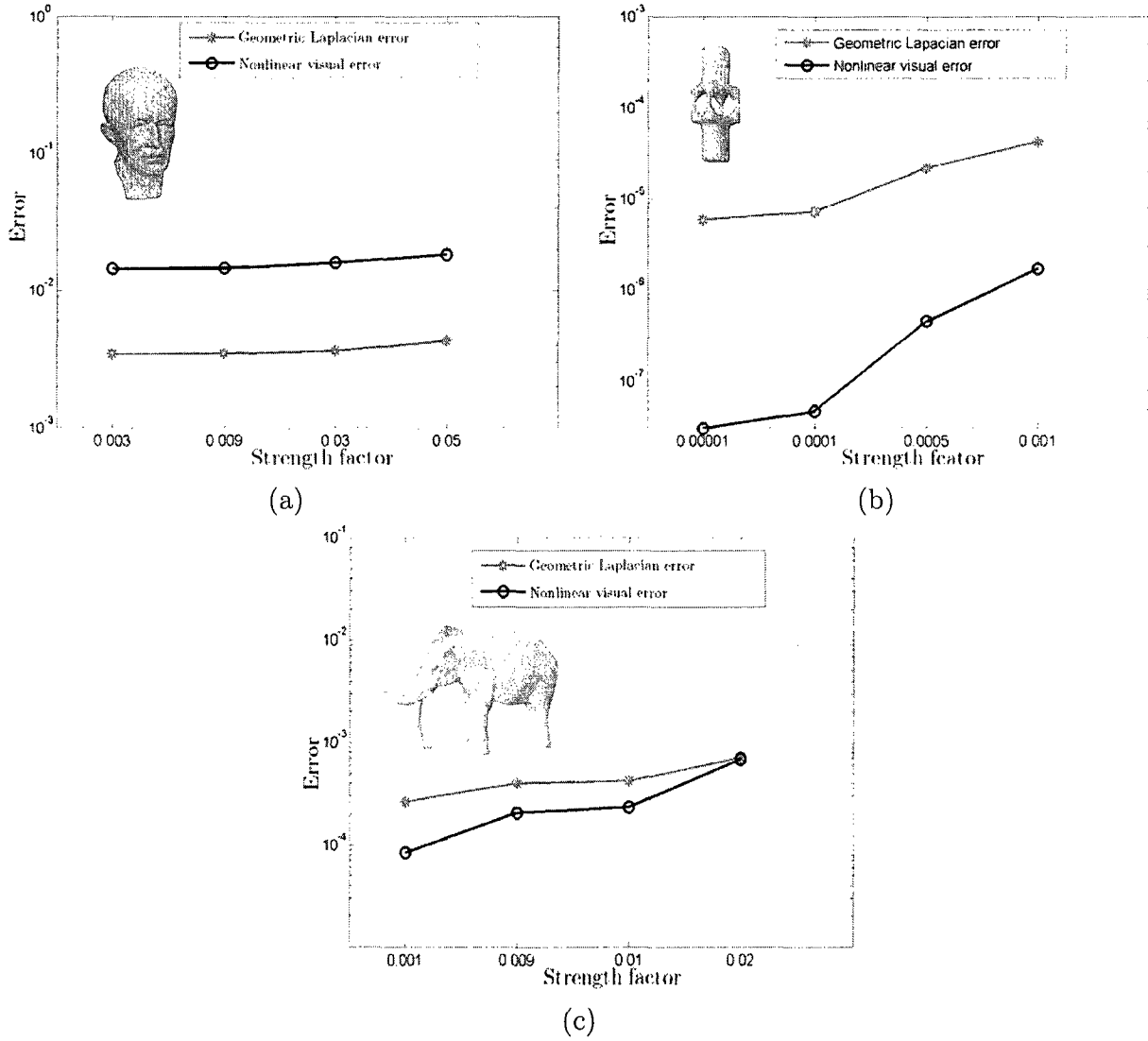
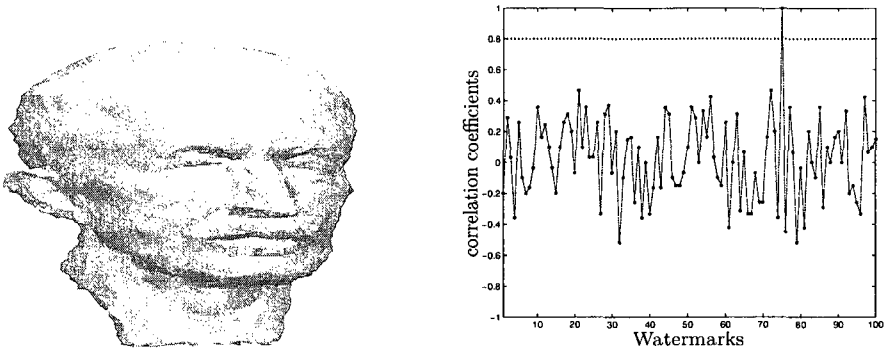


Figure 42: Non-linear visual error and the geometric Laplacian distance error changes with different strength factors (a) Max Planck model, (b) Mesh part model, and (c) Elephant model

($\sigma^2 = 0.0045$) for the Max Planck model.



(a) (b)

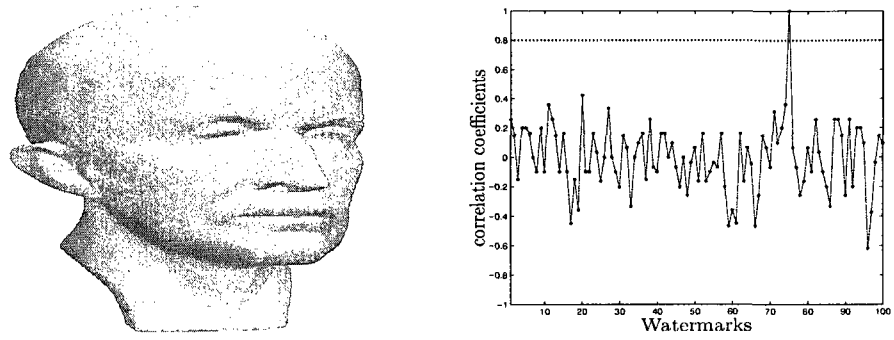
Figure 43: Robustness against additive random noise. (a) Max Planck model with ($\sigma^2 = 0.0035$) additive noise, (b) detector response

Mesh smoothing

Smoothing algorithms may be used by an attacker to destroy the watermark by moving the node geometry of the watermarked mesh. We used the Laplacian filter algorithm [82] that adjusts the location of each mesh vertex to the centroid of its neighboring vertices. Hence the high-frequency components are those that are most affected by low-pass filtering. Our proposed algorithm is robust against smoothing attack as we expected because the watermark was embedded in the low-frequency components. Figure 44(a) depicts the attacked Max Planck model by 7 smoothing iterations, and Figure 44(b) shows the detector response. As can be seen, the mesh is significantly smoothed but the watermark is still perfectly detectable.

Geometric transformations

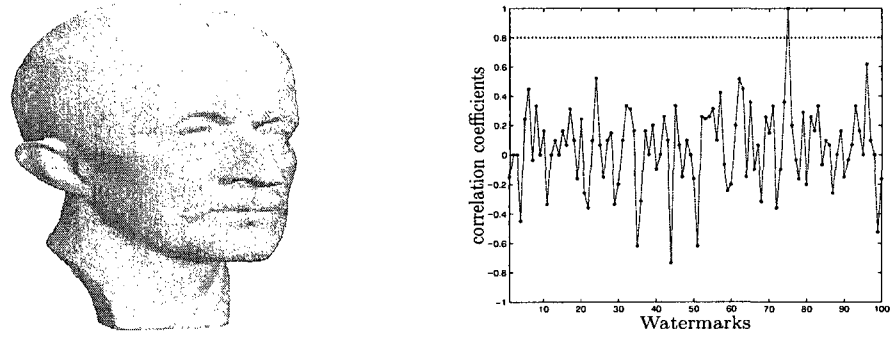
These are the simplest attacks used to test the watermark detectors. The proposed algorithm is robust against geometric attacks because the transformations applied to the mesh can be inverted using mesh registration. In Figure 45(a) the attacked Max Planck model is obtained in two steps. First, the model is scaled in the Z direction by



(a) (b)

Figure 44: Robustness against Laplacian smoothing. (a) Max Planck model after (7 iterations) of the low-pass filter, (b) detector response

a factor of 2. Second, the scaled model is rotated around Y-axis by 20^o. Figure 45(b) depicts the watermark extraction response after the registration process was applied. Clearly the detector is still able to recover the watermark.



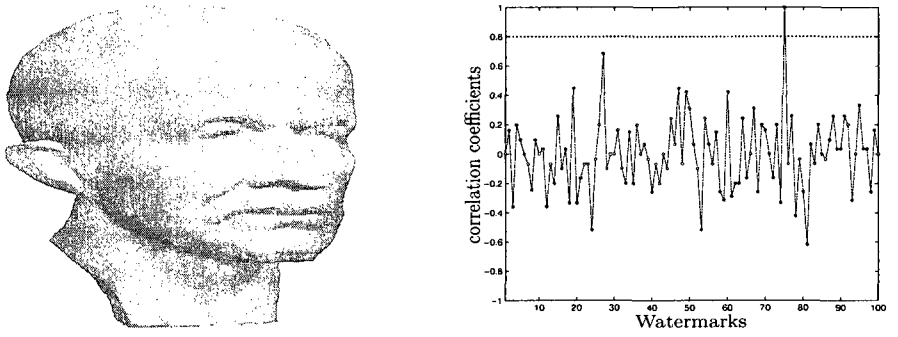
(a) (b)

Figure 45: Robustness against geometric transformation. (a) Max Planck model is scaled in Z direction by factor of 2 then rotated by 20^o around Y-axis, (b) detector response

Mesh compression

Mesh compression has recently become one of the most effective attacks because the new compression techniques [33, 40, 41] reach a very significant compression ratio with very small loss in the mesh quality. We evaluated the robustness of our method

against a compression attack [41]. The proposed method is robust against compression because the watermark is embedded in the spectral coefficient of the compressed mesh. Figure 46(a) depicts the compressed Max Planck model constructed with 3000 basis functions from the original mesh of 5040 basis functions. The detector response is shown in Figure 46(b).

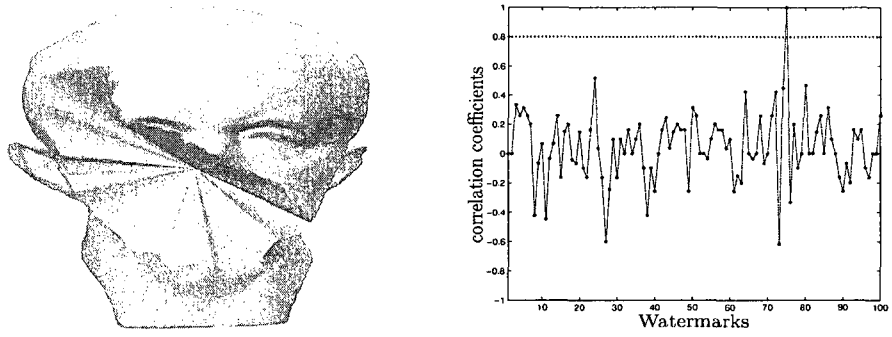


(a) (b)

Figure 46: Robustness against compression attack. (a) compressed Max Planck model of 3000 basis functions, (b) detector response

Mesh cropping

This technique may be used by an attacker to destroy the watermark by removing part of the watermarked mesh. We verified the robustness of the proposed scheme against mesh cropping by trying to extract the watermark from the cropped 3D mesh. Since the watermark is embedded repeatedly using mesh partitioning the watermark can be fully recovered from the deteriorated cropped mesh. Figure 47(a) depicts the cropped Max Planck model (600 vertices have been removed). The watermark is recovered perfectly from the cropped model as it is shown in the detector response in Figure 47(b).

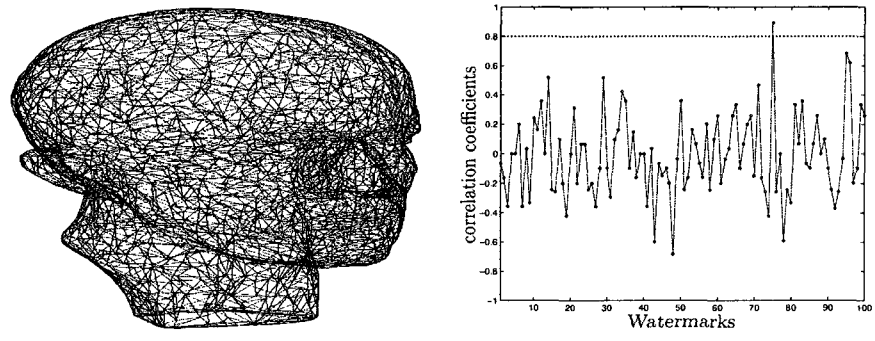


(a) (b)

Figure 47: Robustness against cropping attack. (a) cropped 600 vertices from Max Planck model, (b) detector response

Mesh simplification

This method may also be used by an attacker to reduce the number of faces of the 3D mesh [30]. This reduction could remove or destroy the watermark. See Figure 48(a) for the simplified Max Planck model. The mesh is simplified down from 5040 vertices and 10067 faces to 2502 vertices and 5000 faces. Our proposed method is robust against the simplification attack because of the remeshing process. The detector response for the attacked mesh in Figure 48(a) is illustrated in Figure 48(b).



(a) (b)

Figure 48: Robustness against simplification attack. (a) Max Planck model simplified down to 2502 vertices and 5000 faces, (b) detector response

We also tested the performance of our proposed algorithm using a combination

of the previous attacks. Figure 49 (a,b) show the watermarked Max Planck model with multiple attacks. In Figure 49(a) the watermarked model is passed through low-pass filtering (7 iterations), and then a cropping attack has been applied to remove 540 vertices from the smoothed mesh. Figure 49(b) depicts the attacked model after adding additive random noise of ($\sigma^2 = 0.0025$) and being simplified down to 80% of the original vertices. In both cases the proposed algorithm was able to recover the watermark fully (see the detector responses for (a,b) in (c,d) respectively). More experiments with different models are shown in Figure 50.

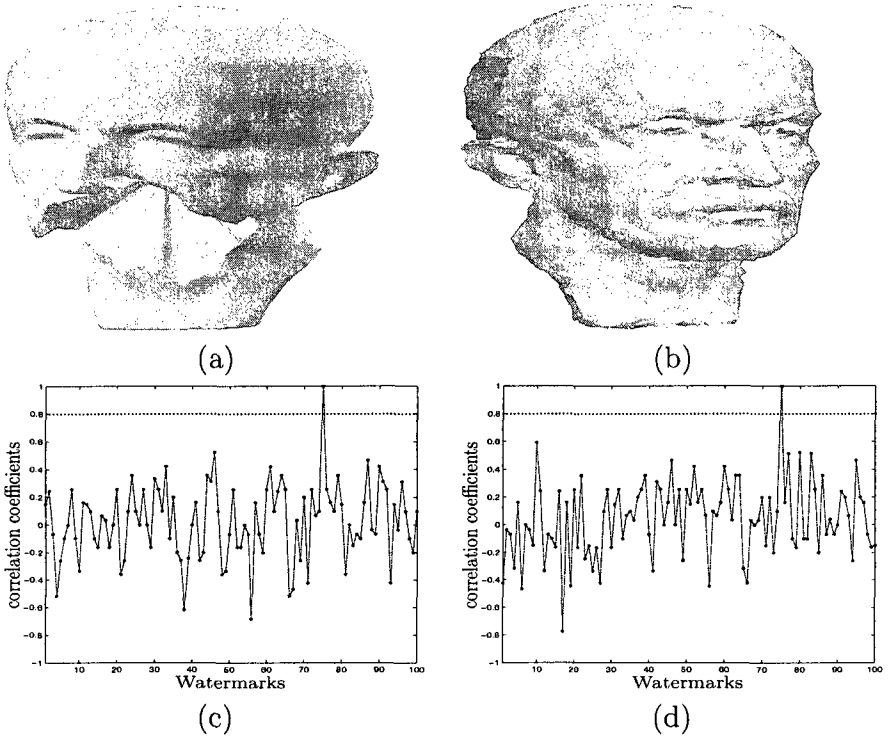


Figure 49: Robustness against multiple attacks. (a) Max Planck model attacked with smoothing (8 iterations) and cropping 540 vertices. (b) Max Planck model attacked with additive noise ($\sigma^2 = 0.0025$) then simplified to (80%) of original faces. (c), (d) detector responses for (a), (b) respectively

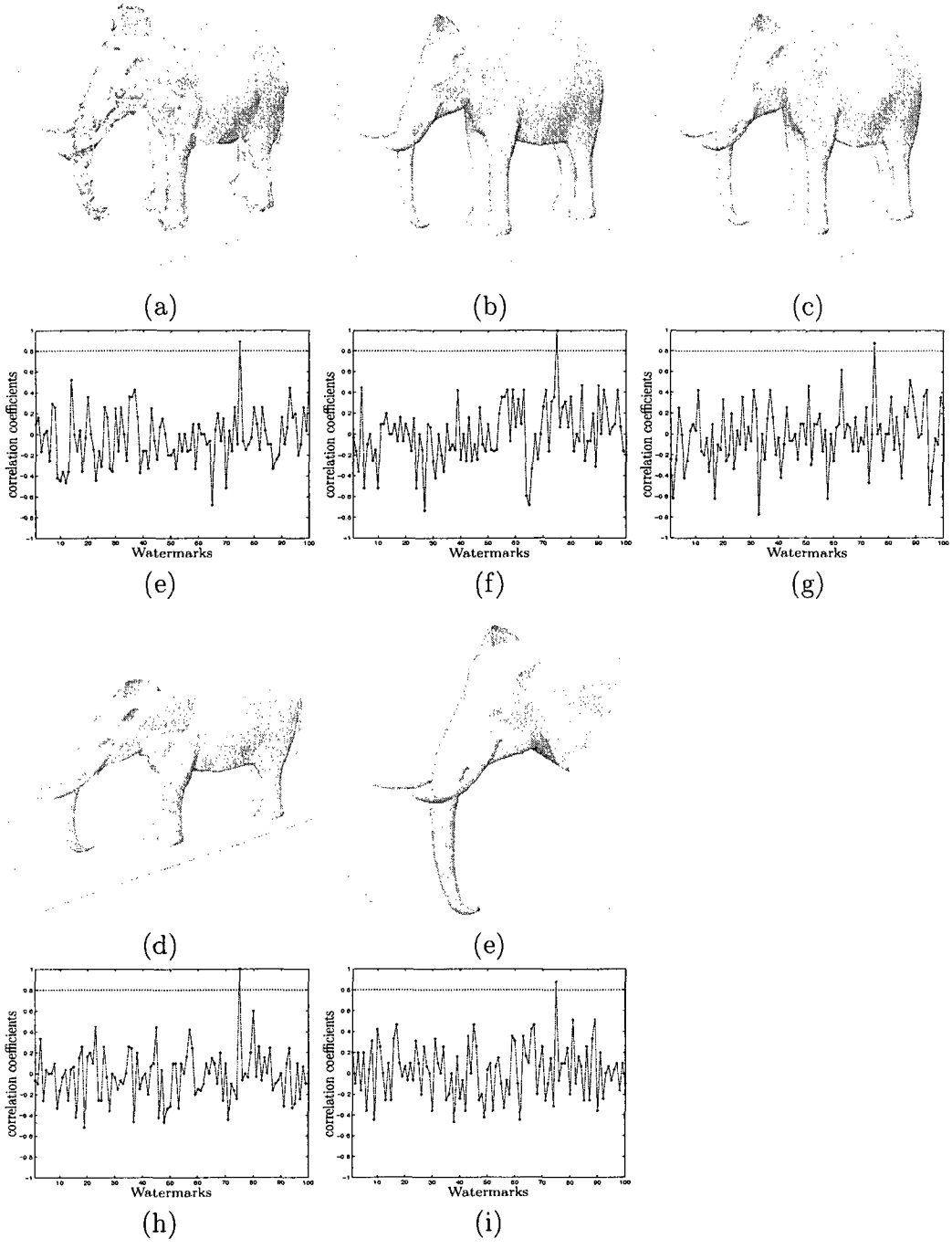


Figure 50: Watermarked elephant model with different attacks and their corresponding detector responses: for each each attack the correlation coefficient between the extracted watermark and 99 different random watermarks are shown (75 on X-axis is the correlation with the real watermark). (a), (e) additive noise ($\sigma^2 = 0.007$), (b), (f) low-pass filter 9 iterations, (c), (g) compression 1500 basis functions, (d), (h) scaling in X direction, (e), (i) cropping 1700 vertices and smoothing 7 iterations.

4.4.3 Comparisons with existing techniques

We conducted several experiments to compare the robustness of the proposed method with related existing techniques that use mesh spectral coefficients vectors to embed the watermark in the frequency domain, and in particular with watermarking 3D meshes in the spectral domain [70] and its extension [69]. In our experiments, we embedded the watermark in all the spectral coefficients vectors. However in [70] the lowest 5 spectral coefficients are used to realign the original and the watermarked mesh before applying the watermark extraction process. So in [70] the watermark is embedded in all the $(n-5)$ higher spectral coefficients.

