UNIVERSITY OF CALIFORNIA,
IRVINE

Robust Data Hiding in Multimedia for Authentication and Ownership Protection

DISSERTATION

submitted in partial satisfaction of the requirements
for the degree of

DOCTOR OF PHILOSOPHY

in Electrical and Computer Engineering

by

Farhan A. Alenizi

Dissertation Committee:
Professor Fadi Kurdahi, Chair
Professor Ahmed Eltaweel
Professor Rainer Doemer

2017

DEDICATION

To my parents...

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACKNOWLEDGMENTS

I would like to give my sincere thankfulness to my dissertation advisor professor Fadi Kurdahi. I am very grateful for his constant support and encouragement, for his constructive advices and patient guidance. I would have never finished this dissertation without his personal commitment and help. I am also very thankful to professor Ahmed Eltaweel, his advices and suggestions were very helpful and fruitful.

CURRICULUM VITAE

Farhan A. Alenizi

EDUCATION

Master of Science in Electrical Engineering **2006**
King Saud University, Riyadh *Riyadh, Saudi Arabia*

Bachelor of Science in Electrical Engineering **1999**
King Saud University, Riyadh *Riyadh, Saudi Arabia*

RESEARCH EXPERIENCE

Graduate Researcher **2012–2017**
University of California, Irvine *Irvine, California*

Graduate Researcher **2010–2012**
Prince Sattam bin Abdulaziz University *Al-Kharj, Saudi Arabia*

Graduate Researcher **2004–2006**
King Saud University, Riyadh *Riyadh, Saudi Arabia*

TEACHING EXPERIENCE

Teaching Assistant **2010–2011**
Prince Sattam bin Abdulaziz University *Al-Kharj, Saudi Arabia*

Teaching Assistant **2016**
University of California, Irvine *Irvine, California*

SELECTED HONORS AND AWARDS

Graduate Research Scholarship **2012-2017**
Prince Sattam bin Abdulaziz University *Al-Kharj, Saudi Arabia*

REFEREED JOURNAL PUBLICATIONS

Hybrid Pyramid-DWT-SVD Dual Data Hiding Technique for Videos Ownership Protection **2017**
EURASIP Journal on Information Security (under editing for publication)

A Pyramid-Based Watermarking Technique for Digital Images Copyright Protection Using Discrete Wavelet Transforms Techniques **2013**
INTECH Open Access Publisher

REFEREED CONFERENCE PUBLICATIONS

A pyramid-based watermarking technique for digital color images copyright protection **2009**
International Conference on Computing, Engineering and Information, 2009. ICC'09.

DWT-based watermarking technique for video authentication **2015**
IEEE International Conference on Electronics, Circuits, and Systems (ICECS), 2015

3D Mesh Optimal Watermarking Technique **2017**
Asilomar Conference on Signals, Systems, and Computers, 2017 (submitted)

Privacy Enhancing Techniques Against Criminal Use of Data Hiding **2017**
Under Preparation

BOOKS

Watermarking for Multimedia Using Pyramid Decomposition Techniques **2006**
King Saud University publications

PROFESSIONAL MEMBERSHIPS

Institute of Electrical and Electronics Engineers (IEEE)

ABSTRACT OF THE DISSERTATION

Robust Data Hiding in Multimedia for
Authentication and Ownership Protection

By

Farhan A. Alenizi

Doctor of Philosophy in Electrical and Computer Engineering

University of California, Irvine, 2017

Professor Fadi Kurdahi, Chair

Establishing robust and blind data hiding techniques in multimedia is very important for authentication, ownership protection and security. The multimedia being used may include images, videos and 3D mesh objects.

A hybrid pyramid Discrete-Wavelet-Transform (DWT) Singular-Value-Decomposition (SVD) data hiding scheme for video authentication and ownership protection is proposed. The data being hidden will be in the shape of a main color logo image watermark and another secondary Black and White (B&W) logo image. The color watermark will be decomposed to Bit-Slices. A pyramid transform is performed on the Y-frames of a video stream resulting in error images; then, a Discrete Wavelet Transform (DWT) process is implemented using orthonormal filter banks on these error images, and the Bit-Slices watermarks are inserted in one or more of the resulting subbands in a way that is fully controlled by the owner; then, the watermarked video is reconstructed. SVD will be performed on the color watermark Bit-Slices. A secondary B&W watermark will be inserted in the main color watermark using another SVD process. The reconstruction was perfect without attacks, while the average Bit-Error-Rates (BER's) achieved under attacks are in the limits of 2% for the

color watermark and 5% for the secondary watermark; meanwhile, the mean Peak Signal-to-Noise Ratio (PSNR) is 57 dB. Furthermore, a selective denoising filter to eliminate the noise in video frames is proposed; and the performance with data hiding is evaluated.

Moreover, a 3D mesh blind optimized watermarking technique is proposed in this research. The technique relies on the displacement process of the vertices locations depending on the modification of the variances of the vertices's norms. Statistical analysis were performed to establish the proper distributions that best fit each mesh, and hence establishing the bins sizes. Experimental results showed that the approach is robust in terms of both the perceptual and the quantitative qualities.

In conclusion, the degree of robustness and security of the proposed techniques are shown. Also the schemes that can be adopted to further enhance the performance, and the future work that can be done in the field are introduced.