In our experimental comparisons we used six different 3D models: camel, elephant, bunny, cow, rocker-arm, and hand, and a watermark sequence of 16 bits generated randomly of $\{-1, 1\}$. For each attack we used various strengths. Table 8 shows the comparison results of the proposed watermarking scheme with the methods introduced in [69, 70] against smoothing attack. Three different numbers of iterations were applied. Clearly our proposed scheme performs the best in terms of the robustness against the smoothing attack. An example of the smoothing attack is shown in Figure 51. 4 against the noise attack is shown in Table 5 where Gaussian random noise was added to each vertex of the watermarked model with three different standard deviations. Figure 52 shows the noise attack on different models. All the spectral techniques have good resistance against noise attacks because the watermark is embedded in all the spectral coefficients. To evaluate the robustness of the three techniques against compression attack, we first simplified the mesh by reducing the number of faces to 10000 for all the large the 3D models, then the 3D model is compressed using the algorithm proposed in [41]. The third column in Table 6 indicates the number of the basis functions used to compress the watermarked models. The correlation coefficients shown in Table 6 clearly demonstrate that our proposed scheme outperforms the existing techniques, and an example is shown in Figure 53.

Table 4: Comparison results: Robustness against smoothing attack. The boldface numbers indicate the best correlation coefficients

Model	α	# of iterations	Corr. Proposed	Corr. [69]	Corr. [70]
Camel	0.02	11	**1**	0.6831	0.5164
		13	**1**	-0.5222	-0.4229
		15	**0.8783**	-0.9853	-0.6181
Elephant	0.02	10	**1**	0.5146	0.2
		12	**0.8783**	0.0667	-0.2437
		15	**0.8783**	-0.3333	-0.3333
Bunny	0.0002	10	**1**	0.6	0.4667
		12	**1**	0.3333	-0.4472
		15	**1**	-0.2	-0.5164
Cow	0.01	10	**1**	0.6831	0.3333
		15	**0.8783**	0	0.0667
		18	**0.7746**	-0.5164	-0.7746
Rocker-arm	0.0001	10	**1**	0.8783	0.8783
		13	**1**	-0.5164	-0.7333
		15	**0.8783**	-0.7333	-0.8704
Hand	0.09	35	**1**	0.8704	0.7454
		40	**1**	0.7746	0.6202
		45	**1**	0.7454	0.6

To evaluate the robustness against smoothing and simplification attacks, we used various simplification rates and a fixed smoothing iteration number. Table 7 demonstrates that the proposed method performs better than the other techniques against the combination of simplification and smoothing attack. This better performance is in fact consistent with a variety of 3D models used for experimentation.

4.4.4 Computational complexity

The computational complexity and memory requirements of the watermark embedding process of our proposed scheme is the same as for the other two schemes proposed in [69, 70]. The computation of the spectral basis functions of the Laplacian matrix is the most expensive and it is common to all the spectral domain schemes. The

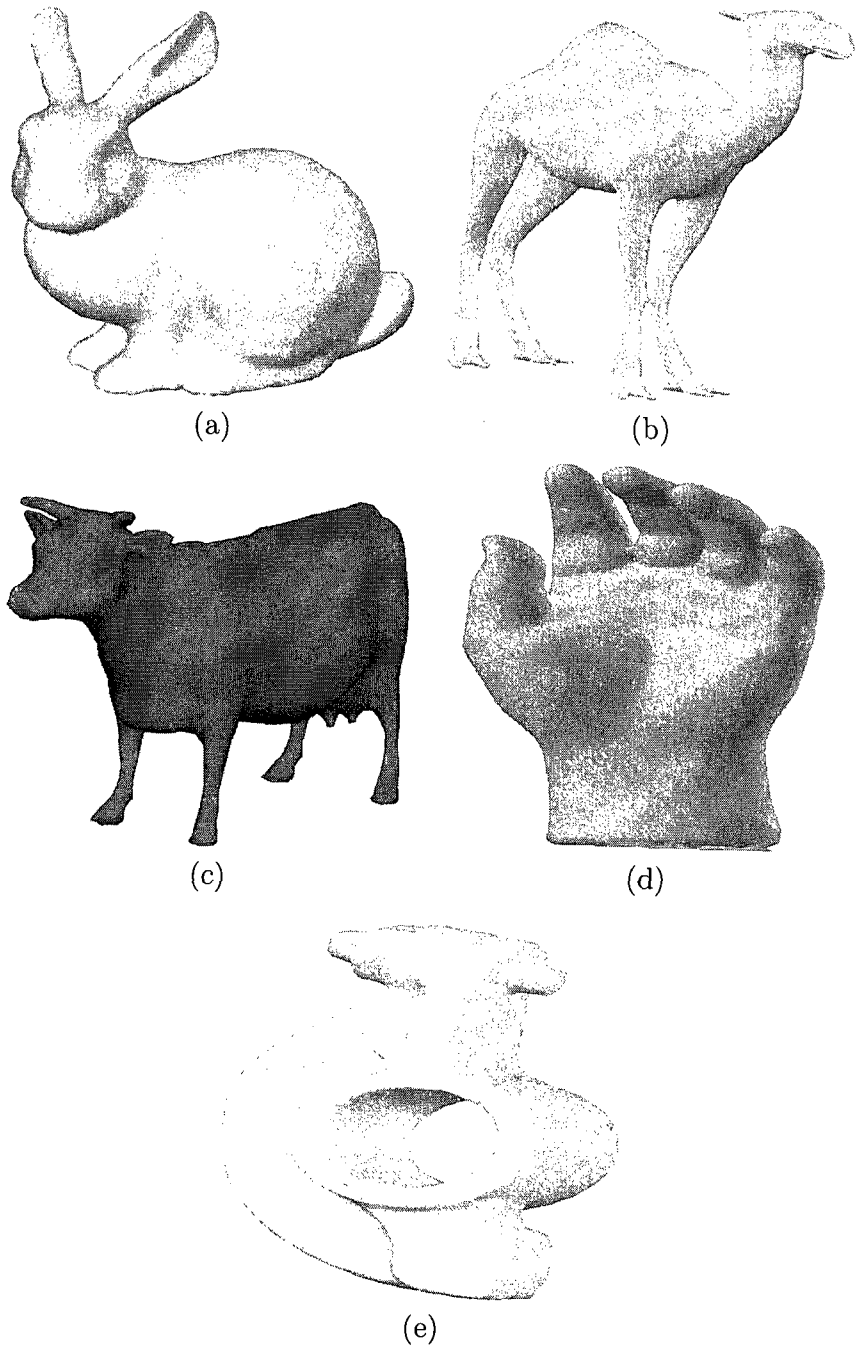


Figure 51: Smoothing attack with different iterations: (a) 12, (b) 13, (c) 10, (d) 30, (e) 12.

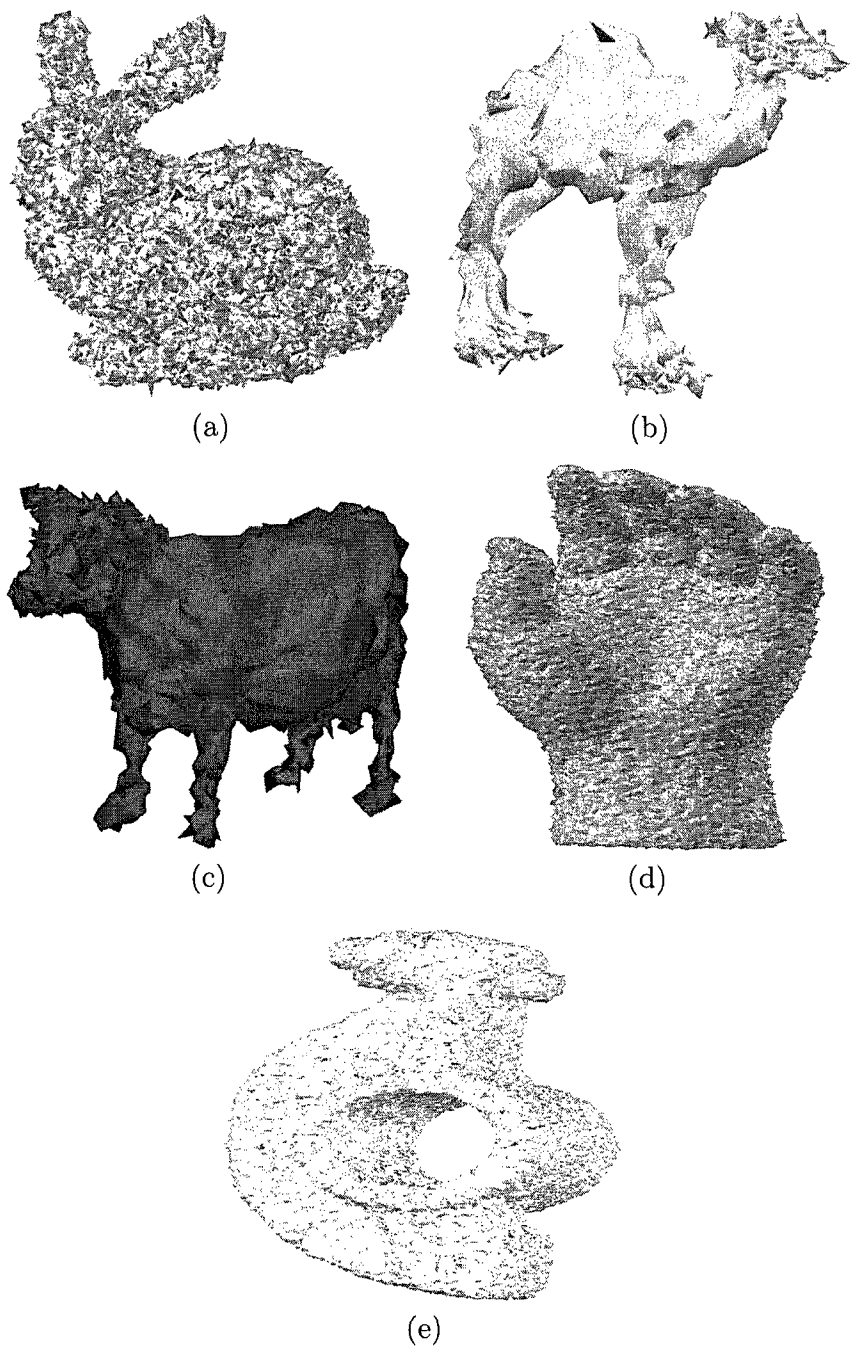


Figure 52: Gaussian noise attack with different standard deviations: (a) 0.013, (b) 0.013, (c) 0.0095, (d) 0.0095, (e) 0.0095.

Table 5: Comparison results: Robustness against additive noise attack. The boldface numbers indicate the best correlation coefficients

Model	α	σ^2	**Corr. Proposed**	**Corr. [69]**	**Corr. [70]**
Camel	0.02	0.013	**1**	0.8783	0.8783
		0.015	**0.8783**	**0.8783**	0.8704
		0.017	**0.8704**	0.8704	0.8704
Elephant	0.02	0.0095	**1**	**1**	0.8783
		0.011	**0.8783**	0.8704	0.7333
		0.013	0.7333	**0.7454**	0.7333
Bunny	0.0002	0.013	**1**	**1**	0.8783
		0.015	0.8783	**1**	0.8704
		0.017	**0.8783**	0.8704	0.7746
Cow	0.01	0.0095	**1**	**1**	**1**
		0.012	**0.8783**	0.8704	1
		0.015	0.7333	**0.7746**	0.8704
Rocker-arm	0.0001	0.095	**1**	**1**	**1**
		0.012	0.8783	**1**	0.8783
		0.015	0.8704	**0.8783**	0.8783
Hand	0.09	0.0095	**1**	**1**	0.8783
		0.012	**0.8783**	**0.8783**	0.8704
		0.015	0.5919	**0.6**	0.5222

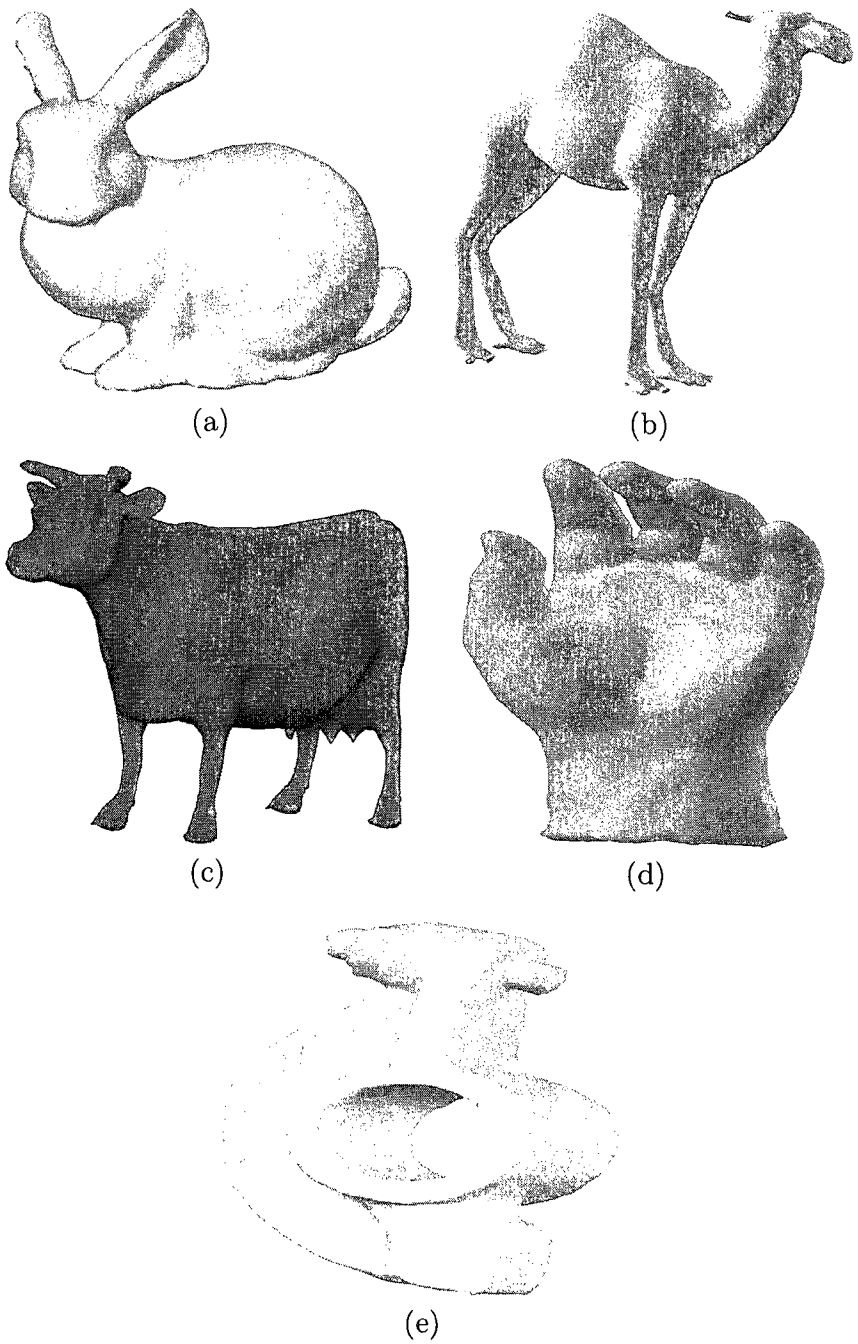


Figure 53: Compression attack with different numbers of basis functions: (a) 800, (b) 250, (c) 400, (d) 1000, (e) 600.

Table 6: Comparison results: Robustness against compression attack. The boldface numbers indicate the best correlation coefficients

Model	α	**# Comp. basis**	**Corr. Proposed**	**Corr. [69]**	**Corr. [70]**
Camel	0.02	300	**1**	0.8783	0.8783
		250	**0.8783**	0.7746	0.6931
		200	**0.6831**	0.6	0.5222
Elephant	0.02	1700	**1**	0.6181	0.8783
		1500	**1**	0.5164	0.7746
		1300	**0.7746**	0.4229	0.6831
Bunny	0.0002	800	**1**	0.731	0.8783
		600	**1**	0.5164	0.8704
		500	**0.8783**	0.3133	0.5164
Cow	0.01	500	**1**	0.7746	0.5146
		400	**0.8783**	0.5164	0.333
		300	**0.7746**	0.4229	0.2582
Rocker-arm	0.0001	800	**1**	0.8783	0.8783
		600	**1**	0.6831	0.6831
		500	**0.8783**	0.3587	0.5164
Hand	0.09	1200	**1**	0.8783	0.8783
		1000	**0.8783**	0.7333	0.7333
		800	**0.7746**	0.4667	0.6181

Table 7: Comparison results: robustness against smoothing and simplification attacks. The boldface numbers indicate the best correlation coefficients

Model	α	**Simpl. rate**	**# of iter.**	**Corr. Proposed**	**Corr. [69]**	**Corr. [70]**
Camel	0.02	20%	9	**0.8783**	0.7746	0.6541
Elephant	0.02	30%	8	**1**	0.8704	0.3456
Bunny	0.0002	35%	7	**1**	0.6	0.522
Cow	0.01	20%	9	**0.8783**	0.7746	0.5146
Rocker-arm	0.0001	35%	8	**1**	0.8783	0.8704
Hand	0.09	30%	25	**1**	**1**	0.8783

spectral analysis involves the computation of the basis functions of the Laplacian matrix of the 3D mesh. We partition the mesh into sub-meshes of n vertices each. The algorithm requires $\mathcal{O}(n^3)$ operations and stores $\mathcal{O}(n^2)$ elements of the calculated basis functions for straightforward implementations. Significant improvement could be further achieved by using the fast multi-resolution method [34] on the Laplacian matrix with an overall time complexity of $\mathcal{O}(n)$.

Chapter 5

3D Watermarking Technique Using NMF

In this chapter we propose a robust 3D polygonal mesh watermarking technique [7] based on the spectral eigen-decomposition and nonnegative matrix factorization. The core idea behind our technique is to apply the NMF to small blocks of the spectral coefficient matrix of the 3D polygonal meshes. The proposed scheme improves the performance of the data embedding system, perceptual invisibility and it is resistant to a variety of the most common attacks.

The remainder of this chapter is organized as follows. In the next Section we introduce the proposed approach and describe in detail the watermark embedding and extraction algorithms. In Section 5.2, we present some experimental results, and we show the robustness of our method against the most common attacks.

5.1 NMF Overview

One major drawback of SVD is that the basis vectors may have both positive and negative components, and the data are represented as linear combination of these vectors with positive and negative coefficients. In many applications, the negative

coefficients contradict physical realities. To address this problem, the NMF approach was proposed to search for a representative basis with only nonnegative vectors. The NMF [49, 50] can be formulated as follows: Given a nonnegative matrix $\boldsymbol{C}$ of size $m \times m$, we can approximately factorized $\boldsymbol{C}$ into the product of two nonnegative matrices $\boldsymbol{B}$ and $\boldsymbol{H}$ with sizes $m \times r$ and $r \times m$ respectively; that is,

$$\boldsymbol{C} \approx \boldsymbol{BH}, \quad where \quad r \leq m. \tag{35}$$

The nonnegative matrix $\boldsymbol{B}$ contains the NMF basis vectors, and the nonnegative weight matrix $\boldsymbol{H}$ contains the associated coefficients (nonnegative weights) [31]. To measure the quality of the approximation factorization $\boldsymbol{C} \approx \boldsymbol{BH}$, a cost function between $\boldsymbol{C}$ and $\boldsymbol{BH}$ needs to be optimized subject to nonnegativity constraints on $\boldsymbol{B}$ and $\boldsymbol{H}$. This is done by minimizing the $\mathcal{I}$- divergence given by

$$\mathcal{I}(C\|BH) = \sum_{ij}(C_{ij} \log \frac{C_{ij}}{(BH)_{ij}} - C_{ij} + (BH)_{ij}), \tag{36}$$

which yields the following multiplicative update rules:

$$H_{kj} \leftarrow H_{kj} \frac{\sum_i B_{ik} C_{ij}/(BH)_{ij}}{\sum_i B_{ik}}, \tag{37}$$

$$B_{ik} \leftarrow B_{ik} \frac{\sum_j H_{kj} C_{ij}/(BH)_{ij}}{\sum_j H_{kj}}, \tag{38}$$

where the matrices $\boldsymbol{B}$ and $\boldsymbol{H}$ are initialized as nonnegative random matrices, and the updates are done alternatively; that is, after updating one row of $\boldsymbol{H}$, we need to update the corresponding column of $\boldsymbol{B}$. In other words, we should not update the whole matrix $\boldsymbol{H}$ first followed by an update of the matrix $\boldsymbol{B}$. The NMF algorithm is therefore an iterative optimization algorithm that modifies at each iteration the nonnegative basis functions (i.e., columns of $\boldsymbol{B}$) and encodings (i.e., $\boldsymbol{H}_{kj}$) until convergence.