Chapter 1

Introduction

1.1 Problem Importance

With the growth and advances in digital communication technologies, images, audio, and video have become easy to be delivered and exchanged. These forms of digital information can be easily copied and distributed through digital media. The ease with which these information types can be used and manipulated may lead to unauthorized copying. These concerns motivated significant research to hide copyright messages and serial numbers into digital media. The target is to distinguish any violations and protecting the ownership. Data hiding in multimedia recently are being used for many applications such as authentication, ownership protection, access control and annotation [1]. New progress in digital technologies, such as compression techniques, has brought new challenges to data hiding, as these techniques are able to make some modifications in the multimedia without affecting its quality. Importance of information hiding can be seen in the spread of its applications. In military organizations, for example, the detection of a signal in a modern battle field may lead

to an attack on the signaler. For this reason, military communications use techniques such as spread spectrum modulation to make signals hard for the enemy to detect or jam. Information hiding techniques can also be used in situations where deniability is required, such as fair voting, personal privacy or limitation of liability [2]. The healthcare industry and especially medical imaging can benefit from information hiding techniques. For example, they use methods to separate image data from caption, such as the name of the patient, the date, and the physician [3]. Other applications can be seen in the context of multimedia, such as tamper proofing, data augmentation for the benefit of the public, and automatic monitoring of copyright material on the Web [4]. There are two important disciplines of information hiding, first is steganography which literally means (covered writing); in contrast with cryptography which protects the content of messages, steganography conceals their very existence. This can be done by hiding the information in other information. The other discipline of information hiding is watermarking. Watermarking has the additional requirement of robustness against possible attacks; a successful attacker will try to make the mark undetectable. The other difference between these two techniques is that the hidden information in watermarking is associated with the digital object while this is not the case in steganography.

1.2 Historical Background of Information Hiding

The most famous examples of steganography go back to antiquity. In his histories, Herodotus (486-425 B.C) tells how around 440 B.C Histiaeus shaved the head of his slave and wrote a message which disappeared when the hair had regrown [5]. Herodotus also tells how a Greek man warned his king of an invasion: he removed the wax from a writing tablet, wrote his message on the wood underneath and then

covered the message with wax; the tablet looked exactly as a blank one. In 1857, Brewsler suggested hiding secret messages in spaces not larger than a full stop or a small dot of ink. In the Franco-Prussian War of 1870-1871, messages on microfilms were sent out by pigeon post. During the Russo-Japanese War of 1905, microscopic images were hidden in ears, nostrils, and under fingerprints. In the World War I, messages to and from spies were reduced to microdots by several stages of photographic reduction. Invisible ink has been used extensively. Progress in chemistry helped create more sophisticated combinations of ink and developer by the first world war, but the technology fell into disuse with the invention of universal developers which could determine which parts of a piece of paper had been welted from the effects on the surface of fibers. Watermarks in papers are good anticounterfeiting techniques. New innovations include ultraviolet fluorescent inks used in printing travelers checks [6]. Another type of steganography is the linguistic steganography which is used in acrostic. The most famous example is Giovanni Boccaccios (1313-1375) Amorosa vision. John Wilkins, in the 17th century, explained how on can hide secretly a message into a geometric drawing using points, lines, or triangles. The points, the ends of the lines and the angles of the figures do each of them by their different situation express a specific letter. An improvement was made when the message is hidden at random locations of the cover-text. This is the idea of most of current steganographic systems. In a method developed in Ancient China, the sender and the receiver had copies of paper masks with holes at random locations. The sender would place his mask over a sheet of paper, write the secret message into the holes, removes the mask, and then compose a message incorporating the hidden message. The receiver would place the mask on the cover message and read the hidden message in the holes. An example of copy protection is the catalog of signal images of the painter Claude Lorrain (1600 1682). To protect his paints, he introduced a method that enables him to recognize any forgery or misuse of his works. Lorrain kept a book containing a collection of

drawings in the form of a sketchbook. Similar technique is being used today, where a central database of images digests is kept.

1.3 Review of Data Hiding Techniques

Data hiding in multimedia has emerged as a solution of problems of ownership verifications and authentications [1]. Figure 1.1 shows how information hiding can be broken down into different areas. One main discipline of data hiding is Steganography, which is derived from the Greek for covered writing and essentially means to hide in plain sight". Steganography is the art and science of communicating in such a way that the presence of a message cannot be detected. Simple steganographic techniques have been in use for hundreds of years, but with the increasing use of files in an electronic format, new techniques for information hiding have become possible. Steganography can be used to hide a message intended for later retrieval by a specific individual or group. In this case the aim is to prevent the message being detected by any other party. The other major area of data hiding is copyright marking, where the message to be inserted is used to assert copyright over a document. This can be further divided into watermarking and fingerprinting [1].