5.2 Proposed 3D Watermarking Scheme

Motivated by the good performance of the NMF watermarking techniques for 2D images [31] and spectral watermarking methods for 3D models [4, 70], we propose a robust imperceptible watermarking approach using the 3D mesh spectra and the NMF. Extensive numerical experiments are performed to demonstrate the much improved performance of the proposed method. The visual error is evaluated by computing the Laplacian operator [41] between the original 3D models and the watermarked models.

In this section, we describe the main steps of the proposed watermark embedding and extraction methods.

5.2.1 Watermark embedding process

The goal of our proposed approach may be described as embedding the watermark in the frequency domain, represented by the nonnegative weights of the NMF technique applied to the spectral coefficient matrix of a 3D mesh. The watermark embedding process description is shown in Algorithm 5.1. We used an identical watermark for all the sub-meshes. Figure 54 shows two different 3D models and their corresponding watermarked models. Clearly the difference between the original and the watermarked models is not noticeable to the human observer.

5.2.2 Watermark extraction process

In order to extract the watermark, our algorithm requires the original unwatermarked model $\mathbb{M}$ as well as the watermarked, probably attacked, model $\widehat{\mathbb{M}}$. Before applying the extraction process, we need to estimate the optimal rotation, scaling and translation to bring $\widehat{\mathbb{M}}$ to its initial scale and location if it has been changed. We apply the registration process (see Section 4.3.2 i) which is very important in order to extract the watermark successfully. After registration, a remeshing (Section 4.3.2 ii) is

Algorithm 5.1 Watermark embedding algorithm

Input *Original Mesh* $\mathbb{M}$ *and a random watermark vector* $\boldsymbol{w}$ *of length* m.
Output: *Watermarked mesh* $\widehat{\mathbb{M}}$.
$\diamond$ The 3D mesh is partitioned into smaller sub-meshes of size n vertices.
for each sub-mesh $\mathbb{S} \subset \mathbb{M}$ **do**

1- Compute the Laplacian matrix $\boldsymbol{L}$ of size $n \times n$.

2- Compute the eigenvalues and the associated eigenvectors (basis functions) of $\boldsymbol{L}$.

3- Project the mesh vertices onto the basis functions to get the spectral coefficients matrix of the original sub-mesh $\mathbb{S}$.

$$\boldsymbol{C} = \boldsymbol{B}^T \boldsymbol{V} \tag{39}$$

4- Add a constant value ε to the spectral matrix $\boldsymbol{C}$ to obtain $\widehat{\boldsymbol{C}}$, where $\widehat{c}_i \geq 0$.

5- Partition the spectral matrix $\widehat{\boldsymbol{C}}$ into blocks of size $\ell \times \ell$ for watermarking.

6- Apply NMF to each block K_s: $K_s = B_\ell H_\ell$, followed by singular value decomposition to the weight matrix:

$$H_\ell = U_\ell \Sigma_\ell V_\ell^{'} \tag{40}$$

7- Modify λ_{max} according to:

$$\lambda_i^d = \lambda_{max} + \alpha w_i \tag{41}$$

where λ_{max} denotes the largest SV of H_ℓ , λ_i^d denotes the distorted SV of H_ℓ, w_i is the i^{th} watermark element, and α is the watermark strength factor.

8- Use all the distorted blocks $K^d = B_\ell H_\ell^d$, where $H_\ell^d = U_\ell \Sigma_\ell^d V_\ell^{'}$ to obtain the modified spectral coefficient matrix.

9- Subtract the constant value ε from all the modified spectral coefficients to obtain $\widehat{\boldsymbol{C}}$.

10- Express the watermarked sub-mesh vertices in the subspace using the modified spectral coefficients. Thus,

$$\boldsymbol{V}_w^T = \widehat{\boldsymbol{C}}^T \boldsymbol{B}^T = \sum_{i=1}^{n} \widehat{\boldsymbol{c}}_i^T \boldsymbol{b}_i^T \tag{42}$$

where $\boldsymbol{V}_w^T$ is the watermarked sub-mesh vertex matrix.
end for

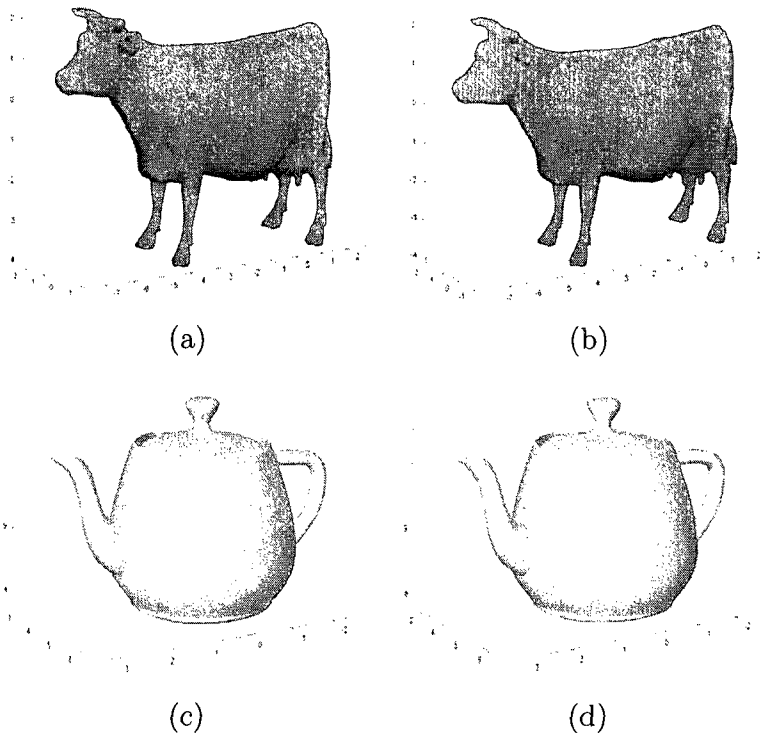


Figure 54: (a)-(c) Original 3D models and their corresponding watermarked models (b)-(d). Cow model with (2903 vertices, 5804 faces) and teapot model with (3241 vertices, 6315 faces). The watermark strength factor $\alpha = 0.03$ for both models

usually necessary to deal with the attacks that may modify the mesh topology.

Algorithm 5.2 Watermark extraction algorithm

Input *Original and watermarked meshes* $\mathbb{M}$ *and* $\widehat{\mathbb{M}}$.
Output: *Extracted watermark vector* $\overline{\boldsymbol{w}}$.
$\diamond$ The 3D meshes are partitioned into smaller sub-meshes using the same procedure as in the embedding process.
for each sub-mesh **do**
1- Apply the first six steps of the embedding process to the initial and the watermarked sub-meshes.
2- Extract the singular values of the visual watermark as follows:

$$\widehat{w}_i = (\widehat{\lambda^i} - \lambda^i)/\alpha \tag{43}$$

where λ^i and $\widehat{\lambda^i}$ are the original and the watermarked singular values respectively.
3- Construct the watermark vector $\widehat{w_\ell}$ using the extracted watermark values $\widehat{w}_i$, where $i \leq m$
end for
$\diamond$ Find the average watermark vector $\overline{w}$ from the n sub meshes.

5.3 Experimental Results

Our experiments were performed using a variety of 3D models represented as triangle meshes. We conducted experiments to test the imperceptibility of the watermark and robustness against attacks.

5.3.1 Robustness

To assess the robustness of our proposed method, we applied different attacks to the watermarked 3D models. These attacks include mesh transformation, mesh simplification, additive random noise, mesh smoothing, compression, and combination of these attacks. In our experiments, we used a random watermark vector of 64 bits. For each of the attacks, we display the attacked 3D model and the detector response for the real watermark, as well as another 499 randomly generated watermarks. For all

the detector responses, the correlation coefficient between the original watermark and the average extracted watermarks is located at 250 on the x-axis. The gray dotted line at 0.7 on the y-axis represents the threshold. Figures 55-58 show the watermarked 3D models with different kinds of attacks and their corresponding average extracted watermarks.

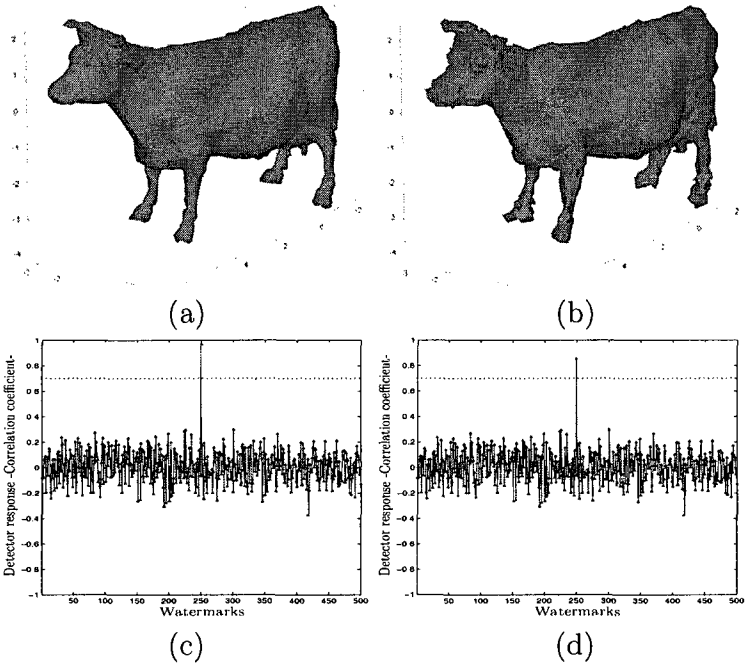


Figure 55: Robustness against additive Gaussian random noise. (a), (b) noisy cow model with noise standard deviation $\sigma = 0.25$ and $\sigma = 0.65$ respectively, (c), (d) detector responses

The Gaussian noise attack is shown Figure 55 (a)-(b) with $\sigma = 0.25$ and $\sigma = 0.65$ respectively. Figure 56 shows the results of applying 2 and 8 iterations of the Laplacian smoothing filter. Figure 57 addresses the geometric attacks.

We evaluated the robustness of our method against the spectral mesh compression attack [41]. See Figure 58 (a)-(b) for the compressed cow model constructed with 1500 and 500 basis functions from the original mesh of 2000 basis functions. Figure 58 (f)-(h) demonstrate the resilience of the watermark against the mesh simplification

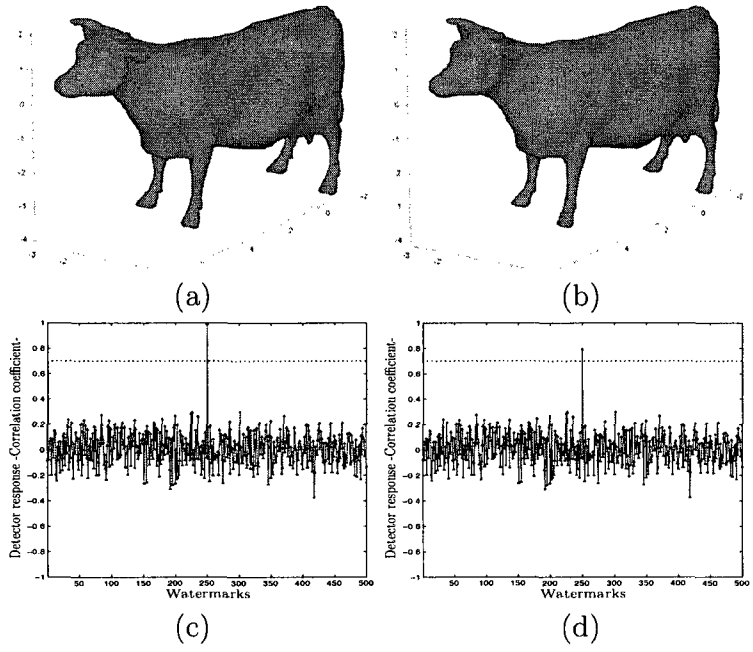


Figure 56: Robustness against Laplacian smoothing attack. (a), (b) cow model after 2 and 8 iterations of the low-pass filtering. (c), (d) detector responses for (a), (b) respectively

attack.

Figures 60-62 depict the robustness of the cow, camel and teapot models to noise, smoothing, and compression attacks. Different strength factors, noise rates, smoothing iterations, and compression rate have been used. The results obtained from our experiments demonstrate the robustness of the proposed algorithm against the commonly used attacks in the 3D domain.

5.3.2 Imperceptibility

In order to achieve high visual quality of the watermarked model, the watermark strength factor α should be taken into consideration. Figure 59 shows an example of the influence of the strength factor on the watermark perceptibility. The watermark

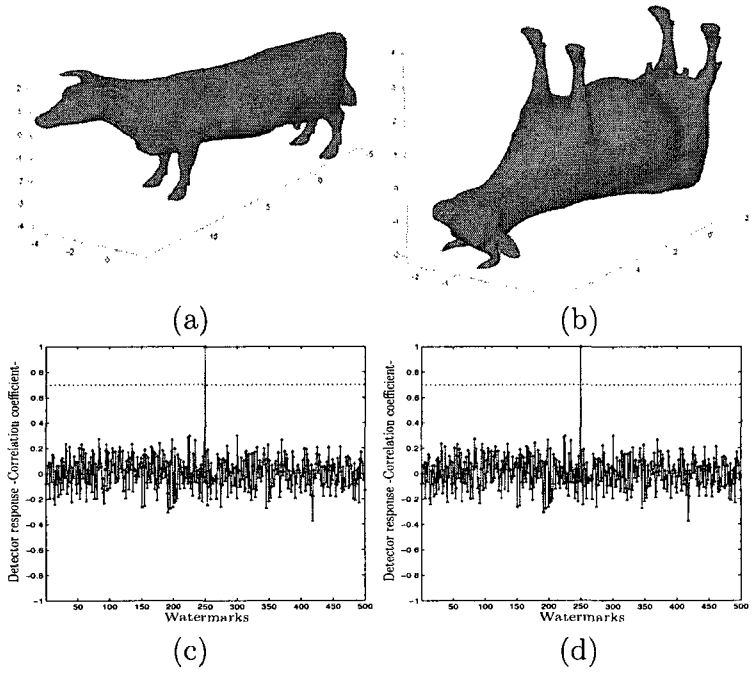


Figure 57: Robustness against transformation attacks. (a) cow model is scaled in x direction by factor of 2. (b) cow model is rotated by 180^o around x-axis. (c), (d) detector responses for (a), (b) respectively

embedded in the cow and the teapot models with strength factor 0.1 is not only perceptible to the human observer but also may destroy the overall geometric structure of those models as shown in Figure 59.

To quantify the imperceptibility of the proposed approach, we used the geometric Laplacian distance error [41]. Table 1 shows the geometric Laplacian distance error for three different watermarked models with different strength factors. Clearly our experimental results show that the proposed method gives low visual metric errors which guarantee the imperceptibility of the watermark.

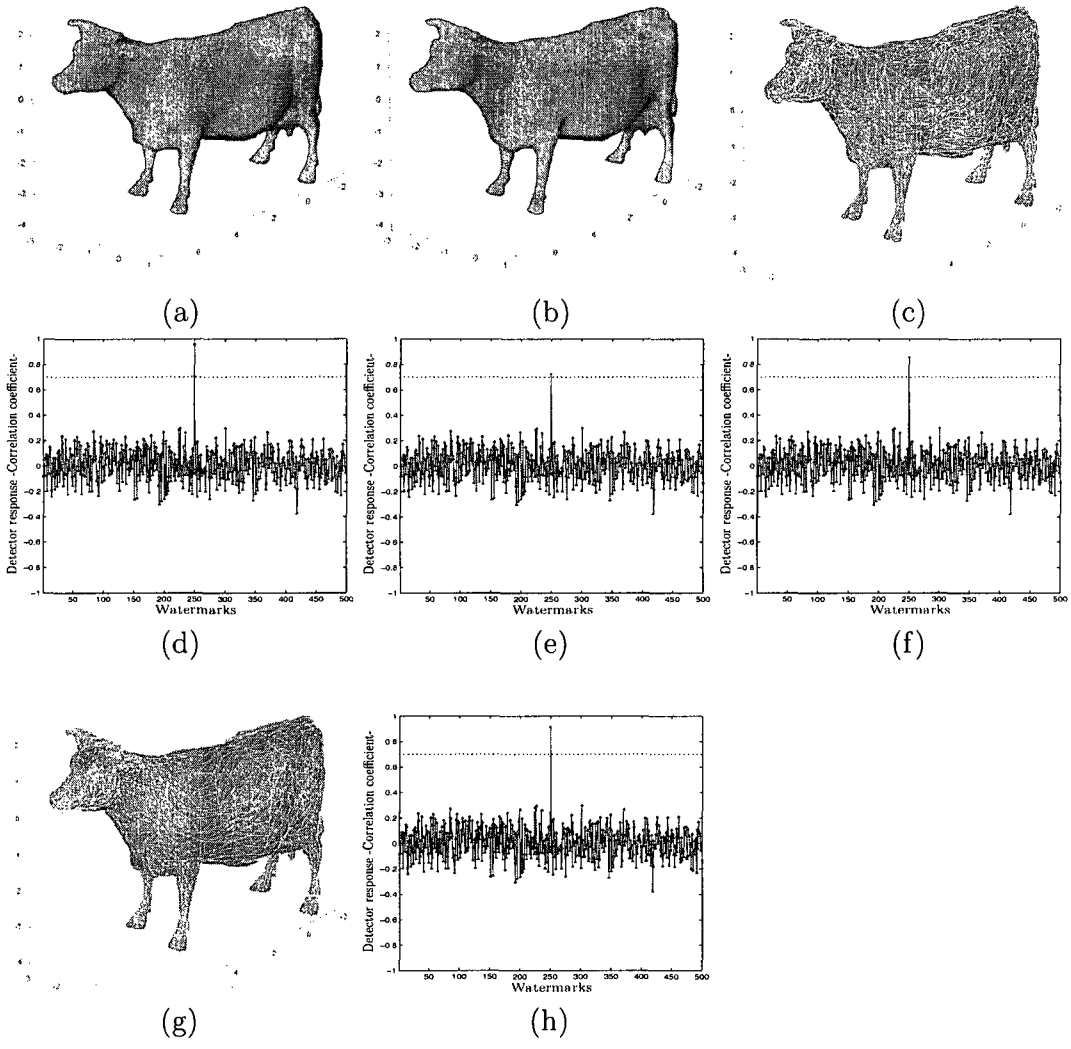
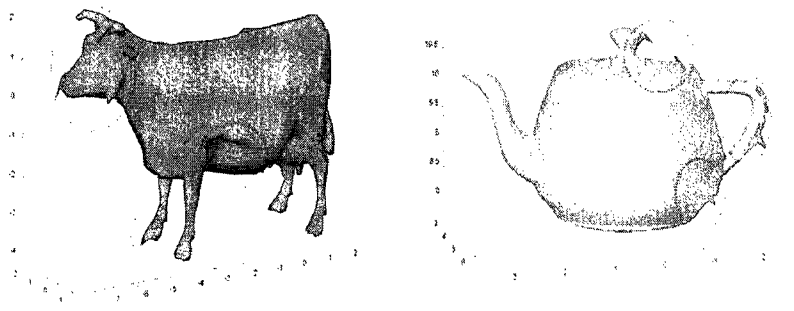


Figure 58: Robustness against compression and mesh simplification attacks. (a), (b) compressed cow model of 1500, 500 basis functions. (c), (g) cow model simplified down to 90% and 70% of the original vertices. (d),(e),(f) and (h) detector responses for (a), (d), (c), and (g) respectively

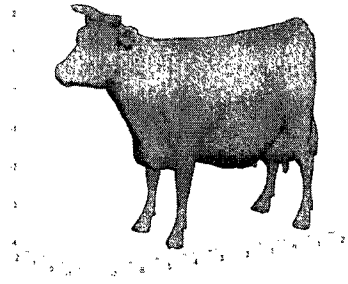


(a) $\alpha = 0.1$ (b) $\alpha = 0.1$

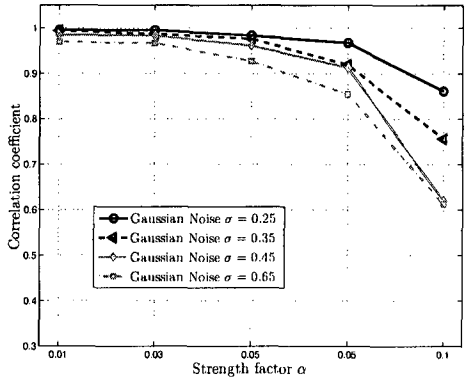
Figure 59: Watermark perceptibility. The watermark embedded in the cow and teapot models have high strength factors