Data hiding and encryption are both used to ensure data confidentiality. However the main difference between them is that with encryption, anybody can see that both parties are communicating in secret. Data embedding techniques hide the existence of a secret message, and in the best case, nobody can see that both parties are communicating in secret. This makes data hiding suitable for some tasks for which encryption isn't, such as copyright marking. Adding encrypted copyright information to a file could be easy to remove, but embedding it within the contents of the file itself can prevent it from being easily identified and removed. Encryption allows secure

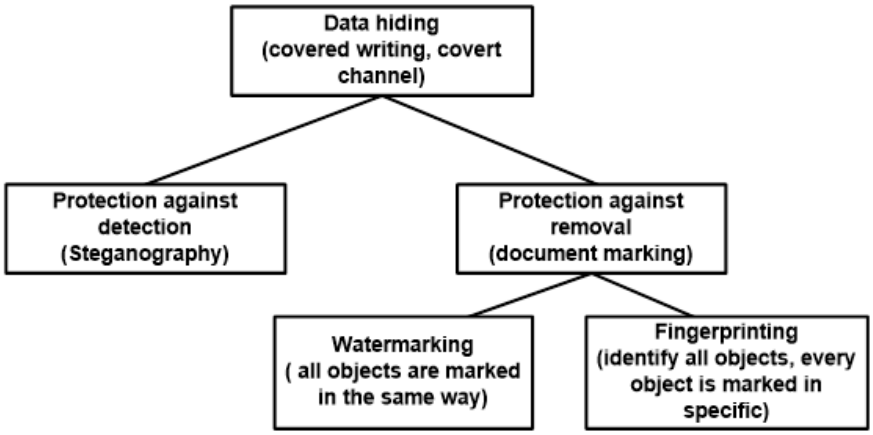


Fig. 1.1: Different areas of data hiding techniques

communication requiring a key to read the information. An attacker cannot remove the encryption but it is relatively easy to modify the file, making it unreadable for the intended recipient. Digital signatures allow authorship of a document to be asserted. The signature can be removed easily but any changes made will invalidate the signature, therefore integrity is maintained. Data hiding provides a means of secret communication which cannot be removed without significantly altering the data in which it is embedded. The embedded data will be confidential unless an attacker can find a way to detect it. Steganography, which is a major discipline of data hiding, conceals existence of messages. This can be done by hiding the information in other information. Watermarking, on the other hand, has the additional requirement of robustness against possible attacks. A successful attacker will try to make the mark undetectable. The other difference between these two techniques is that the hidden information in watermarking is associated with the digital object while this is not the case in steganography [1]. Furthermore, Capacity of hidden data in steganography is higher than watermarking. Table. 1.1 shows the main differences between steganography and watermarking [7].

Table 1.1: Steganography vs. watermarking

	Steganography	**Watermarking**
Attacks	Detection of hidden data	Disabling or removing hidden data
Communication	Point-to-point	One-to-many
Cover-object	Not important	Important
Amount of information	Large amount	Small amount

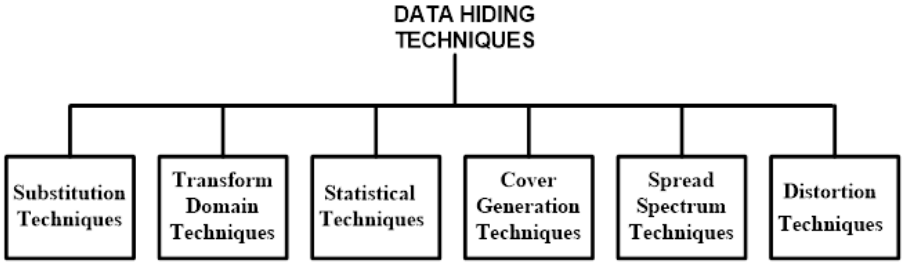


Fig. 1.2: Categories of data hiding techniques

Classifications of information hiding techniques can be done according to the modifications that are being applied to the cover message during the embedding process. Depending on this classification, it is possible to group the information hiding methods into six categories as shown in Fig. 1.2. There are other ways of classifying the data hiding methods, but these categories give more detailed picture of the techniques. Moreover, data hiding techniques that are used in images can be used in videos; so when we talk about images that includes videos as well.

1.3.1 Substitution Techniques

Basic substitution systems try to encode secret information by substituting insignificant parts of the cover by secret message bits [8]. The receiver can extract the hidden information if she has knowledge of the positions of embedding. Since the changes are assumed to be minor, the embedder does not expect that the changes will be noticed. The most common example of this technique is the Least Significant Bit (LSB) insertion. The approaches depending on LSB insertion are common and easily

applied in images and audio [1]. The embedding process consists of choosing a subset of cover elements, and performing the substitution operation that changes the LSB of the elements. It is possible to change more than one bit of the cover. Fig. 1.3 shows a basic example of watermarking RGB color image using LSB substitution in space domain. LSB substitution, however, despite its simplicity brings some drawbacks. Although it may survive transformations such as cropping, any addition of noise or lossy compression is likely to defeat the embedding process. An even better attack would be, to simply set the LSB bits of each pixel to one. This will destroy the hidden information with negligible impact on the cover object. Furthermore, once the algorithm is discovered, the embedded information could be easily modified by an intermediate party. An improvement on basic LSB substitution would be to use a pseudo-random number generator to determine the pixels to be used for embedding based on a given seed or key [1].