Table 8: The geometric Laplacian distance error results

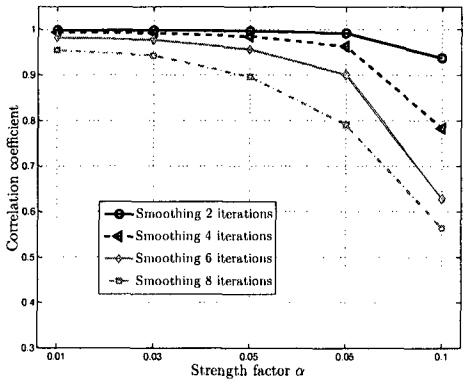
3D Model	**Strength factor α**	**Geometric Laplacian error**
Cow	0.01	2.4×10^{-5}
	0.03	2.2×10^{-4}
	0.05	6.1×10^{-4}
	0.08	1.2×10^{-3}
	0.1	2.4×10^{-3}
Camel	0.01	2.4×10^{-5}
	0.03	2.2×10^{-4}
	0.05	6.2×10^{-4}
	0.08	1.3×10^{-3}
	0.1	2.5×10^{-3}
Teapot	0.01	2.3×10^{-5}
	0.03	2.1×10^{-4}
	0.05	5.9×10^{-4}
	0.08	2.2×10^{-3}
	0.1	5.1×10^{-3}



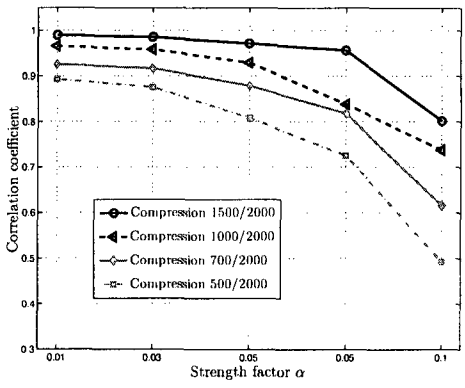
Cow model



Gaussian noise attack

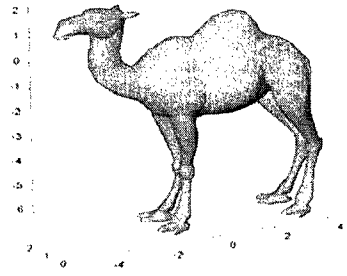


Smoothing attack

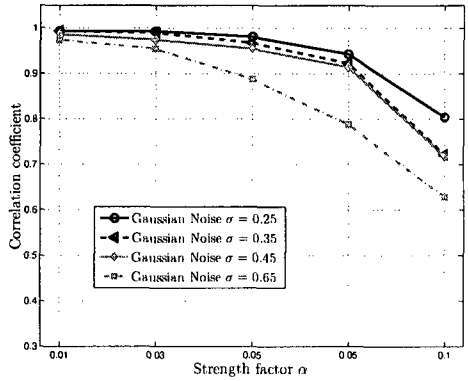


Compression attack

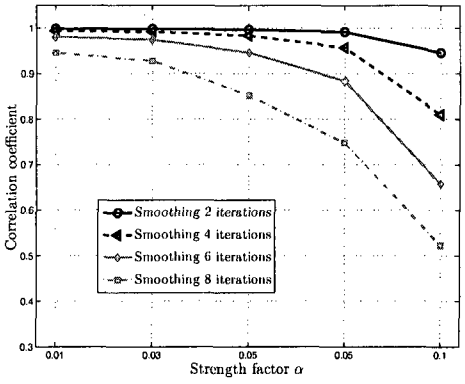
Figure 60: Correlation coefficient results for the cow model using five different strength factors and noise, smoothing, and compression attacks



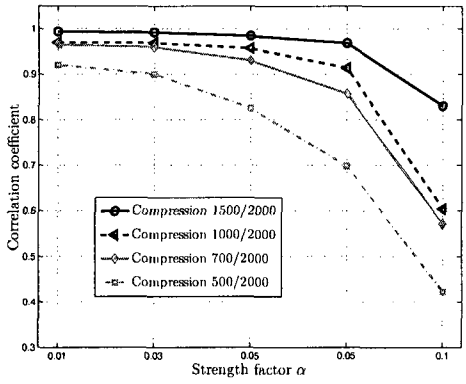
Camel model



Gaussian noise attack

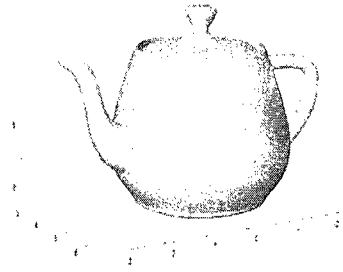


Smoothing attack

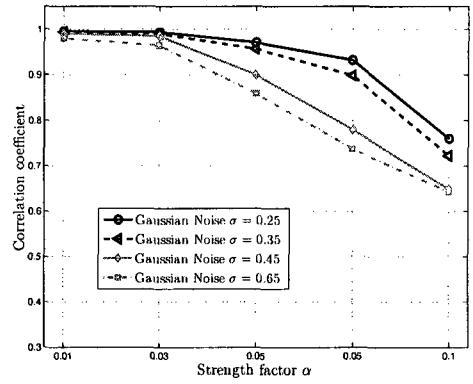


Compression attack

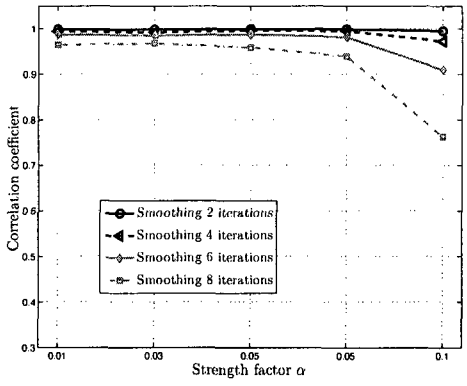
Figure 61: Correlation coefficient results for the camel model using five different strength factors and noise, smoothing, and compression attacks



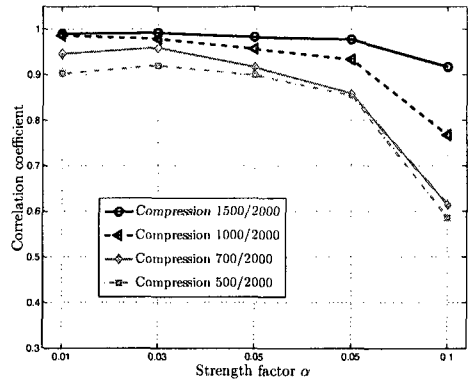
Teapot model



Gaussian noise attack



Smoothing attack



Compression attack

Figure 62: Correlation coefficient results for the teapot model using five different strength factors and noise, smoothing, and compression attacks

Chapter 6

Video Watermarking using DWT and TSVD

In this chapter, we present two robust, hybrid MPEG video watermarking algorithms to embed an invisible watermark into the intra-frames of an MPEG video sequence. Unlike previous methods where each video frame is marked separately, our proposed approaches use high-order tensor decomposition of videos. The core idea behind our proposed techniques is to use the scene-change analysis to embed the watermark repeatedly into the singular values of high-order tensors, which have a good stability and represent the video properties. These singular values are computed form the DWT coefficients of selected frames of each scene. The main attractive features of these approaches are simplicity and robustness.

6.1 Video Watermarking Overview

A variety of watermarking techniques have been proposed to embed a robust watermark into digital images. Image watermarking techniques can be extended easily to watermark video image sequences [35, 47]. However, video watermarking schemes need to meet some other challenges. Applying a fixed image watermark to each frame

in the video leads to problems of maintaining statistical and perceptual invisibility. Furthermore, applying independent watermarks to each frame also yields a frame averaging problem which is usually used by the attackers to remove the watermark [16, 54].

The early video watermarking techniques add a visible signature or a logo to the video frames [21]. These watermarks do not usually cover significant areas of the video frames, making them easy to remove by a cropping attack. Recently, a real-time digital video watermarking scheme [55] has been proposed to embed the watermark in intra-pictures of an MPEG video sequence by modifying the variable length codes (VLCs) directly in order to avoid inverse quantization. The main advantage of modifying the VLCs is that the perceptual degradation of video quality caused by the embedded watermark is minimized [55]. In [87], a blind MPEG2 video watermarking technique was proposed by focusing on geometric attacks. The discrete Fourier transform of 3D chunks of a video scene was used in [22] for video watermarking, where the embedding and the extraction algorithms are applied to uncompressed video data. In [37], only the DCT coefficients of the intra-frames in the MPEG compressed video are watermarked, and the spread spectrum signal was used as a copyright information that was added to the non-zero DCT coefficients under the condition of not increasing the bit rate. Embedding the watermark in the uncompressed domain was proposed in [54], where the watermark was embedded in the intra-frames by adopting the block matching algorithm to find the motion vector of each block and also by using the motion feature to embed the watermark.

We present in this chapter a scene-change watermarking approach using a hybrid scheme based on DWT and tensor singular value decomposition (TSVD). Our approach generalizes the method proposed in [27] by embedding the watermark data in all the frequencies of the video scenes. The experimental results show that the proposed schemes are robust against a variety of attacks including frame dropping,

frame averaging, frame swapping, geometric transformations, adaptive random noise, low-pass filtering, and histogram equalization.

The rest of the chapter is organized as follows. In Section 6.2, we briefly review the multidimensional tensor singular value decomposition, In Section 6.3, we provide a brief review of some previous works that are closely related to our proposed watermarking schemes. In Section 6.4, we introduce the proposed methodologies, and describe in detail the watermark embedding and extraction algorithms. Experimental results are presented in Section 6.5 to demonstrate the performance of the proposed watermarking scheme in comparison with existing methods.

6.2 Tensor Algebra

In this section we review the multilinear generalization of the singular value decomposition. We briefly discuss the mathematical notation, foundations and the algorithmic methodology.

6.2.1 Multidimensional tensor singular value decomposition

Higher-order singular value decomposition (HOSVD) has been proposed in [48] to analyze multilinear structures. Transforming a 3D tensor into a matrix is usually referred to as a "matricization" process [10, 44, 71]. The n-mode matricizing of a tensor $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2 \ldots I_N}$ is denoted by a matrix $\mathbf{A}_n \in \mathbb{R}^{I_n \times (I_{n+1} \times \ldots \times I_N \times I_1 \times \ldots \times I_{n-1})}$, as is shown in Figure 63.

The n-mode product of a tensor $\mathbf{A}$ by a matrix $\mathbf{U} \in \mathbb{R}^{J_n \times I_n}$ is an $I_1 \times \ldots \times I_{n-1} \times J_n \times I_{n+1} \times \ldots \times I_N$ tensor denoted by $\mathbf{A} \times_n \mathbf{U}$, whose entries are defined by:

$$[\mathbf{A} \times_n \mathbf{U}]_{i_1 i_2 \ldots i_{n-1} j_n i_{n+1} \ldots i_N} = \sum_{i_n} a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N} u_{j_n i_n} \tag{44}$$

where $a_{i_1 i_2 \ldots i_{n-1} i_n i_{n+1} \ldots i_N}$ is an entry of $\mathbf{A}$, and $u_{j_n i_n}$ is an entry of $\mathbf{U}$.

The n-mode product $\times_n$ satisfies commutability [48, 71]. Given a tensor $\mathbf{A} \in \mathbb{R}^{I_1 \times I_2 \times \ldots \times I_m \times \ldots \times I_n \times \ldots \times I_N}$ and two matrices $\mathbf{B} \in \mathbb{R}^{J_n \times I_n}$ and $\mathbf{C} \in \mathbb{R}^{J_m \times I_m}$, we have

$$\mathbf{A} \times_n \mathbf{B} \times_m \mathbf{C} = \mathbf{A} \times_m \mathbf{C} \times_n \mathbf{B} \tag{45}$$

In this approach, we deal mainly with video sequences that are represented as a 3D tensor with two dimensions in space and one dimension in time. Let $\boldsymbol{A}$ be 3D video tensor of size $m \times n \times p$. The tensor $\boldsymbol{A}$ can be rearranged into a matrix of size $k \times \ell$ in three different ways: left-right matrix $\boldsymbol{A_1}$, front-back matrix $\boldsymbol{A_2}$, and top-bottom matrix $\boldsymbol{A_3}$, as shown in Figure 63. Clearly, the number of elements in the matrices $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ and $\boldsymbol{A_3}$ must be the same as the number of elements in the tensor $\boldsymbol{A}$.

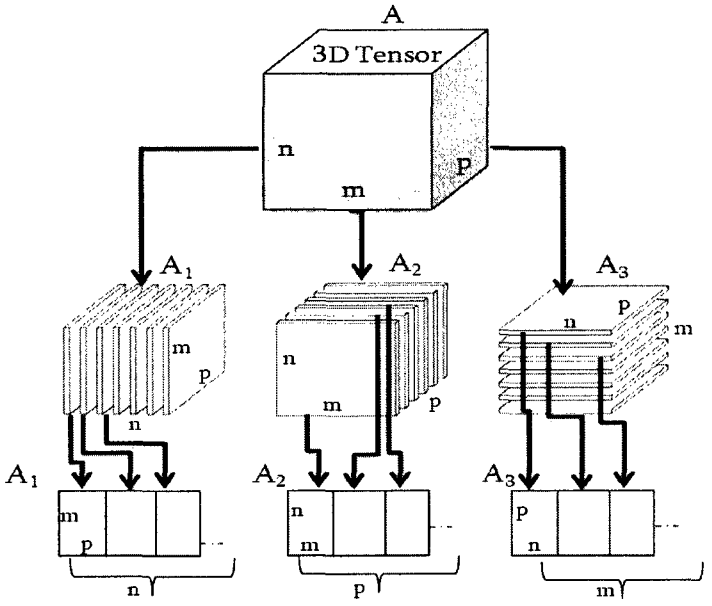


Figure 63: Illustration of matricizing a third-order tensor $\mathbf{A}$ into a matrix in three ways. $\boldsymbol{A_1} \in \mathbb{R}^{n \times (m \times p)}$ is the one-mode matricizing of the tensor $\mathbf{A}$. $\boldsymbol{A_2} \in \mathbb{R}^{p \times (n \times m)}$ is the two-mode matricizing of the tensor $\mathbf{A}$. $\boldsymbol{A_3} \in \mathbb{R}^{m \times (p \times n)}$ is the three-mode matricizing of the tensor $\mathbf{A}$

Extending matrix decompositions such as the SVD to higher-order tensors has proven to be quite difficult [10]. Given an $m \times n \times p$ tensor $\boldsymbol{A}$, the Tucker decomposition

(see Figure 64) is given by:

$$\boldsymbol{A} = \boldsymbol{\Sigma} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{W} = \sum_{i=1}^{r_1} \sum_{j=1}^{r_2} \sum_{k=1}^{r_3} \sigma_{ijk} (u_i \otimes v_j \otimes w_k) \tag{46}$$

where $r_1 \leq m$, $r_2 \leq n$, $r_3 \leq p$ and the columns of $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ are the left singular vectors of the matrices $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ and $\boldsymbol{A_3}$. The tensor $\boldsymbol{\Sigma} = (\sigma_{ijk})$, is called the core tensor and it is given by:

$$\boldsymbol{\Sigma} = \boldsymbol{A} \times_1 \boldsymbol{U}^T \times_2 \boldsymbol{V}^T \times_3 \boldsymbol{W}^T \tag{47}$$

The core tensor does not necessarily have the same dimension as $\boldsymbol{A}$. In general, we can have either orthogonal columns of $\boldsymbol{U}$, $\boldsymbol{V}$, and $\boldsymbol{W}$ or a diagonal core tensor $\boldsymbol{\Sigma}$ [48].

Applying SVD to the matrices $\boldsymbol{A_1}, \boldsymbol{A_2}$ and $\boldsymbol{A_3}$ yields:

$$\begin{aligned} \boldsymbol{A}_1 &= \boldsymbol{U}\boldsymbol{D}_1\boldsymbol{G}_1^T \\ \boldsymbol{A}_2 &= \boldsymbol{V}\boldsymbol{D}_2\boldsymbol{G}_2^T \\ \boldsymbol{A}_3 &= \boldsymbol{W}\boldsymbol{D}_3\boldsymbol{G}_3^T \end{aligned} \tag{48}$$

where the columns of $\boldsymbol{G}_1$, $\boldsymbol{G}_2$, and $\boldsymbol{G}_3$ are the right singular vectors of $\boldsymbol{A_1}$, $\boldsymbol{A_2}$ and $\boldsymbol{A_3}$ respectively. Moreover, we have

$$\begin{aligned} \boldsymbol{A}_1 &= \boldsymbol{U}\boldsymbol{\Sigma}_1(\boldsymbol{V} \otimes \boldsymbol{W})^T \\ \boldsymbol{A}_2 &= \boldsymbol{V}\boldsymbol{\Sigma}_2(\boldsymbol{W} \otimes \boldsymbol{U})^T \\ \boldsymbol{A}_3 &= \boldsymbol{W}\boldsymbol{\Sigma}_3(\boldsymbol{U} \otimes \boldsymbol{V})^T \end{aligned} \tag{49}$$

where $\boldsymbol{\Sigma}_1 = \boldsymbol{D}_1\boldsymbol{G}_1^T(\boldsymbol{V} \otimes \boldsymbol{W}), \boldsymbol{\Sigma}_2 = \boldsymbol{D}_2\boldsymbol{G}_2^T(\boldsymbol{W} \otimes \boldsymbol{U})$ and $\boldsymbol{\Sigma}_3 = \boldsymbol{D}_3\boldsymbol{G}_3^T(\boldsymbol{U} \otimes \boldsymbol{V})$

6.3 Video Watermarking Related Works

In this section, we review three representative methods for digital video watermarking that are closely related to our proposed approach. We briefly discuss their mathematical foundations and algorithmic methodologies.

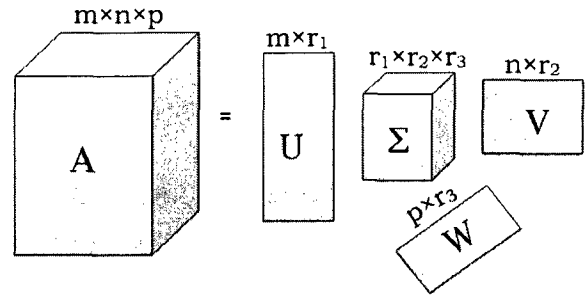


Figure 64: Tucker decomposition of a 3D tensor

6.3.1 MPEG video watermarking in the DWT domain

The idea of the method introduced in [25] is to embed a binary pattern in the form of a binary image as an invisible watermark in the four wavelet sub-bands of each intra-frame of the MPEG video. This could be done as follows. The cover video frame $\boldsymbol{C}$ is decomposed into four sub-bands: the approximation coefficient LL, and the detailed coefficients HL, LH, and HH. The DWT coefficients of each sub-band $C^k \in$ {LL,HL,LH,HH} are modified with the binary image as follows

$$\widehat{C}_{ij}^k = C_{ij}^k + \alpha_k w_{ij}, \quad 1 \leq k \leq 4 \tag{50}$$

where C_{ij}^k and $\widehat{C}_{ij}^k$ denote the original and the distorted DWT coefficients of the sub-band K respectively, and w_{ij} denotes the $(i,j)^{th}$ pixel value of the watermark image. Hence, we get the four modified sub-bands. Then, the inverse DWT is applied using the four sets of the modified DWT coefficients to produce the watermarked frame.

The algorithm is invertible and the watermark can be extracted from the watermarked video frames by extracting the binary values of the visual watermark using: $w_{ij}^{*k} = (\widehat{C}_{ij}^k - C_{ij}^k)/\alpha_k$.

6.3.2 Blind hybrid scene-based watermarking scheme

In [16], the watermark is divided into small parts as a preprocess before embedding these parts into scenes. Four levels of DWT are applied to all frames in a video

sequence, producing a low-frequency sub-band LL4, and three series of high-frequency sub-bands. Different watermarks are embedded in frames of different scenes and an identical watermark is used for each frame in the same scene. The watermark embedding is applied to the video frame by changing the position of some DWT coefficients. Let W_j be the j^{th} pixel value of a corresponding watermark image, and C_i the i^{th} DWT coefficient of the video frame then,

If $W_j = 1$, exchange C_i with the maximum of $(C_i, C_{i+1}, C_{i+2}, C_{i+3}, C_{i+4})$

Else exchange C_i with the minimum of $(C_i, C_{i+1}, C_{i+2}, C_{i+3}, C_{i+4})$,

This algorithm is blind, that is, the retrieval of the embedded watermark does not need the original video frames. To extract the watermark, each video frame is transformed to the wavelet domain with four levels. Then the watermark is extracted using the following condition:

If $WC_i >$ median $(WC_i, WC_{i+1}, WC_{i+2}, WC_{i+3}, WC_{i+4})$ then $EW_j = 1$

Else $WC_i <$ median $(WC_i, WC_{i+1}, WC_{i+2}, WC_{i+3}, WC_{i+4})$ then $EW_j = 0$

where WC_i is the i^{th} DWT coefficient of the watermarked video frame, and EW_j is the j^{th} pixel value of a extracted watermark image.

To improve the robustness against image processing attacks on a video frame, a hybrid approach is applied by using different watermarking schemes for different scenes. The watermark is still decomposed into different parts which are embedded in the corresponding frames of different scenes in the original video. Each part of the watermark, however, is embedded with a different watermarking scheme. Within a scene, all the video frames are watermarked with the same part of a watermark by the same watermarking scheme. Thus, the hybrid approach enhances the robustness against image processing attacks.