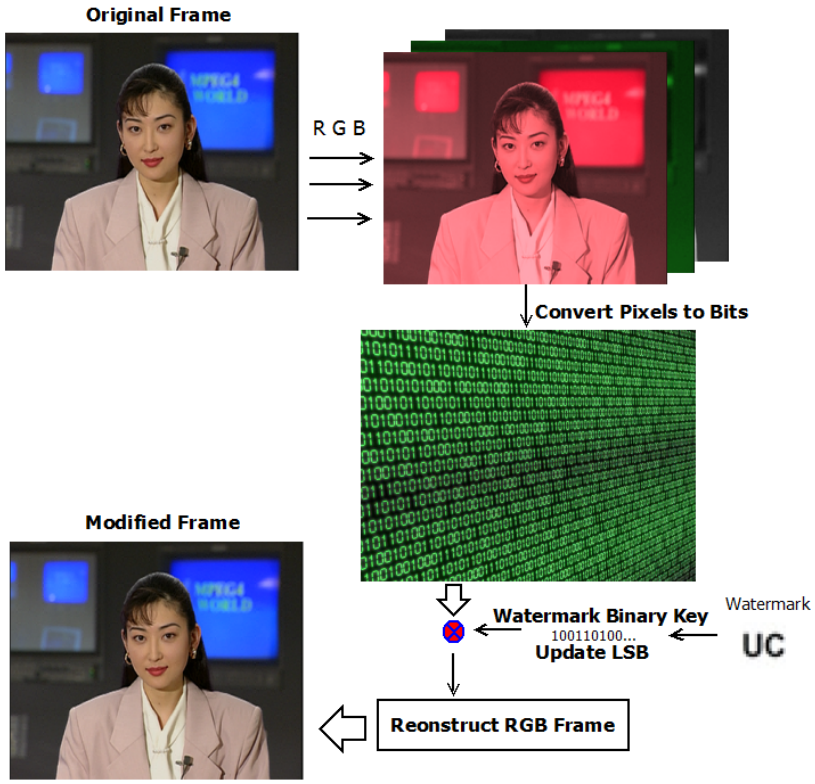


Fig. 1.3: Watermarking RGB color image using Spatial LSB substitution.

1.3.2 Transform Domain Techniques

An advantage of the substitution techniques discussed above is that they can be easily applied to any image; regardless of subsequent processing [9, 10]. A possible disadvantage of substitution techniques is that they do not allow for the exploitation of the subsequent processing in order to increase the robustness of the hiding process. In addition, adaptive information hiding techniques are a bit more difficult in the spatial domain. Both the robustness and quality of the hiding process could be improved if the properties of the cover image could similarly be exploited. For instance, it is generally preferable to hide information in noisy regions and edges of images, rather then in smoother regions. The benefit is that, the degradation in smoother regions of an image is more noticeable to the Human Visual System (HVS), and that these regions are more affected by lossy compression schemes. Taking these aspects into consideration, working in a frequency domain becomes very attractive. The classic and still most popular domain for image processing is that of the Discrete Cosine Transform (DCT). The DCT allows an image to be broken up into different frequency bands, making it much easier to embed information into the middle frequency bands of an image [11]. The middle frequency bands are chosen such that they do not affect the most visual important parts of the image (low frequencies) and without exposing themselves to removal through compression and noise attacks. Another possible domain for information hiding is that of the wavelet domain [12, 13]. The Discrete Wavelet Transform (DWT) decomposes an image into a lower resolution approximation image (LL) as well as horizontal (HL), vertical (LH) and diagonal (HH) detail components. The process can then be repeated to computes multiple scale wavelet decompositions. The embedding can be done on one subband or several subbands. Quantization can be applied to minimize the effect of embedding. One of the many advantages of the wavelet transform is its adaptivity to the HVS as compared to the