6.4 Proposed Methodologies

6.4.1 Video watermarking using tensor singular value decomposition

Watermark embedding process

In an MPEG system, there are three kinds of coded images in each group of pictures: I (intra) frames, P (predicted) frames, and B (bidirectional) frames. The P-frames are forward predicted from the last I-frame or P-frame and it may not be possible to reconstruct them without the data from another I or P frames. The B-frames are forward predicted and backward predicted from the last/next I-frame or P-frame. In order to improve the robustness of the proposed scheme [6] we only use the I-frames to embed the watermark. Partitioning MPEG video into scenes is done by counting the percentage of each type of blocks in a frame. For example, a large percentage of the intra-blocks in the P-frame implies the beginning of a new scene [87]. The watermark embedding process description is shown in Algorithm 6.1. We used an identical watermark for each group of the I-frames in the same scene, and different watermarks for different scenes.

Watermark extraction process

In order to extract the watermark, our algorithm requires the original video sequence as well as the watermarked video sequence. The watermark extraction process description is shown in Algorithm 6.2. Figure 67 shows an example of video watermarking using our proposed scheme. Clearly the difference between the original and the watermarked videos is not noticeable to the human observer.

Algorithm 6.1 Watermark embedding algorithm

Input *MPEG video sequence* $\boldsymbol{V}$, *and a watermark image* $\boldsymbol{W}$ *of size* $m \times m$.
Output: *Watermarked video* $\widehat{\boldsymbol{V}}$.
For each scene:
1- Convert the I-frames from RGB to YUV.
2- Divide the converted Luminance layers (Y) of the I-frames into chunks (Groups) of fixed length (10 frames). Each Group of the I-frames is represented by 3D tensor as shown in Figure 65.
3- Matricize the tensor in three different ways (Left-right, Front-back, and Top-bottom) to obtain $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, $\boldsymbol{A_3}$ respectively.
4- Apply SVD to the matrices $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, and $\boldsymbol{A_3}$ that is, $\boldsymbol{A_1} = \boldsymbol{U}\boldsymbol{D}_1\boldsymbol{G}_1^T$, $\boldsymbol{A_2} = \boldsymbol{V}\boldsymbol{D}_2\boldsymbol{G}_2^T$, and $\boldsymbol{A}_3 = \boldsymbol{W}\boldsymbol{D}_3\boldsymbol{G}_3^T$.
5- Calculate the 3D singular values matrix $\boldsymbol{\Sigma_{3D}} = \boldsymbol{A} \times_1 \boldsymbol{U}^T \times_2 \boldsymbol{V}^T \times_3 \boldsymbol{W}^T$.
6- Apply SVD to $\boldsymbol{W}$, that is: $\boldsymbol{W} = \boldsymbol{U}_w\boldsymbol{\Sigma}_w\boldsymbol{V}_w^T$
7- Modify the largest SVs of $\boldsymbol{\Sigma_{3D}}$ with the SVs of $\boldsymbol{W}$ using: $\widehat{\lambda^i} = \lambda^i + \alpha\lambda_w^i$, where λ^i and λ_w^i, $1 \leq i \leq m$ are the singular values of $\boldsymbol{\Sigma_{3D}}$ and $\boldsymbol{\Sigma}_w$ respectively, $\widehat{\lambda^i}$ denotes the distorted SVs, and α is a constant scaling factor.
8- Produce the watermarked tensor $\boldsymbol{A}_w = \widehat{\boldsymbol{\Sigma}}_{3D} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{W}$, where $\widehat{\boldsymbol{\Sigma}}_{3D}$ is the modified 3D core tensor.
9- Finally use the modified I-frames to produce the watermarked cover video sequence as depicted in Figure 66.

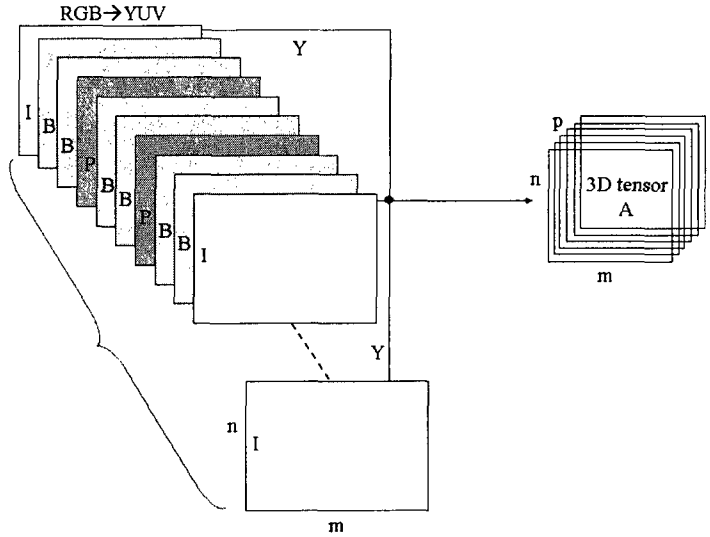


Figure 65: Illustration of a multidimensional tensor produced from one group of the I-frames

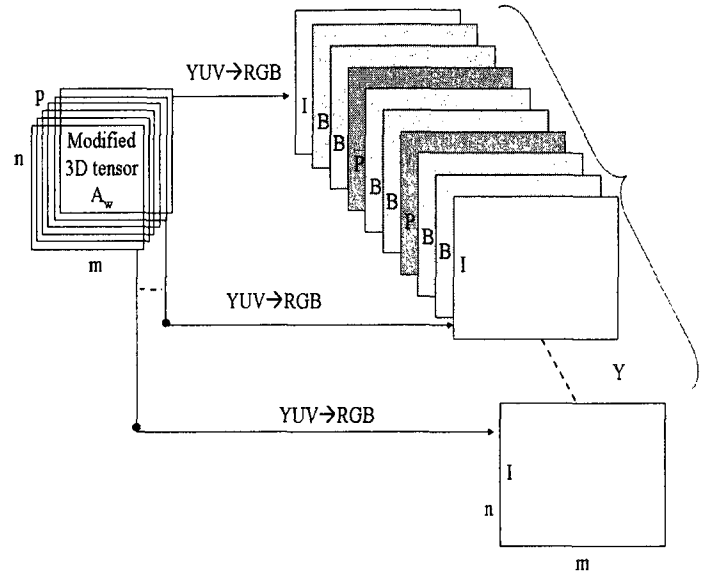


Figure 66: Producing the watermarked cover video sequence diagram

Algorithm 6.2 Watermark extraction algorithm

Input *Original and watermarked MPEG video sequences.*
Output: *Extracted watermark image* $\widehat{\boldsymbol{W}}$.
For each scene:
1- Apply the first five steps of the watermark embedding process to the original and the watermarked video sequences.
2- Extract the singular values of the visual watermark as follows: $\widehat{\lambda}_w^i = (\widehat{\lambda}^i - \lambda^i)/\alpha$, where λ^i and $\widehat{\lambda}^i$ are the original and the watermarked singular values respectively.
3- Construct the watermark image using the extracted singular values $\widehat{\boldsymbol{W}} = \boldsymbol{U}_w \widehat{\boldsymbol{\Sigma}}_w \boldsymbol{V}_w^T$, where $\boldsymbol{U}_w$ and $\boldsymbol{V}_w$ are the left and right singular vectors of $\boldsymbol{W}$ respectively, and $\widehat{\boldsymbol{\Sigma}}_w$ =diag$(\widehat{\lambda}_w^i)$ is the extracted matrix of SVs.

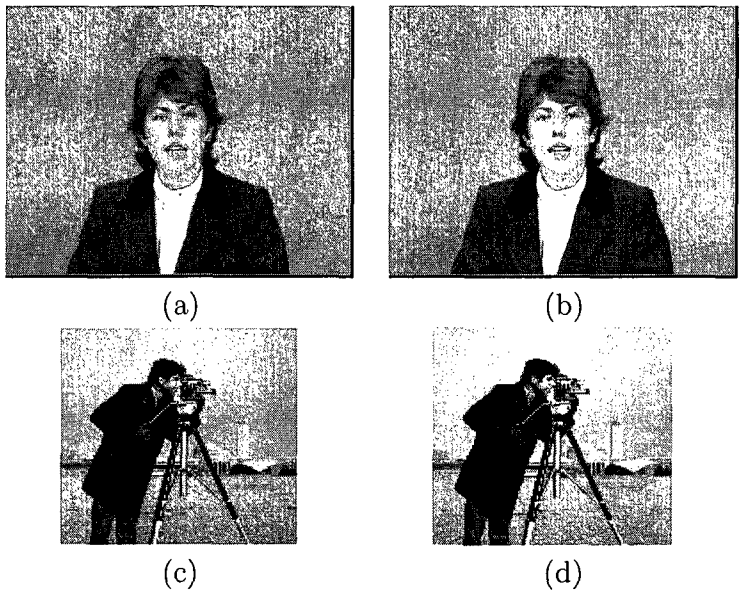


Figure 67: (a) Original video frame, (b) Watermarked video frame, (c) Original watermark image, and (d) Extracted watermark image from the 3D tensor containing the video frame shown in (a)

6.4.2 Video watermarking using wavelet transform and tensor singular value decomposition

In [27], the authors showed experimentally that embedding the watermark in the low and high-frequency components of an image increases the robustness against attacks. Therefore, the goal of our proposed approach [5] is to apply multiple transforms to selected frames of the cover video in order to embed the watermark many times in all the frequencies that provide better robustness against attacks. This would amplify the difficulty of destroying the watermark from all the frequencies, and provide a high visual quality of the watermarked video sequence.

In order to achieve a low complexity and improve the robustness, we only used the I-frames to embed the watermark. The watermark Ω used for embedding is a sequence of random numbers that are produced from an integer random number generator. An identical watermark has been used for each group of the I-frames in the same scene, and different watermarks for different scenes.

Watermark embedding

The watermark embedding process description is summarized as follows:
For each scene:

1) Convert the I-frames from RGB to YUV (Y represents the luminance component i.e. the brightness, U and V represent the chrominance components i.e color). In order to make the watermark imperceptible, we use the luminance layer to embed the watermark and we leave the chrominance layer unchanged.

2) Apply DWT to the converted luminance layers (Y) of the I-frames to obtain 4 sub-bands of each frame (LL, LH, HL, HH).

3) For each set of I-frames, divide the sub-bands into four chunks (groups). The first group is created from LL sub-bands, the second one from LH sub-bands, the third one from HL sub-bands, and the fourth one from HH sub-bands. All these groups are represented as 3D tensors as shown in Figure 68.

4) Matricize the four tensors in three different ways (left-right, front-back, and top-bottom) to obtain $\boldsymbol{A}_1^k$, $\boldsymbol{A}_2^k$, $\boldsymbol{A}_3^k$ respectively, where the index $k \in \{1, 2, 3, 4\}$ represents the four tensors.

5) For each tensor, apply SVD to the matrices $\boldsymbol{A_1}$, $\boldsymbol{A_2}$, and $\boldsymbol{A_3}$ that is, $\boldsymbol{A}_1 = \boldsymbol{U}\boldsymbol{D}_1\boldsymbol{G}_1^T$, $\boldsymbol{A}_2 = \boldsymbol{V}\boldsymbol{D}_2\boldsymbol{G}_2^T$, and $\boldsymbol{A}_3 = \boldsymbol{W}\boldsymbol{D}_3\boldsymbol{G}_3^T$.

6) Calculate the 3D singular values matrix $\boldsymbol{\Sigma_{3D}} = \boldsymbol{A} \times_1 \boldsymbol{U^T} \times_2 \boldsymbol{V^T} \times_3 \boldsymbol{W^T}$.

7) For all the tensors, modify the largest SVs of $\boldsymbol{\Sigma_{3D}}$ with a random watermark vector $\boldsymbol{\Omega_m}$ using: $\widehat{\lambda^i} = \lambda^i + \alpha w^i$, where $\widehat{\lambda^i}$, λ^i are the distorted and the original SVs of $\boldsymbol{\Sigma_{3D}}$ respectively, α is a constant scaling factor, and $1 \leq i \leq m$ where m is the size of the watermark vector.

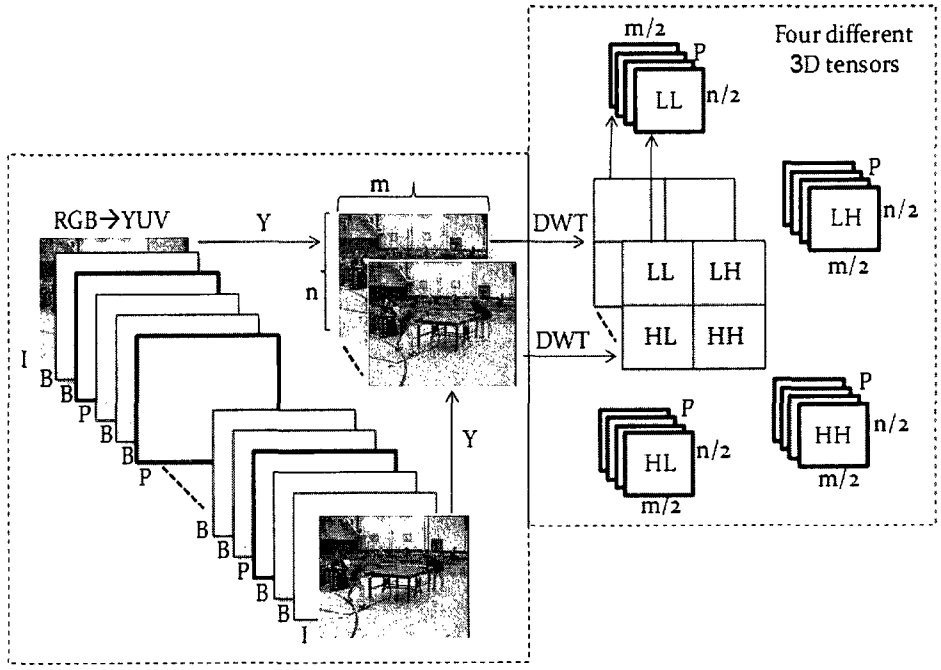


Figure 68: Illustration of a multidimensional four tensors produced from one group of the I-frames

8) Produce the watermarked tensor $\boldsymbol{A}_w = \widehat{\boldsymbol{\Sigma}}_{3D} \times_1 \boldsymbol{U} \times_2 \boldsymbol{V} \times_3 \boldsymbol{W}$, where $\widehat{\boldsymbol{\Sigma}}_{3D}$ is the modified 3D core tensor.

9) Use $\boldsymbol{A}_w$ to produce the modified luminance layer of the I-frames.

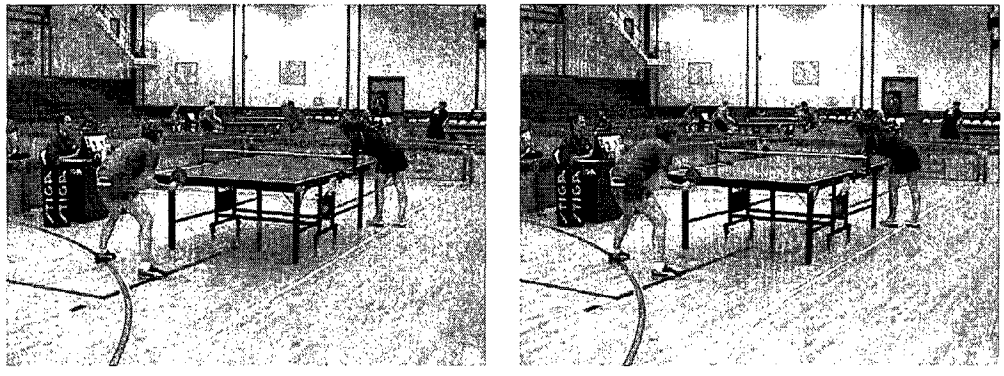
10) Finally use the modified I-frames to produce the watermarked cover video sequence.

Watermark extraction

The watermark extraction is performed by applying the first six steps of the watermark embedding process to the original as well as the watermarked video sequences. Then, for each set of the I-frames we extract the watermark vector four times from the four tensors representing the transformed wavelet coefficients using: $\widehat{w}^i = (\widehat{\lambda}^i - \lambda^i)/\alpha$, where λ^i and $\widehat{\lambda}^i$ are the original and the watermarked singular values respectively. Finally we select the extracted watermark vector that has the highest correlation with

the original watermark. Figure 69 shows an example of video watermarking using our proposed scheme. Clearly the difference between the original and the watermarked videos is unnoticeable to the human observer.

The proposed method is computationally inexpensive. This is because the watermark embedding and extraction algorithms are not applied to each and every single frame of the video image sequence. However, the watermark algorithms is applied to 3D tensors computed from the I-frames of the cover video.



(a) (b)

Figure 69: (a) Original table tennis video frame, (b) Watermarked table tennis video frame

6.5 Video Watermarking Experimental Results

We tested the performance of the proposed watermarking scheme on several video sequences that are shown in Figure 70.

6.5.1 Robustness of the pure TSVD method

The experimental tests are mainly performed to verify the robustness against attacks including rescaling, rotation, Gaussian noise, histogram equalization, gamma correction, low-pass filtering, sharpening, motion blurring, frame compression, frame dropping, frame swapping, frame averaging, cropping, and also combinations of these

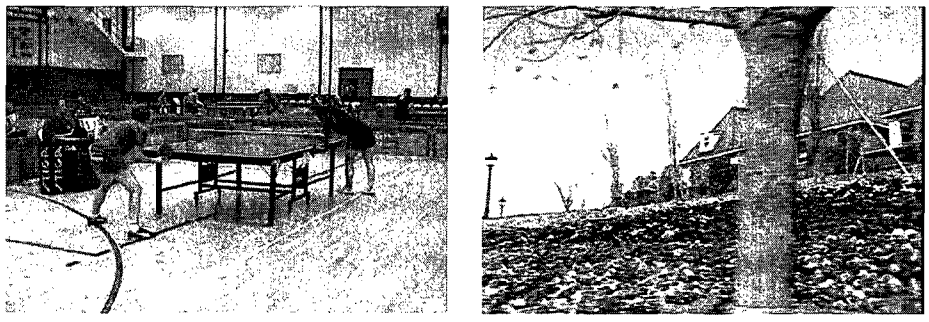
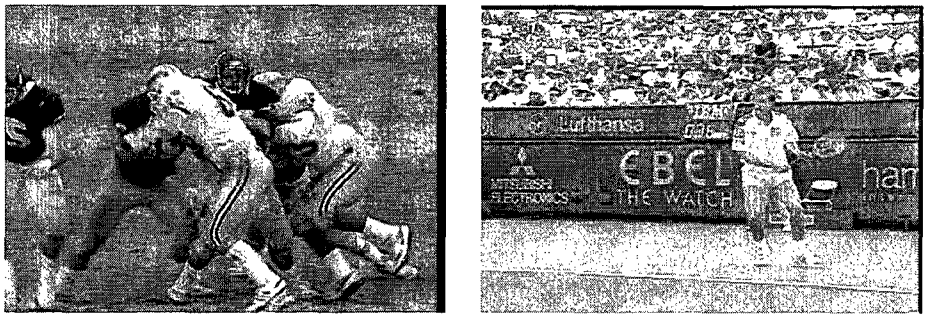


Table tennis Flower garden



Football Tennis

Figure 70: Sample frames from videos used in the experiments

attacks. For all the attacks, we show one of the attacked I-frames and the extracted watermark. Figures 71-74 show one of the watermarked I-frames with different kinds of attacks. The corresponding extracted watermarks also shown in Figures 71-74. The title below each extracted watermark displays the correlation coefficient between the original and the best extracted watermark. Figure 75 depicts the correlation coefficients for different video sequences under the scaling attack with ratio from 0.1 to 0.9. The results obtained from our experiments clearly indicate the robustness of the proposed algorithm against the commonly used attacks in videos.

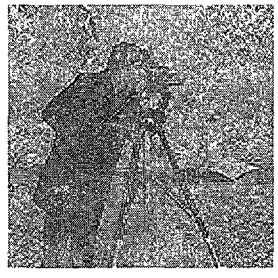
6.5.2 Imperceptibility of the proposed DWT-TSVD method

A watermark vector of 128 random numbers has been used in these experiments. We used the same watermark for each group of the I-frames in the same scene, and different watermarks for different scenes. The experiments are performed to verify the

Rescaling $100\% - 50\% - 100\%$

$\rho = 0.9852$

Salt & peppers noise (2%)



$\rho = 0.7537$

Histogram equalization

$\rho = 0.9128$

Figure 71: First group of the watermarked Claire video I-frames distorted by different attacks and their extracted watermarks from the tensor containing the attacked I-frames

Gamma correction

$\rho = 0.901$

Low-pass filter (3×3)

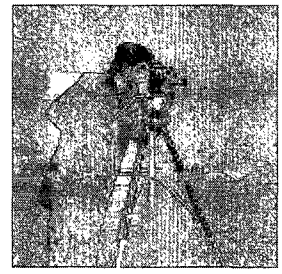
$\rho = 0.9071$

Sharpening $\alpha = 0.2$

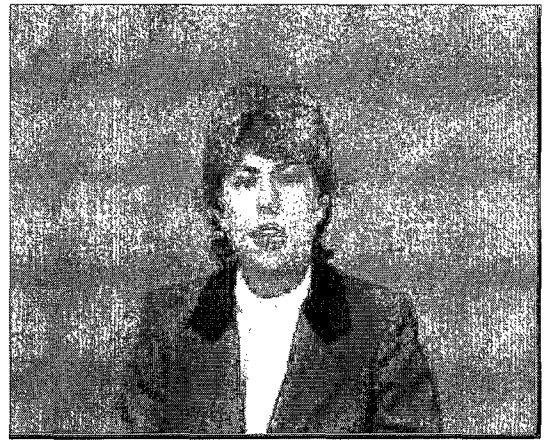
$\rho = 0.7267$

Figure 72: Second group of the watermarked Claire video I-frames distorted by different attacks and their extracted watermarks from the tensor containing the attacked I-frames

Rotation 20^o



$\rho = 0.7141$



JPEG compression $Q = 25$ (3×3)

$\rho = 0.9412$

Cropping 8% from the top and 8% from the bottom

$\rho = 0.9783$

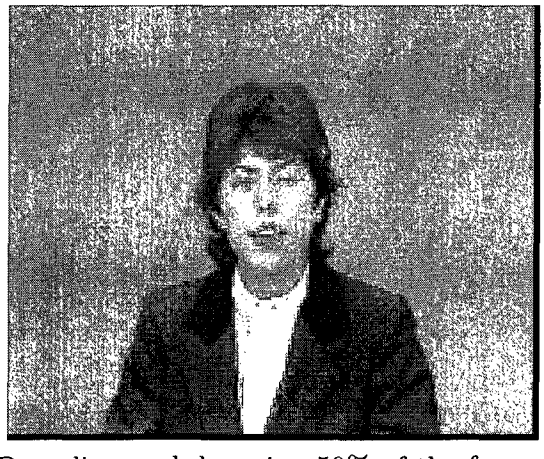
Figure 73: Third group of the watermarked Claire video I-frames distorted by different attacks and their extracted watermarks from the tensor containing the attacked I-frames

Cropping 8% from the left and 8% from the right

$\rho = 0.9905$

Cropping 5% from top and 5% from the bottom and drooping 50% of the frames

$\rho = 0.9591$



Rescaling and drooping 50% of the frames

$\rho = 0.9759$

Figure 74: Fourth group of the watermarked Claire video I-frames distorted by different attacks and their extracted watermarks from the tensor containing the attacked I-frames

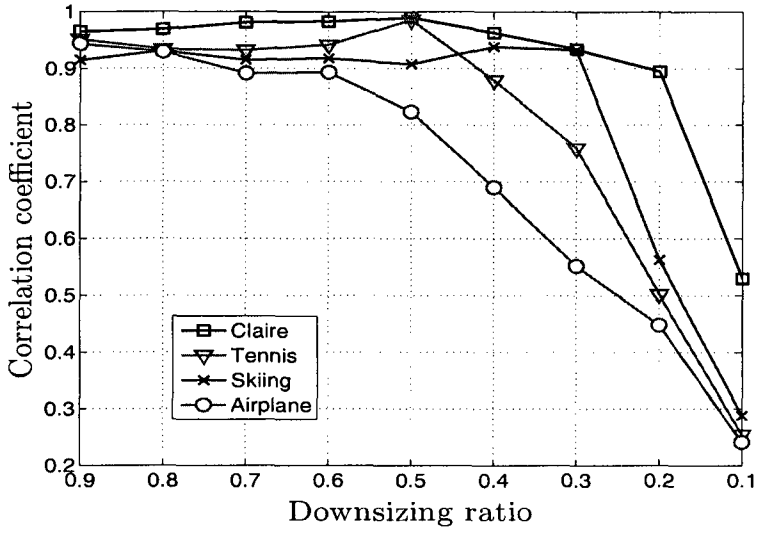
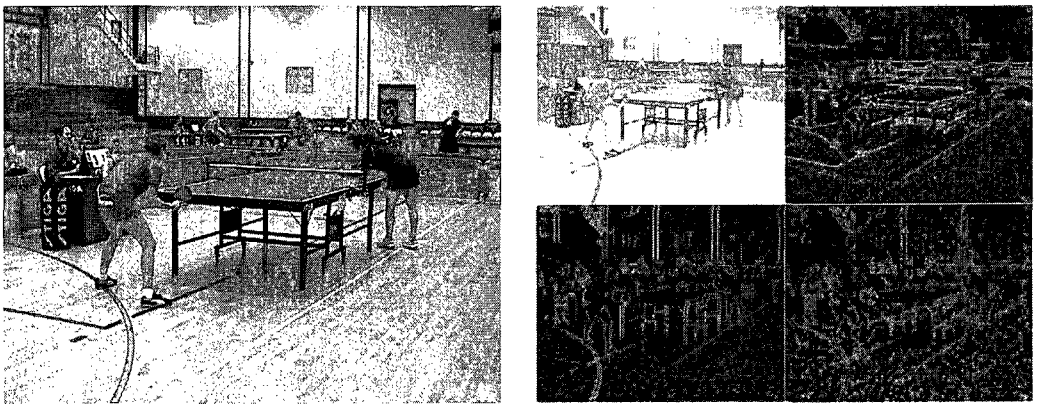


Figure 75: Robustness of the proposed pure tensor watermarking scheme against rescaling attack for different video sequences: Claire, Tennis, Skiing, and Airplane

watermark imperceptibility and robustness against attacks. In order to achieve a high visual quality of the watermarked video sequence, the watermark strength factor α should be taken into consideration. Experimentally a constant scaling factor $\alpha = 0.1$ was used for the tensors representing the LL sub-bands and $\alpha = 0.05$ for all the other tensors. The strength factors are chosen according to the wavelet coefficients of the sub-bands frames. The LL sub-bands have the lowest frequency components of the cover video frames and the highest wavelet coefficients (highest magnitude). The (HL, LH and HH) sub-bands have very similar wavelet coefficients values, and therefore we used the same strength factor for all middle and high-frequency sub-bands.

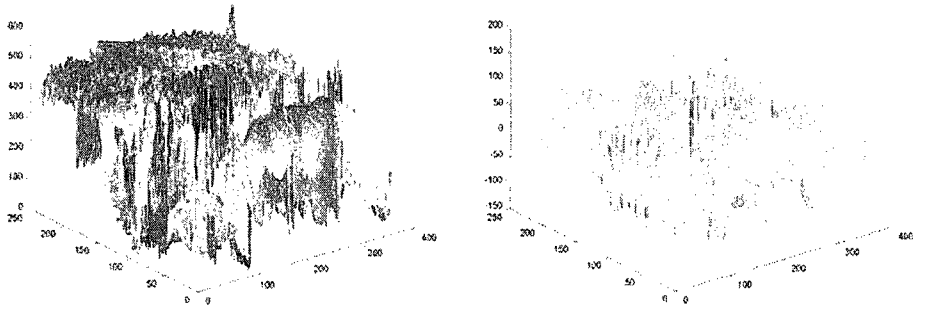
Figure 76 (a)-(b) show an example of one frame of the tennis video sequence with one level discrete wavelet transform. Figure 77 shows the wavelet coefficients of all the four sub-bands shown in Figure 76 (b), where it can be seen that the wavelet coefficients of the LL sub-band are the highest among all the coefficients of the other sub-bands.

In general, the accurate measurement of the perceptual quality as perceived by a human observer is a great challenge in image/video processing. The reason is that the

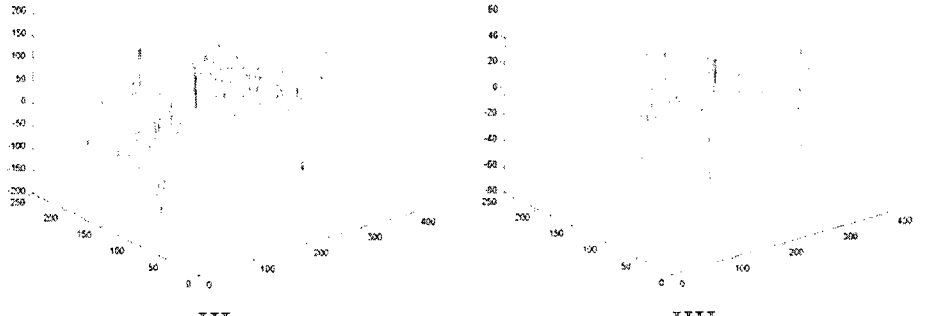


(a) (b)

Figure 76: One frame of the Tennis video sequence with one level of DWT decomposition



LL LH



HL HH

Figure 77: DWT coefficients of all the four sub-bands shown in Figure 6.14 (b)

amount and visibility of distortions introduced by the watermarking attacks strongly depend on the actual image/video content [88]. To measure the perceptual quality, we calculate the PSNR [64] that is used to estimate the quality of the watermarked frames in comparison with the original ones.

The PSNR experimental results are shown in Figure 78, and indicate that the proposed method provides a high visual quality of the reconstructed video sequences, and hence guarantees the watermark imperceptibility. Table 9 shows the effect of the watermark strength factor α. Note that a smaller value of α increases the watermark imperceptibility, however it decreases the robustness against attacks.

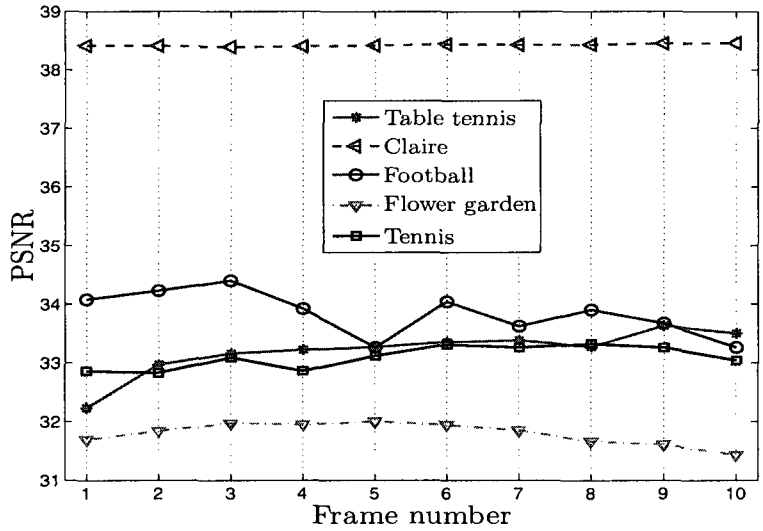


Figure 78: PSNR results for different video sequences. Ten continuous frames are chosen from one video scene

6.5.3 Robustness of the proposed DWT-TSVD method

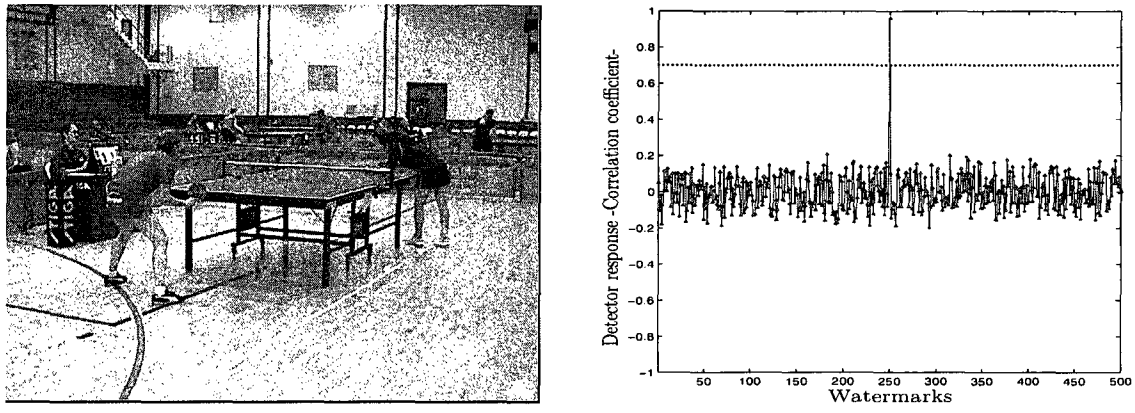
To assess the robustness of our proposed method, we applied different attacks to the watermarked video sequence. For each attack, we display one of the attacked I-frames and the best detector response for the real watermark, as well as 499 randomly generated other watermarks. For all the detector responses, the correlation coefficient

Table 9: PSNR of the Claire video sequence. PSNR between 10 frames and their corresponding watermarked frames with different strength factors. The left-hand side α value is used for the LL band and the right-hand side α value is used for the LH, HL and HH sub-bands.

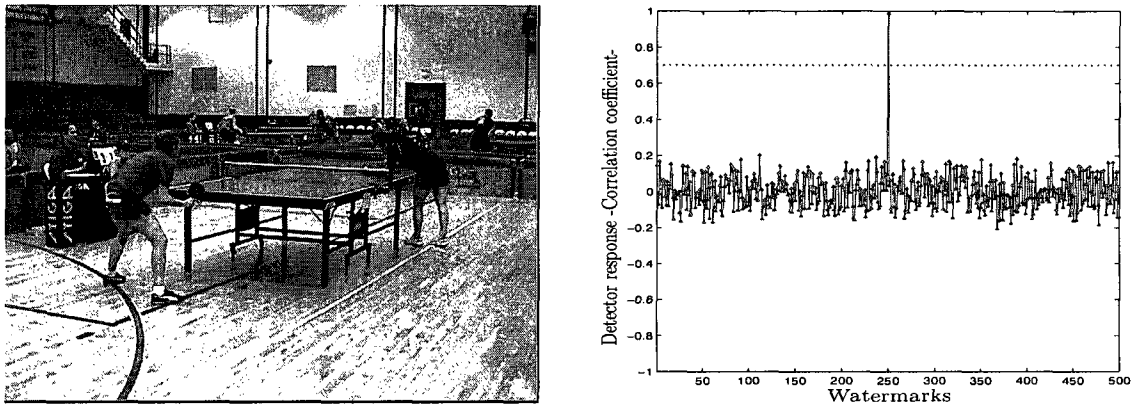
	$\alpha = 0.25\|0.15$	$\alpha = 0.2\|0.1$	$\alpha = 0.1\|0.05$	$\alpha = 0.05\|0.01$
Frame 1	31.49	33.74	38.41	42.53
Frame 2	31.50	33.74	38.41	52.52
Frame 3	31.47	33.72	38.39	42.51
Frame 4	31.48	33.73	38.43	42.52
Frame 5	31.51	33.75	38.42	42.53
Frame 6	31.53	33.78	38.44	42.53
Frame 7	31.52	33.77	38.43	42.54
Frame 8	31.52	33.77	38.43	42.54
Frame 9	31.54	33.79	38.45	42.54
Frame 10	31.57	33.81	38.46	42.54

between the original watermark and the extracted watermark is located at 250 on the X-axis. The gray dotted line at 0.7 on the Y-axis represents the threshold. Figures 79 and 80 show one of the watermarked frames with different kinds of attacks and their corresponding best extracted watermarks. For each attack, we extracted four watermarks from the four tensors, and then we selected the best watermark that has the highest correlation coefficient with the original watermark. The caption of each sub-figure of Figure 79 and Figure 80 display the correlation coefficient between the original and the four extracted watermarks. The boldface numbers indicate the best correlation. Table 2 displays the results for some other attacks. Figure 81 illustrates the robustness of our proposed method for different video sequences. The results obtained from our experiments clearly indicate the robustness of the proposed algorithm against the commonly used attacks in videos.

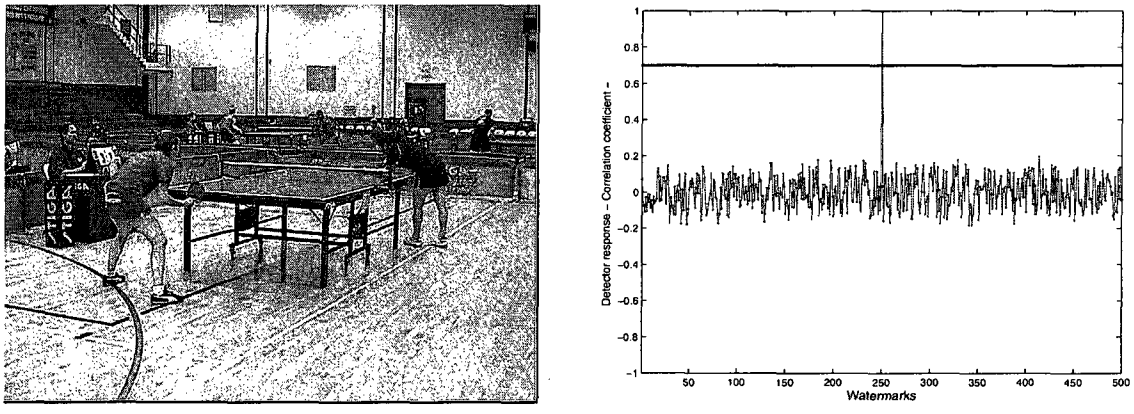
Frame dropping Any video sequence may contain a large number of redundancies between the frames. So, the frame dropping attack is very common and effective on video watermarking. In our proposed method the watermark is embedded into



(a) Gaussian noise (σ =0.1): $\varrho_{LL} = -0.8181$, $\varrho_{HL} = 0.287$, $\varrho_{LH} = 0.7168$, $\varrho_{HH} = \mathbf{9616}$.

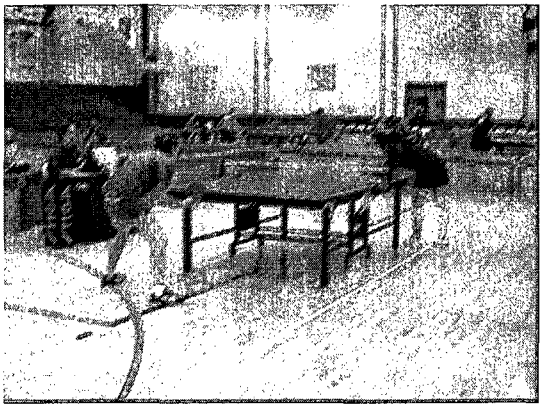
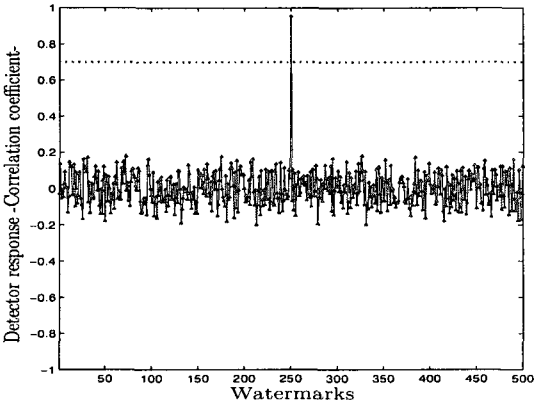


(b) Histogram Equalization: $\varrho_{LL} = -0.2815$, $\varrho_{HL} = 0.8507$, $\varrho_{LH} = 0.966$, $\varrho_{HH} = \mathbf{0.987}$.

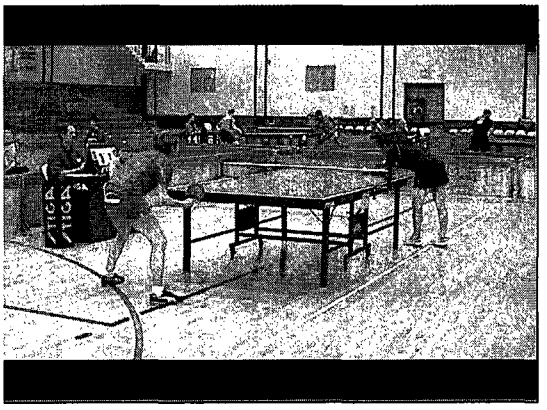
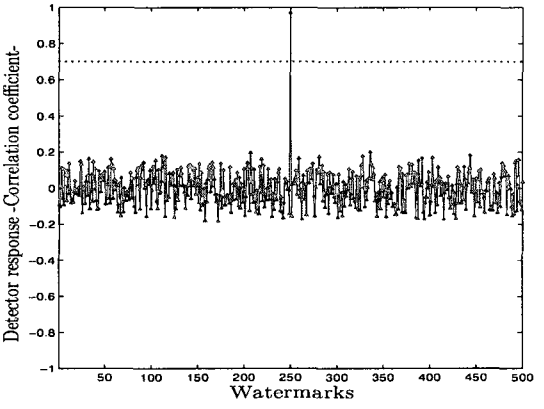


(c) Sharpening LL=0.9392, HL=0.8954, LH=0.9592, **HH=0.9935**.