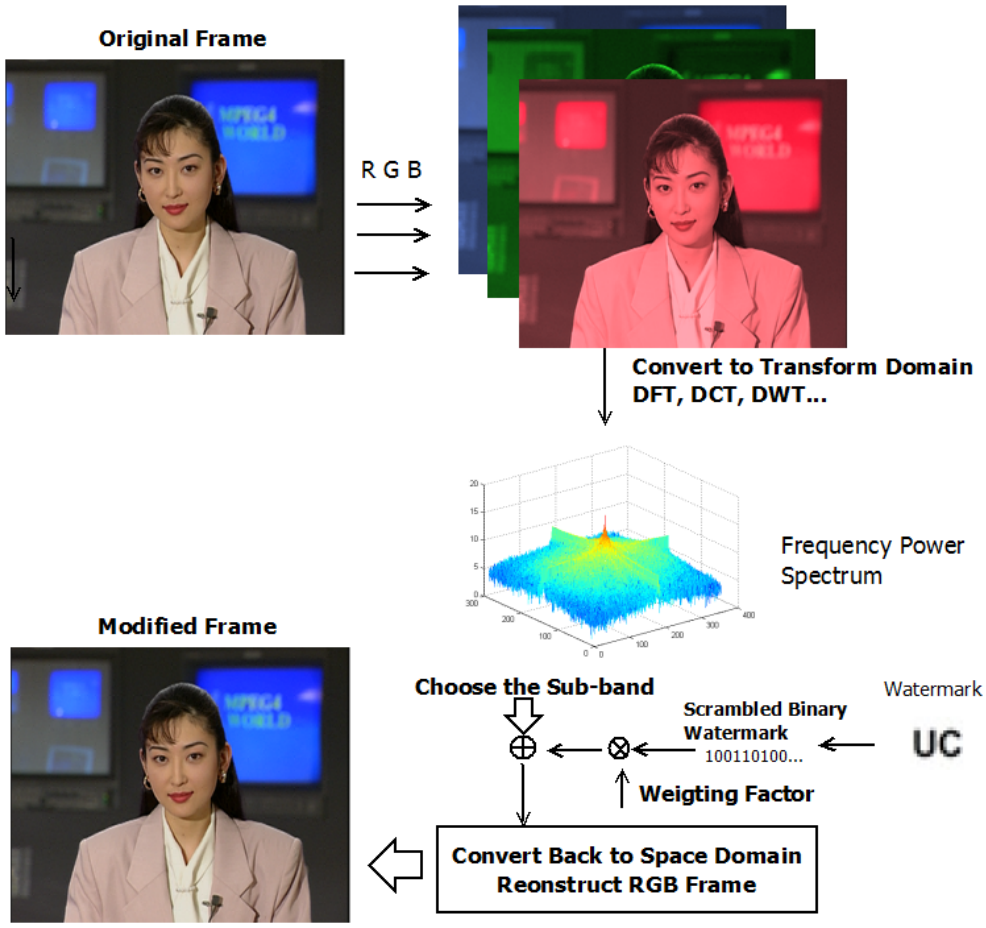


Fig. 1.4: General hiding process using the Transform Domain.

Fast Fourier Transform (FFT) or DCT. This allows us to use higher energy for the embedded information in regions that the HVS is known to be less sensitive to, such as the high resolution detail bands. It has been found that the capacity of the hidden information is related to quality of the cover message; increasing the capacity of embedding will result in degradation in the quality and decreasing in the robustness. In general, doing the embedding process in the frequency domain will result in robustness against many attacks, and at the same time they remain imperceptible to the human sensory system. Fig. 1.4 shows a general hiding process using the Transform Domain.

1.3.3 Statistical Techniques

Statistical techniques are based on the statistical characteristics of digital carriers that are being used as covers for the hidden information [14, 15]. If 1 is transmitted, there will be changes in these characteristics; otherwise the cover is left unchanged. This is shown in Fig. 1.5 where a color image is watermarked using the statistical characteristics of its components. The receiver must be able to distinguish unmodified covers from modified ones. In order to hide the information, the cover message is divided into disjoint blocks. A secret bit is inserted into the corresponding block by placing a 1 into it if the bit is 1; otherwise the block is not changed. The detection of a specific bit is done by using a test function. The test function can be interpreted as a hypothesis-testing function by using the theory of hypothesis-testing from mathematical statistics. Statistical information hiding techniques are difficult to apply in many cases, the distribution of the cover elements should be known. Practical implementations try to determine closed formula for these distributions [16].

1.3.4 Cover Generation Techniques

In cover generation techniques, a digital object is generated only for the purpose of being a cover of secret communications. This is not the case in most of information hiding techniques where the secret information is added to specific covers. By using automated systems, it is possible to check communications by examining keywords and the statistical profile of a message. This makes it possible to distinguish encrypted messages from unencrypted ones. Mimic functions use this fact to hide the identity of a message by changing its statistical profile to match the profile of any innocent text [17]. This depends on that English language characters have non-uniform distribution. Mimic functions can use Huffman coding scheme in this application [18]. The problem

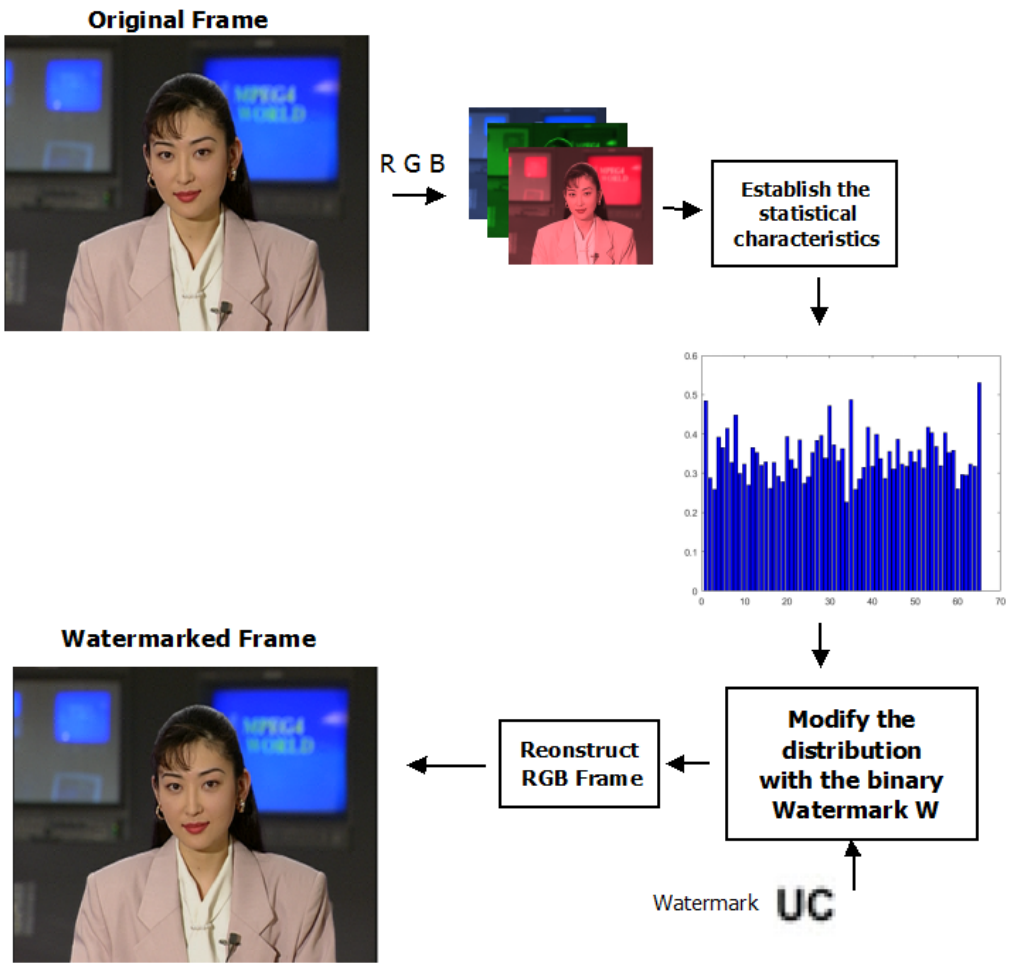


Fig. 1.5: Watermarking a color image using the statistical characteristics of its components.

of mimic functions is that the generated text is full of grammatical and typographical errors. To overcome this problem, automated generation of English text based on context-free grammar has been proposed [19].

1.3.5 Spread Spectrum Techniques

Spread spectrum (SS) communication technologies have been developed since the 1950th to avoid interception and anti-jamming communications. In these techniques, the signal occupies a bandwidth in excess of the minimum necessary to send the

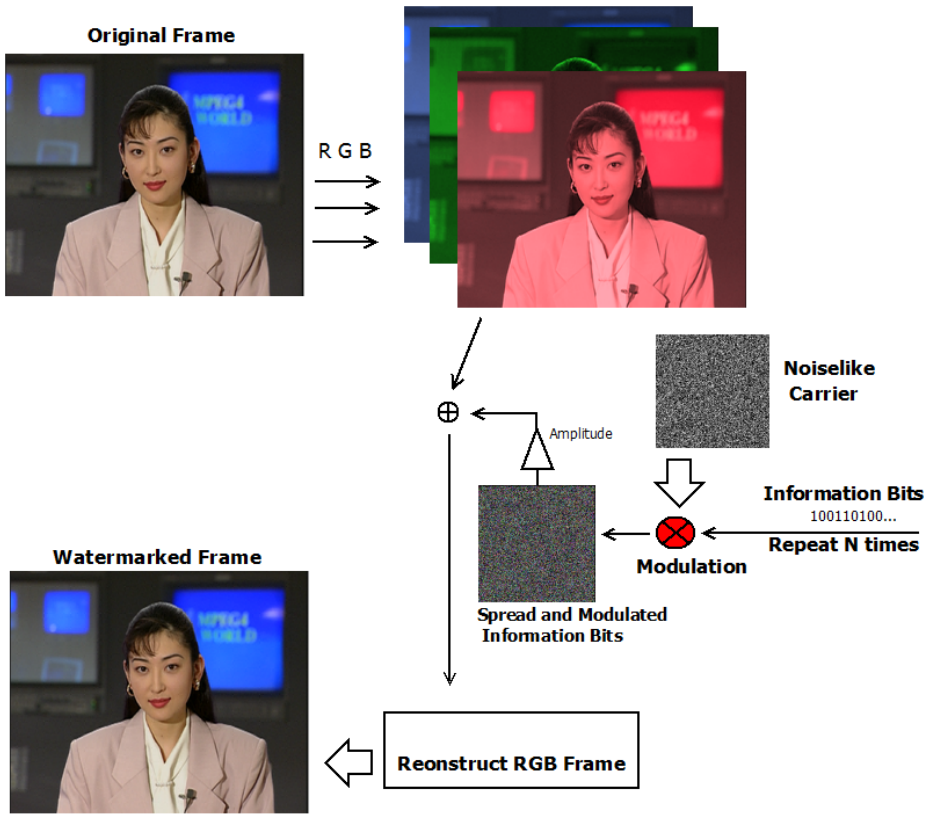


Fig. 1.6: Basic spread spectrum watermarking process for an RGB color image.

information. The band spread is accomplished by a code which is independent of the data, and a synchronized reception with the code at the receiver is used for dispreading and subsequent data recovery [20, 21]. In information hiding, two techniques of SS are used: direct sequence and frequency-hopping schemes. In direct-sequence schemes, the signal is spread by a constant called chip rate, modulated with a pseudorandom signal and added to the cover. On the other hand, in frequency-hopping schemes, the frequency of the carrier signal is altered in a way that it hops rapidly from one frequency to another. Even if parts of the signal are removed in several frequency bands, enough information remains in other bands that enable us to recover the massage. Spread spectrum techniques are robust against many attacks and can be used in watermarking [22, 23]. Fig. 1.6 shows a basic spread spectrum watermarking process for an RGB color image.

1.3.6 Distortion Techniques

Distortion Techniques require the knowledge of the original cover in the decoding process [24]. The sender applies a sequence of modifications to a cover; these modifications correspond to a specific secret message. The receiver, on the other hand, measures the difference to the original cover in order to reconstruct the modifications and finally extracts the hidden message. In order to make these techniques practical, original covers should be distributed through secure channel. Most of text-based hiding methods are of distortion type. For instance, some text-based steganographic schemes use the distance between consecutive lines of text or between consecutive words to transmit secret information [25]. Distortion techniques can be applied to images for embedding information [26].