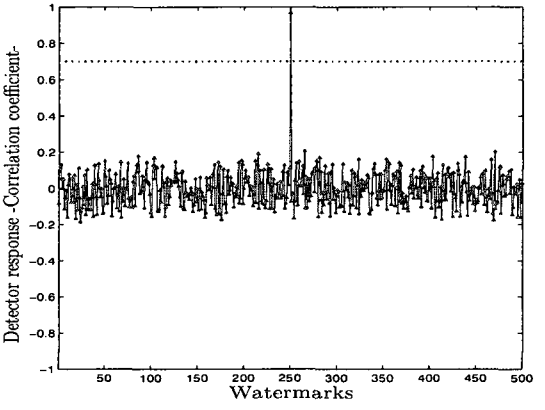
Figure 79: Illustration of one of the MPEG Tennis video I-frame under different attacks with the corresponding detector responses. The boldface numbers indicate the best correlation coefficient values

(d) Motion blurring: $\varrho_{LL} = \mathbf{0.9544}$, $\varrho_{HL} = 0.8953$, $\varrho_{LH} = 0.8096$, $\varrho_{HH} = 0.7556$.

(e) Cropping: $\varrho_{LL} = -0.0787$, $\varrho_{HL} = 0.9772$, $\varrho_{LH} = 0.9771$, $\varrho_{HH} = \mathbf{0.9771}$

(f) Rotation: $\varrho_{LL} = -0.3068$, $\varrho_{HL} = 0.8071$, $\varrho_{LH} = -0.8438$, $\varrho_{HH} = \mathbf{0.967}$.

Figure 80: Illustration of one of the MPEG Tennis video I-frame under different attacks with the corresponding detector responses. The boldface numbers indicate the best correlation coefficient values

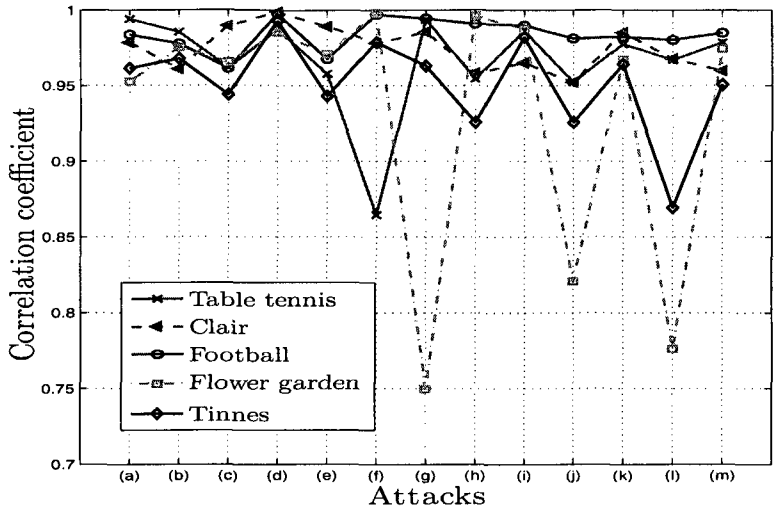


Figure 81: Best correlation coefficient results for different video sequences. The scaling factor α used in this experiment is 0.1 for the LL band and 0.05 for all the other sub-bands. (a) rescaling 100% – 50% – 100%, (b) salt&peppers noise 1%, (c) Gaussian noise $\sigma = 0.2$, (d) histogram equalization, (e) Gamma correction, (f) low-pass filtering, (g) sharpening, (h) motion blurring 45^o, (i) JPEG compression quality= 30%, (j) frame dropping 50%, (k) cropping 10% from the top and 10% from the bottom, (l) rotation 5^o, and (m) cropping 10% from the left and 10% from the right

Table 10: MPEG Tennis video under different attacks with the corresponding correlation coefficients. Boldface numbers indicate the best correlation

Attacks	ϱ_{LL}	ϱ_{HL}	ϱ_{LH}	ϱ_{HH}
Rescaling 100-50-100	**0.9939**	-0.3217	-0.0559	0.8656
Salt and peppers noise 1%	0.9853	0.8565	0.8849	**0.9748**
Gamma correction	0.6965	0.7171	0.8846	**0.9578**
Low-pass filtering	**0.9671**	-0.5129	-0.9127	0.8644
JPEG Compression	**0.9863**	0.9644	0.9555	0.9093
Frame Dropping 50%	-0.9925	0.9108	0.9206	**0.9522**
Cropping from left 10 and right 10	-0.9674	**0.9784**	0.9723	0.9759

frames of a scene, and due to the large amount of redundancies between frames, the calculated SVD for the 3D tensor will not change significantly by frame dropping up to 60% of the highly correlated frames. To test the performance of the proposed method against the frame dropping attack, we dropped different percentages of the video frames and then we obtained the correlation coefficients between the original watermark and the extracted watermark. As shown in Figure 85(a) the proposed method achieves better performance as compared to other methods. Similar results were obtained under frame swapping attack.

Frame averaging Frame averaging is another common attack in video watermarking. The attackers can use multiple frames and try to eliminate the watermark by statistical averaging of the watermarked video frames [78]. In the proposed algorithm we used different watermarks for each scene. This can prevent attackers from colluding with frames from completely different scenes to extract the watermark. Also, we used the same watermark within the same scene in order to prevent the attackers from statistically compare and remove the watermark from the motionless regions in the successive video frames. A video sequence of 8 scenes and 1100 frames was used to test the performance of the proposed method against this attack. Figure 82 shows the robustness of the proposed method against frame averaging attack.

Scaling Geometric transformations are the simplest attacks used to test the watermark detectors. Scaling is one of the very common geometric attacks in video watermarking. To investigate the robustness of the proposed method against this attack, we applied the scaling operation with factors of 50%, 70%, 110%, and 150% on the watermarked video frames. Figure 81 shows the robustness of the proposed scheme against scaling attack for different video sequences, and Figure 85(b) depicts the correlation coefficients for different watermarking schemes under the scaling attack.

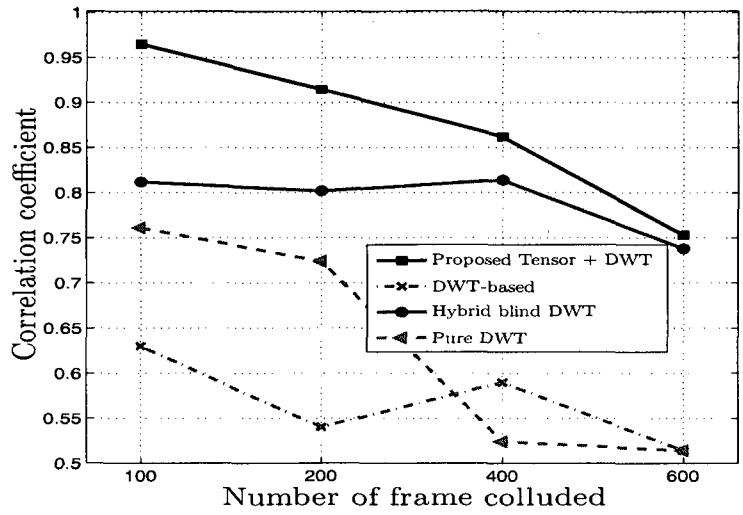


Figure 82: Robustness against averaging attack. Comparison results between the proposed scheme and the methods introduced in [16, 25, 66]

6.5.4 Comparisons with existing techniques

We also conducted several experiments to compare the robustness of the proposed method with existing techniques. Two types of comparisons are performed: in the first set we compare the proposed technique with two non-blind methods that were proposed in [6] and [25]. In the second set, we compare our results with three blind techniques that were proposed in [87], [16], and [66].

Comparisons with blind techniques

Comparing non-blind with blind techniques is not really a fair comparison. However, we have performed some experiments to compare the proposed scheme with three blind watermarking schemes proposed in [87], [16], and [66]. These experiments are done to indicate that the high robustness of the proposed method compared to the blind methods may help to overcome the limitations of the non-blind approachs. Figure 85 (a) shows the robustness against frame dropping attack, while Figure 85 (b) depicts the robustness against rescaling attack. Figure 85 (c) shows the result against

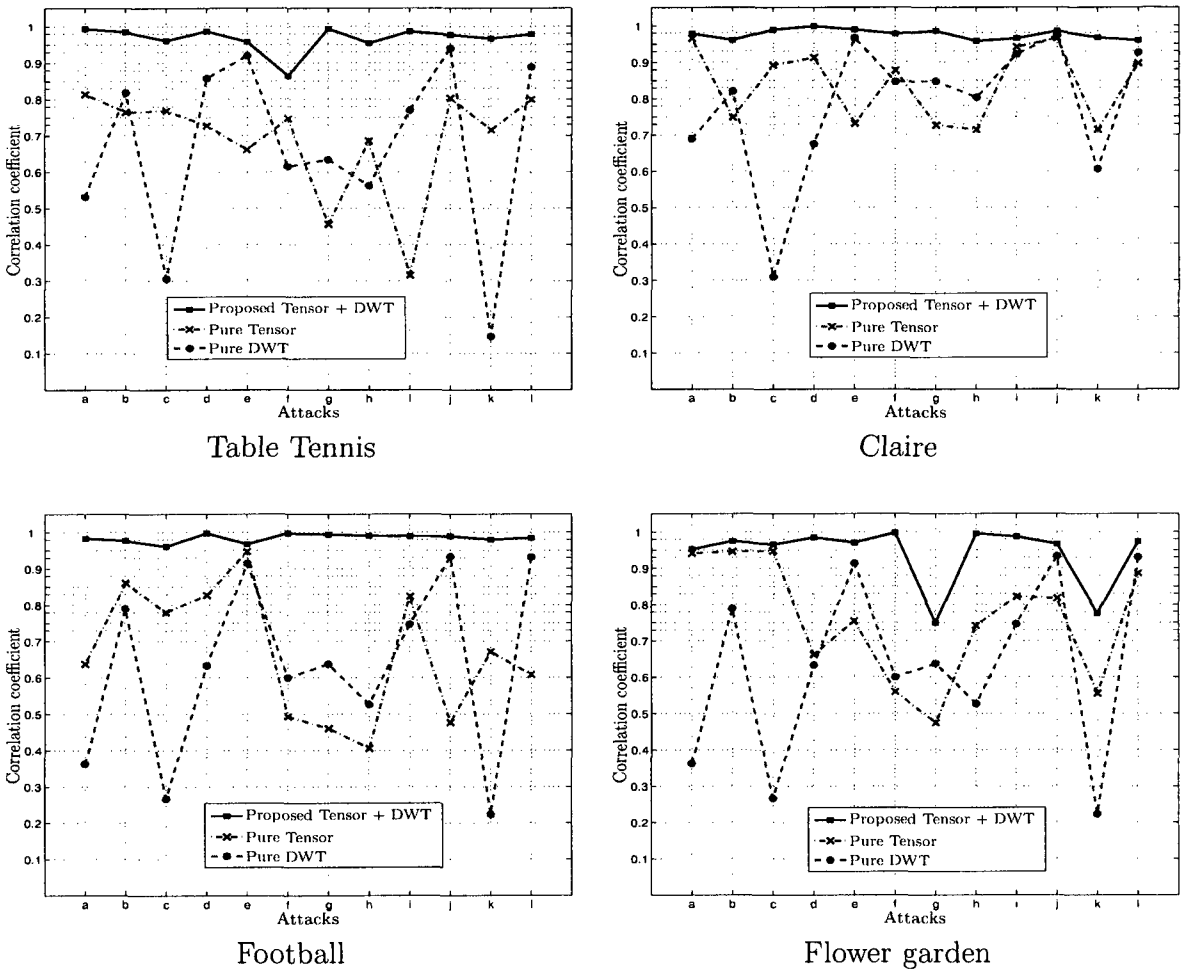


Figure 83: Correlation coefficient comparison results between the proposed scheme and the methods introduced in [6] and [25]. Different video sequences distorted by different attacks are used: (a) rescaling $100\% - 50\% - 100\%$, (b) salt&peppers noise (2%), (c) Gaussian noise $\sigma = 0.3$, (d) histogram equalization, (e) Gamma correction, (f) low-pass filter (3×3), (g) sharpening, (h) motion blurring 45^o, (i) JPEG compression quality= 25%, (j) cropping 8% from the top and 8% from the bottom, (k) rotation 20^o, (l) cropping 8% from the left and 8% from the right

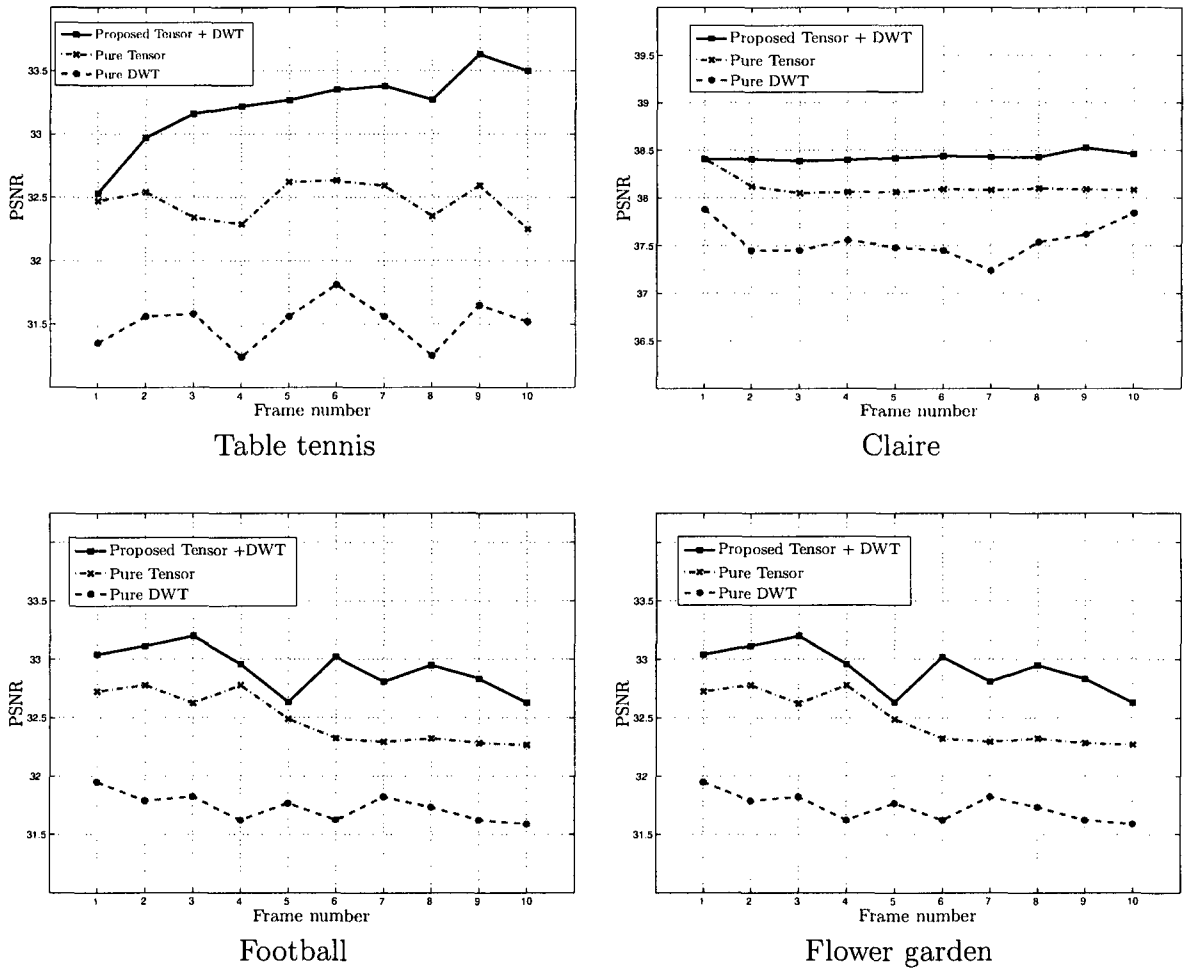


Figure 84: PSNR comparison results between the proposed scheme and the method introduced in [6] and [25]. Ten successive frames are chosen from one video scene of each video sequence

cropping attack. Figure 85 (d) tests the robustness against noise, rotation, compression, and low-pass filtering attacks. The results shown in Figure 85 clearly indicate that our proposed watermarking scheme performs the best in terms of robustness against attacks.

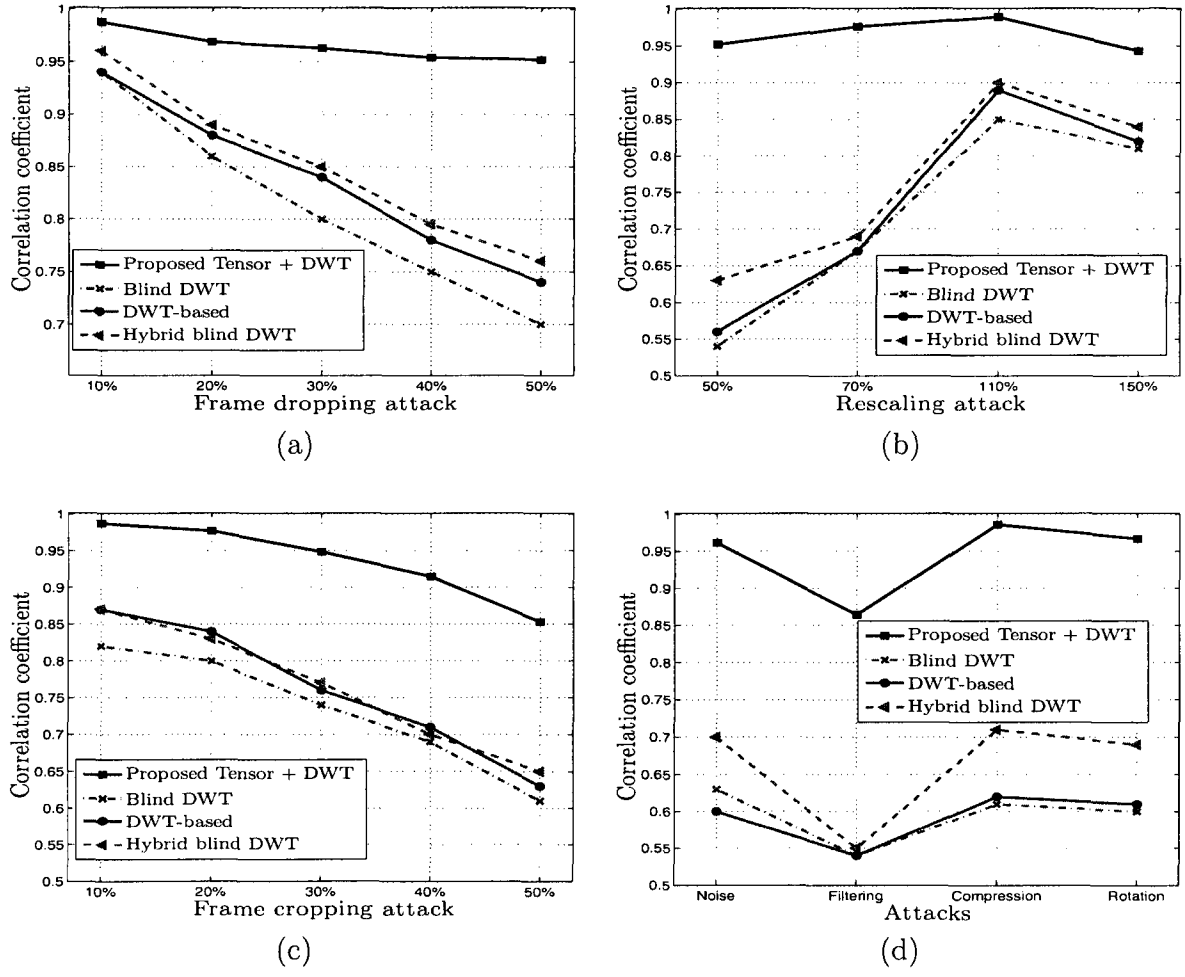


Figure 85: Comparison results between the proposed scheme and the methods introduced in [16] and [66]

Chapter 7

Conclusions and Future Research

This chapter briefly concludes the thesis and highlights the major contributions of this research. A brief description of the new watermarking schemes is included.

This thesis has presented robust watermarking algorithms for 2D images, 3D models, and video image sequences. We have demonstrated the use of these algorithms in multimedia copyright protection. The objective was to embed information about the owner of the data in order to prevent other parties from claiming the copyright on that data. We have demonstrated the effectiveness of the proposed methods through experimental results with a variety of 2D images, 3D models, and video image sequences. We have achieved the balance between the imperceptibility of the watermarked multimedia document and its robustness against intentional and geometric attacks.

In the next section, the contributions made in each of the previous chapters and the concluding results drawn from the associated research work are presented. Suggestions for future research directions related to this thesis are provided in Section 7.2.

7.1 Contributions of the Thesis

7.1.1 Robust and efficient image watermarking schemes using FHT

In Chapter 3, we proposed two simple and robust image watermarking methodologies for embedding a watermark in the transform domain. In the first technique, we introduced a high-rate of watermark embedding into digital images using FHT and SVD. We embedded the singular values of the watermark image in the DC components of the FHT blocks of the cover image. In the second technique, we proposed an improved method using FHT and DWT. The key idea of the latter approach is to encode the SVs of the watermark image after applying the FHT to small blocks computed from the four DWT sub-bands. To gain further insight into the robustness of the proposed techniques and the better visual imperceptibility, numerical experiments were provided to demonstrate the potential and the much improved performance of the proposed methods in comparison with other watermarking techniques.