1.4 Contribution and Organization of the Thesis

Upon designing watermarking systems, many requirements should be met. Visual quality of the host signal, robustness of the techniques, storage requirements, security, and running time of the algorithms are important aspects that should be taken into account in the designing process of information embedding systems. Today, the proposed techniques in the information hiding field try to achieve these requirement or most of them. The aim of this research is to develop some data hiding techniques for multimedia. In chapter 2, basics of the data watermarking techniques is introduced. Some of the factors that affect the design process will be discussed; these factors include distortions and attacks. Requirements of the design process such as imperceptibility, robustness and security will be covered as well. One data hiding technique that uses multi-resolution SVD analysis for video frames will be introduced

in Chapter 3. Moreover, another data hiding technique for 3D Meshes will be introduced in Chapter 4. The performance of our techniques will be tested and compared with other work in the field. The main contribution of the dissertation is that it deals with both 2-D and 3-D data. New techniques of dual data hiding was done in videos watermarking, where color and Black and White (B&W) watermarks were used. Furthermore, optimizations of the 3-D data hiding was achieved and deep analysis of the attacks was performed.

Chapter 2

Requirements for Multimedia Watermarking Applications

2.1 Introduction

Watermarking in multimedia has emerged as a solution of problems of ownership verifications and authentications. For still images, it is used for copyright protection. The hidden information can be of any nature such as a number, a text or an image. The keys can be used to increase security. All practical systems use at least one key, or a combination of keys. The input to any information hiding system is the hidden data, the host signal, and an optional secret or public key. In watermarking systems, the watermark is embedded into a harmless message which is defined as the host signal, generally with the use of a key. The resulting message is the watermarked signal. Ideally the watermarked signal is not distinguished from the original one, appearing as if no other information had been encoded. The extraction of the watermark is the inversion of the hiding process. For ideal detection, the extracted watermark should

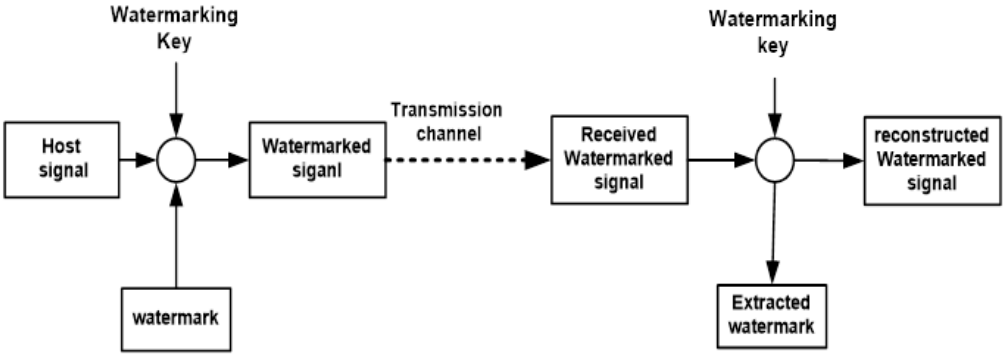


Fig. 2.1: Watermarking system.

be identical to the hidden one [1]. This can be seen in Fig. 2.1

There has been some confusion about the naming of various types of watermarking techniques and the main reason is that people involved in this field come from different backgrounds. Some terminology has been imported from the related field of steganography [27]. Originally, public watermarking and blind watermarking mean the same, but the wording was confusing with public-key watermarking, so that only the later tends to remain. In these schemes, the cover signal is not needed during the detection process to detect the watermark. Solely, the key which is typically used to generate random sequences during the embedding process, is required. These types of schemes can be used easily in mass market electronic equipment or software. In some cases, extra information is needed to help the detector, in particular to synchronize its random sequence on the possibly distorted test signal. In particular, some watermarking schemes require access to the 'published' watermarked signal, which is the original signal just after adding the watermark. People refer to these schemes as semi-blind watermarking schemes. Private watermarking and non-blind-watermarking mean the same. The original cover signal is required during the detection process. By asymmetric watermarking or public-key watermarking, people refer to watermarking schemes with properties reminding asymmetric cryptosystem (or public key cryptosystem)

[28]. No such system really exists yet although some possible suggestions have been made. In this case, the detection process (and in particular the detection key) is fully known to anyone as opposed to blind watermarking where a secret key is required. So here, only a 'public key' is needed for verification and a 'private key' (secret) is used for the embedding process. The watermarks, on the other hand, can be fragile or robust. A fragile watermark is a mark which is highly sensitive to a modification of the watermarked signal. A fragile watermarking scheme should be able to detect any change in the signal and identify where it has taken place and possibly what the signal was before modification. It serves at proving the authenticity of a document. On the opposite, a robust watermark should be stuck to the document in which it has been embedded, in such a way that any signal transform of reasonable strength cannot remove the watermark. Hence, anyone willing to remove the watermark will not succeed unless he affects the document too much to be of low commercial interest. The latter form is the very challenging and attracts most research.

2.2 Requirements for Multimedia Data Hiding Applications

Depending on the watermarking application, different requirements arise with various design issues. The most important issues that are addressed are the imperceptibility and the robustness.