7.1.2 Spectral graph-theoretic approach to 3D mesh watermarking

In Chapter 4, we presented the main issues related to the context of 3D shape watermarking. We exposed the main applications making use of such data and the need of reliable copyright protection techniques. The contribution of this chapter consisted of using the spectral compression of mesh geometry to propose a simple and computationally inexpensive watermarking methodology for embedding a watermark in the frequency domain of 3D models. We also used the fact that the rough approximation of the model may be reconstructed using small low-frequency spectral coefficients. The goal of our proposed scheme was to embed the watermark in the low-frequency

components by repeating the watermark embedding process as much as possible. We carried this out by encoding a watermark vector repeatedly into the spectral coefficients of the compressed 3D mesh. A nonlinear visual error was used to measure the perceptual quality of the watermarked 3D mesh. In Chapter 5, we introduced a 3D mesh watermarking technique based on nonnegative matrix factorization. The performance of these proposed 3D mesh watermarking methods was evaluated through extensive experiments that clearly showed an excellent resiliency against a wide range of attacks.

7.1.3 Video watermarking techniques using TSVD

In Chapter 6, we introduced simple, robust, and computationally inexpensive hybrid watermarking methodologies for embedding a watermark in the transform domain of an MPEG video. The primary motivation behind our proposed schemes was to use the scene change analysis to embed the watermark repeatedly in a fixed number of the intra-frames represented as a 3D tensor. This compact and computationally simple representation was then used to modify the singular values of the 3D tensor, which has a good stability and represents the video properties. We encoded a vector of random numbers into all the frequencies of the video scenes. We demonstrated through extensive experiments the excellent resistance of the proposed approaches against a wide range of attacks and also the better visual imperceptibility of the watermark.

7.2 Future Research Directions

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

7.2.1 Blind image watermarking

The watermarking schemes introduced in Chapter 3 provide robustness against several attacks. However, we note from our experimental results that significant additional performance gains from the proposed technique are still possible by using adaptive scaling factors for each block of the cover image. Also further studies are needed to implement a robust blind watermark recovery. Several blind image watermarking techniques have been proposed; however, their robustness is not high enough compared to the non-blind techniques.

7.2.2 Fully automatic 3D watermarking monitoring system

In the proposed watermarking scheme in Chapter 4, we need an initial search step in order to find the right 3D model for the watermark detection process. It is worth having a fast rejection process for the 3D models that do not match any of the models in the owner's databases without having to run completely the detection algorithm.

Theoretically, it is impossible to expect all possible attacks on 3D models. Also it is not easy to assess the degree of damage that a certain attack may have on a 3D model. We hope to develop a more robust system that can repel simplification and reordering vertices attacks without applying the remeshing process. Furthermore, it would be interesting to analyze the relationship among the number of basis vectors used in the compression process, watermark length, mesh partition size, and strength factor to further improve robustness against attacks.

On the other hand, we plan to apply this spectral analysis to other 3D applications, including 3D fingerprinting, mesh partitioning, and mesh smoothing.

7.2.3 Video watermarking scheme using tensor nonnegative matrix factorization and wavelet transform

Inspired by the successful results obtained in in the experiments in [31], where the nonnegative matrix factorization and wavelet transform are used in image watermarking, and the good robustness of the tensor SVD in video watermarking, we would like to extend the proposed algorithms and develop a robust watermarking scheme for video image sequences using tensor nonnegative matrix factorization.

We believe that the robustness of our approaches could be improved if we used an audio watermarking combined with our proposed tensor techniques. The watermark could be embedded in the audio channel. This watermark could provide error correction and detection for the video watermark. Error correction could have an important role especially when the watermark is significantly damaged. An error correcting code could eliminate the corruption of a watermark.

Bibliography

[1] E. E. Abdallah, A. Ben Hamza, P. Bhattacharya, "Improved image watermarking scheme using fast Hadamard and discrete wavelet transforms," *Journal of Electronic Imaging*, vol. 16, no. 3, pp. 033020.1-033020.9, 2007.

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

[3] E. E. Abdallah, A. Ben Hamza, P. Bhattacharya, "Watermarking 3D models using spectral mesh compression," Accepted for publication *Signal, Image and Video Processing*, Springer Journal 2008.

[4] E. E. Abdallah, A. Ben Hamza, P. Bhattacharya, "Spectral graph-theoretic approach to 3D mesh watermarking," *Proc. ACM Graphics Interface Conference*, pp. 327-334, Montreal, Canada, 2007.

[5] E. E. Abdallah, A. Ben Hamza, P. Bhattacharya, "Video watermarking using wavelet transform and tensor singular value decomposition," Submitted to *Signal, Image and Video Processing*, Springer Journal 2008.

[6] E. E. Abdallah, A. Ben Hamza, and P. Bhattacharya, "MPEG video watermarking using tensor singular value decomposition," *Proc. Int. Conf. Image Analysis and Recognition – Lecture Notes in Computer Science,* vol. 4633, pp. 772-783, Montreal, Canada. 2007.

[7] E. E. Abdallah, A. Ben Hamza, and P. Bhattacharya, "A Robust 3D watermarking technique using eigen-decomposition and nonnegative matrix factorization," *Proc. Int. Conf. Image Analysis and Recognition – Lecture Notes in Computer Science,* vol. 5112, pp. 253-262, Portugal, 2008.

[8] P. R. Alface, M. D. Craene, and B. Macq, "Three-dimensional image quality measurement for the benchmarking of 3D watermarking schemes," *Proc. Secu., Steganography, and Watermarking of Multimedia Contents*, pp. 230-240, 2005.

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

[10] B. W. Bader and T. G. Kolda, "MATLAB tensor classes for fast algorithm prototyping," *ACM Trans. Mathematical Software*, vol. 32, no. 4, pp. 635-653, 2006.

[11] A. Ben Hamza and H. Krim "Geodesic matching of triangulated surfaces," *IEEE Trans. Image Processing*, vol. 15, no. 8, pp. 2249-2258, 2006.

[12] O. Benedens, "Geometry-based watermarking of 3-D polygonal models," *IEEE Computer Graphics and Applications*, vol. 19, no. 1, pp. 46-45, 1999.

[13] J. B. Besl and D. N. Mckay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, no. 2, pp. 239- 256, 1992.

[14] S. Bhattacharya, T. Chattopadhyay, and A. Pal, "A survey on different video watermarking techniques and comparative analysis with reference to H.264/AVC," *Proc. IEEE Int. Symp. Consumer Electronics*, pp. 1-6, 2006.

[15] G. Braudaway, "Protecting publicly-available images with an invisible image watermark," *Proc. IEEE Int. Conf. Image Processing,* pp. 524-527, 1997.

[16] P. W. Chan, M. R. Lyu, and R.T. Chin, "A novel scheme for hybrid digital video watermarking: approach evaluation and experimentation," *IEEE Trans. Circuits and Systems for Video Technology*, vol 15, no. 12, pp. 1638-1649, 2005.

[17] D. S. Chandra, "Digital image watermarking using singular value decomposition," *Proc. IEEE Symp. Circuits and Sys.*, pp. 264-267, 2002.

[18] J. W. Cho, R. Prost, and H. Y. Jung, "An oblivious watermarking for 3-D polygonal meshes using distribution of vertex norms," *IEEE Trans. Signal Processing*, vol. 55, no. 1, pp. 142-155, 2007.

[19] D. Cotting, T. Weyrich, M. Pauly, and M. Gross, "Robust watermarking of point-sampled geometry," *Proc. Int. Conf. Shape Modeling and Application*, pp. 233-242, 2004.

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

[21] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Morgan Kaufmann, San Francisco, 2001.

[22] F. Deguillaume, G. Csurka, J. ÒRuanaidh, and T. Pun, "Robust 3D DFT video watermarking," *Proc. Security and Watermarking of Mult. Content*, vol. 3657, pp. 113-124, 1999.

[23] I. Djurovic, S. Stankovic, and I. Pitas, "Digital watermarking in the fractional fourier transformation domain," *Journal of Network and Computer Applications*, vol. 24, no. 4, pp. 167173, 2001.

[24] E. Elbasi and A. M. Eskicioglu, "A dwt-based robust semi-blind image watermarking algorithm using two bands," *Proc. Symposium on Electronic Imaging,*

Security, Steganography, and Watermarking of Multimedia Contents, pp. 777-787, 2006.

[25] E. Elbasia and A. M. Eskicioglu, "MPEG-1 video semi-blind watermarking algorithm in the DWT domain," *Proc. IEEE Int. Symp. Broadband Multimedia Systems and Broadcasting*, Las Vegas, 2006.

[26] E. Ganic, D. S. Dexter, and A. M. Eskicioglu, "Embedding multiple watermarks in the DFT domain using low-and high-frequency bands," *Proc. Symposium Elect. Imag. Secu. Steganography and Watermarking of Multimedia Contents*, pp. 175-184, 2005.

[27] E. Ganic and A. M. Eskicioglu, "Robust DWT-SVD domain image watermarking: embedding data in all frequencies," *Proc. ACM Multimedia and Security Workshop*, pp. 166-174, 2004.

[28] E. Ganic, N. Zubair, and M. Eskicioglu, "An optimal watermarking scheme based on singular value decomposition," *Proc. Comm., Network, and Information Security*, pp. 527-533, 2003.

[29] E. Garcia and J. L. Dugelay, "Texture-based watermarking of 3-D video objects," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 853-866, 2003.

[30] M. Garland and P. Heckbert, "Surface simplification using quadric error metrics," *Proc. SIGGRAPH*, pp. 209-216, 1997.

[31] M. Ghaderpanah and A. Ben Hamza, "Nonnegative matrix factorization scheme for digital image watermarking," *Proc. IEEE Int. Conf. Multimedia & Expo.*, pp. 1573-1576, 2006.

[32] S. Gilani and A. Skodras, "Watermarking by multiresolution Hadamard transform," *Proc. Electr. Imaging Visual Arts*, pp. 73-77, 2001.

[33] S. Gumhold and W. Strasser, "Real time compression of triangle mesh connectivity," *Proc. SIGGRAPH*, pp. 133-140, 1998.

[34] I. Guskov, W. Sweldens, and P. Schroeder, "Multiresolution signal processing for meshes," *Proc. SIGGRAPH*, pp. 325-334, 1999.

[35] A. D. Gwenael and J. L. Dugelay, "A guide tour of video watermarking," *Signal Processing: Image Commun.*, vol. 18, no. 4, pp. 263-282, 2003.

[36] T. Harte and A. Bors, "Watermarking 3d models," *Proc. IEEE Int. Conf. Image Processing*, pp. 661-664, 2002.

[37] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *IEEE Trans. Signal Processing*, vol. 66, no. 3, pp. 283-301, 1998.

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

[39] J. Hernandez, F. Gonzalez, J. Rodriguez, and G. Nieto, "Performance analysis of a 2-d-multipulse amplitude modulation scheme for data hiding and watermarking of still images," *IEEE Journal Selected Areas in Communications,* Vol. 16, no. 4, pp. 510-524, 1998.

[40] M. Isenburg and J. Snoeyink, "Mesh collapse compression," *Proc. 12th Brazilian Sympo. Computer Graphics Image Processing*, pp. 27- 28, 1999.

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

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

[43] S. Katzenbeisser and F. Petitcolas, *Information Hiding Techniques for Steganography and Digital Watermarking,* Artech House, 1999.

[44] T. G. Kolda, "MATLAB Tensor Toolbox," Version 2.2, http://csmr.ca.sandia.gov/ tgkolda/TensorToolbox/, 2007.

[45] K. Kwon, S. Kwon, S. Lee, T. Kim, and K. Lee, "Watermarking for 3D polygonal meshes using normal vector distributions of each patch," *Proc. Int. Conf. Image Processing*, pp. 499-502, 2003.

[46] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Image Processing*, vol. 10, no. 1, pp. 148-158, 2001.

[47] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking digital image and video data. A state-of-the-art overview," *IEEE Signal Processing Mag.*, vol. 17, no. 5, pp. 20-46, 2000.

[48] L. D. Lathauwer, B. D. Moor, and J. Vandewalle, "A multilinear singular value decomposition," *SIAM J. Matrix Anal. Appl.*, vol. 21, no. 4, pp. 1253-1278, 2000.

[49] D. Lee and H. Seung,"Algorithms for nonnegative matrix factorization," *Adv. in Neural Info. Proc. Systems,* 13, 2000.

[50] D. Lee and H. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature,* vol. 401, pp. 788-791, 1999.

[51] L. Li, Z. Pan, M. Zhang, and K. Ye, "Watermarking subdivision surfaces based on addition property of Fourier transform," *Proc. Int. Conf. Computer Graphics and Interactive Tech.*, pp. 46-49, 2004.

[52] E. T. Lin, "Video and image watermarking synchronization," Ph.D. dissertation, Purdue University, 2005.

[53] E. Lin, A. Eskicioglu, R. Lagendijk, and E. Delp, "Advances in digital video content protection," *Proc. IEEE: Special Isssue on Advance in Video Coding and Delivery,* Vol. 93, no. 1, pp. 171-183, 2005.

[54] Y. R. Lin and W. H. Hsu, "An embedded watermark technique in video for copyright protection," *Proc. Int. Conf. Pattern Recog.*, vol. 4, pp. 795-798, 2006.

[55] H. Liu and L. Chang, "Real time digital video watermarking for digital rights management via modification of VLCS," *Proc. Int. Conf. Parallel and Distributed Systems*, vol. 2, pp. 295-299, 2005.

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

[57] R. Mehul and R. Priti, "Discrete wavelet transform based multiple watermarking scheme," *Proc. IEEE Region Technical Conf. Convergent Technology,* pp. 935-938, 2003.

[58] N. Memon and P. Wong, "Digital watermarks: protecting multimedia content," *Comm. of the ACM*, vol. 47, no. 7, pp. 35-43, 1998.

[59] A. S. Mian, M. Bennamoun, and R. Owens, "A novel representation and feature matching algorithm for automatic pairwise registration of range images", *Int. J. of Computer Vision*, vol. 66, no. 1, pp. 19-40, 2006.

[60] M. K. Mihcak and R. Venkatesan, "Blind image watermarking via derivation and quantization of robust semi-global statistics," *Proc. IEEE Int. Conf. Acoustics, Speech and Signal Processing,* vol. 4, pp. 3453-3456, 2002.

[61] F. Mintzer, G. Braudaway, and M. Yeung, "Eficfective and inefiective digital watermarks," *Proc. Intational Conf. Image Processing,* pp. 9-12, 1997.

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

[63] E. Muharemagic, "Adaptive two-level watermarking for binary document images," Ph.D. dissertation, Florida Atlantic University, 2004.

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

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

[66] X. Niu and S. Sun, "A new wavelet-based digital watermarking for video," *Proc. IEEE Digital Signal Processing Workshop*, Texas, 2000.

[67] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal meshes," *Proc. ACM Multimedia*, pp. 261-272, 1997.

[68] R. Ohbuchi, H. Masuda, and M. Aono, "Watermarking three-dimensional polygonal models through geometric and topological modifications," *IEEE J. Selected Areas in Comm.*, vol. 16, no. 4, pp. 551-560, 1998.

[69] R. Ohbuchi, A. Mukaiyama, and S. Takahashi, "A Frequency domain approach to watermarking 3D shapes," *Computer Graphics Forum*, vol. 21, no. 3, pp. 373-382, 2002.

[70] R. Ohbuchi, S. Takahashi, T. Miyasawa, and A. Mukaiyama, "Watermarking 3-D polygonal meshes in the mesh spectral domain," *Proc. Computer Graphics Interface*, pp. 9-17, 2001.

[71] S. W. Park, M. Savvides, "Individual kernel tensor-subspaces for robust face recognition: A computationally efficient tensor framework without requiring mode factorizatin," *IEEE Trans. Systems, Man and Cybernetics*, vol. 37, no. 5, pp. 1156-1166, 2007.

[72] F. P. Petitcolas, R. J. Anderson, and M.G. Kuhn, "Attacks on copyright marking systems," *Proc. Workshop Info. Hiding*, pp. 218-238, 1998.

[73] F. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Information hiding-a survey," *Proc. IEEE special issue on protection of multimedia content*, vol. 87, no. 7, pp. 1062-1078, 1999.

[74] E. Praun, H. Hoppe, and A. Finkelstein, "Robust mesh watermarking," *Proc. SIGGRAPH*, pp. 49-56, 1999.

[75] J. J. Qiu, D. M. Ya, B. H. Jun, and P. Q. Sheng, "Watermarking on 3D mesh based on spherical wavelet transform," *J. Zhejiang Univ. Sci.*, vol. 5, no. 3, pp. 251-258, 2004.

[76] M. R. Raghuveer and S. B. Ajit, *Wavelet Transforms - Introduction to Theory and Applications*, Addison-Wesley, 2000.

[77] W. J. Rey, *Introduction to Robust and Quasi-robust Statistical Methods*, Springer, Berlin, Heidelberg, 1983.

[78] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scene-based video watermarking using perceptual models," *IEEE Jour. Selected Areas in Commun.*, Vol. 16, No. 4, pp. 540-550, 1998.

[79] P. Tao and A. M. Eskicioglu, "A robust multiple watermarking scheme in the DWT domain," *Proc. Optics East Symp., Internet Multimedia Management*, pp. 133-144, 2004.

[80] F. Uccheddu, M. Corsini, and M. Barni, "Wavelet-based blind watermarking of 3d models," *Proc. ACM Multimedia and Security Workshop*, pp. 143-154, 2004.

[81] C. Vleeschouwer, J. Delaigle, and B. Macq, "Invisibility and application functionalities in perceptual watermarking-an overview," *Proc. IEEE,* Vol. 90, no. 1, pp. 64-77, 2002.

[82] J. Vollmer, R. Mencl, and H. Muller, "Improved Laplacian smoothing of noisy surface meshes," *Proc. EUROGRAPHICS*, pp. 131-138, 1999.

[83] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun, "Attack modelling: Towards a second generation watermarking benchmark," *Signal Processing,* Vol. 81, no. 6, pp. 1177-1214, 2001.

[84] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su, " Attacks on digital watermarks: classification, estimation based attacks, and benchmarks," *IEEE Comm. Mag.*, vol. 39, no. 8, pp. 118-126, 2001.

[85] Y. Wang, J. F. Doherty, and R. E. Van Dyck, "A Wavelet-based watermarking algorithm for ownership verification of digital images," *IEEE Trans. Image Processing*, vol. 11, no. 2,pp. 77-88, 2002.

[86] Y. Wang and A. Pearmain, "Blind image data hiding based on self reference," *Pattern Recognition Letters*, Vol. 2, No. 15, pp. 1681-1689, 2004.

[87] Y. Wang and A. Pearmain, "Blind MPEG-2 video watermarking robust against geometric attacks: a set of approaches in DCT domain," *IEEE Trans. Image Processing*, vol. 15, no 6, pp. 1536-1543, 2006.

[88] S. Winkler, E. Drelie Gelasca, and T. Ebrahimi, "Toward perceptual metrics for video watermark evaluation," *Proc. SPIE Applications of Digital Image Processing*, vol. 5203, pp. 371-378, 2003.

[89] J. Wu and L. Kobbelt, "Efficient spectral watermarking of large meshes with orthogonal basis functions," *The Visual Computer*, vol. 21, no. 8-10 pp. 848-857, 2005.

[90] M. Yeung and F. Mintzer, "Digital watermarking for high-quality imaging," *Proc. Signal Processing Society Workshop Multimedia,* pp. 357-362, 1997.

[91] S. Zafeiriou, A. Tefas, and I. Pitas, "Blind robust watermarking schemes for copyright protection of 3D mesh objects," *IEEE Trans. Visualization and Computer Graphics*, vol. 11, no. 5, pp. 596-607, 2005.

[92] Y. Zhang and A. Ben Hamza, "PDE-based smoothing for 3D mesh quality improvement," *Proc. IEEE Int. Conf. Electro/Info. Technol.*, pp. 334-337, 2006.

[93] Y. Zhang and A. Ben Hamza, "Vertex-based diffusion for 3D mesh denoising," *IEEE Trans. Image Processing*, vol. 16, no. 4, pp. 1036-1045, 2007.

List of Acronyms

Three-Dimensional	$3D$
Two-Dimensional	$2D$
Fast Hadamard Transform	FHT
Inverse Fast Hadamard Transform	IFHT
Discrete Wavelet Transform	DWT
Inverse Discrete Wavelet Transform	IDWT
Singular Value Decomposition	SVD
Singular Values	SVs
Discrete Cosine Transform	DCT
Inverse Discrete Cosine Transform	IDCT
Discrete Fourier Transform	DFT
Inverse Discrete Fourier Transform	IDFT
Computer Aided Design	CAD
Moving Picture Experts Group	MPEG
Tensor Singular Value Decomposition	TSVD

High-order Tensor Singular Value Decomposition	HTSVD
Low-Low	LL
Low-High	LH
High-Low	HL
High-High	HH
Point-spread Function	PSF
Direct Current	DC
Peak Signal-to-Noise Ratio	PSNR
Iterative Closest Point	ICP
Nonnegative Matrix Factorization	NMF
Inverse Nonnegative Matrix Factorization	INMF
Variable Length Codes	VLCs