2.2.1 Imperceptibility

The hidden data should not add any artifacts that may degrade the quality of the host message. It is important to design the data hiding system in such a way that

exploits effects of human visual or auditory system to maximize the energy of the hidden data without affecting the perceptual transparency. For instance, in digital images watermarking systems, the algorithm must modify the bits of the cover in such a way that the statistics of the image are not modified in any fashion that may confuse the presence of a watermark. This requirement is not quite as important here as it is in steganography, but some applications may require it. Evaluating the imperceptibility of the watermarking system can be done either through subjective tests or quality metrics. When using a subjective test, a testing protocol has to be followed, describing the testing and evaluation procedure. The distorted data sets are ranked from best to worst, and then they are rated based on the perceptual transparency of the watermark. ITU-T Rec. P.910 describes the subjective video quality assessment methods for multimedia applications. There are other methods to measure the visual quality of images and videos. The structural similarity (SSIM) is used to measure the quality of images and videos; it has the property of being well correlated with the HVS [29]. SSIM is designed to improve on traditional methods such as peak signal-to-noise ratio (PSNR) and mean squared error (MSE), which have proven to be inconsistent with human visual perception. For 3-D meshes, MSDM and MSDM2 (Mesh Structural Distortion Measure) and MRMS (Maximum Root Mean Square Error) are used. The human visual perception are well integrated in the MSDM and MSDM2 measurements [30, 31].

Traditional quantitative distortion metrics are still being used, since they are relatively easy to compute, and they are able to give some indication on the quality. Peak signal-to-noise ratio (PSNR) is the standard method for quantitatively comparing a distorted image with the original. For an 8-bit grayscale image, the peak signal value is 255. Hence, the PSNR of an 8-bit grayscale image x and its watermarked version

y is calculated as [32]:

$$PSNR = 10log_{10}XY\frac{255^2}{\sum_{x,y}(p_{x,y} - \hat{p}_{x,y})^2} \tag{2.1}$$

Where X and Y are the dimensions of the original and watermarked images, $p_{x,y}$ is the intensity of the pixel at location (x, y) of the original image, and $\hat{p}_{x,y}$ is the intensity of the pixel at location (x, y) of the watermarked image. PSNR is measured in decibels (dB).

2.2.2 Robustness

The designed watermarking system should resist any kind of distortions introduced by standard or malicious data processing. In most watermarking applications, the marked data is likely to be processed in some way before it reaches the watermark receiver. The processing could be lossy compression, signal enhancement, or digital-to-analog (D/A) and analog-to-digital (A/D) conversions. An embedded watermark may unintentionally or intentionally be impaired by such processing. Other types of processing may be applied with the explicit goal of disturbing the watermark reception. In watermarking terminology, an attack is any processing that may weaken detection of the watermark or communication of the information conveyed by the watermark. The processed watermarked data is then called attacked data. An important aspect of any watermarking scheme is its robustness against attacks. A watermark is robust if it can not be destroyed without also severely degrading the quality of the attacked data. Watermark impairment can be measured by criteria such as miss probability, probability of bit error, or channel capacity. For multimedia, the usefulness

of the attacked data can be measured by considering its perceptual quality or distortion. Hence, robustness can be evaluated by simultaneously considering watermark impairment and the distortion of the attacked data. An attack succeeds in defeating a watermarking scheme if it impairs the watermark beyond acceptable limits while maintaining the perceptual quality of the attacked data [33].

2.2.2.1 Robustness Factors

When designing a Multimedia data hiding system, many aspects should be taken into account to achieve the robustness condition. This depends naturally on the type of the Multimedia, the transmission channels, and the different processes that are being performed. In image watermarking, for example, filtering or compression may destroy the watermark in space domain such as LSB substitution technique, but, the watermark can survive these processes in transform domains. On the other hand, geometric attacks have great artifacts on transform domain watermarking systems, but, space domain watermarking systems can survive these attacks. Some factors that influence the robustness of watermarking schemes are [28]:

- **Amount of Hidden Data**:

  There is a limitation on the amount of information that can be hidden depending on the methods and the applications. In general, the more information the one wants to hide, the lower the robustness of hiding. In watermarking systems, the hidden information can be a serial number, a text, or a small logo which does not require big space.

- **Watermarking Strength**:

  When embedding information, it is required to have the ability to extract them with minimum errors. This should be accompanied by the lowest contribution to

the imperceptibility; these two conditions should be achieved for any embedding system.

- **Secret Information**:

  The nature of secret information, in general, does not affect the robustness or the perceptual transparency of the embedding process, but they play important role in the security of the system. Secret information is the key of the embedding process; number of keys used should be large enough that it becomes impossible to know them in a reasonable period of time. Any designer of information hiding system should take into account that resisting attacks is strongly related to the security of the keys.

  Taking these parameters into account, the information hiding system can be designed and tested. The size of the hidden watermark, the quality of the extracted watermark, and the quality of the cover message are important aspects that should be taken into consideration. Other measurements of robustness include the correlation distortion metric, which indicates the similarity between the extracted watermark and the embedded one. Another, more indicative parameter that is increasingly being used is the bit error rate (BER) between these two watermarks. Zero BER means perfect extraction.

2.2.2.2 Watermarking Systems Attacks

When designing digital watermarking methods, an important issue addresses evaluation and benchmarking. In general, there is a trade-off between robustness and imperceptibility. Hence, for fair performance evaluation, one has to ensure that the methods under investigation are tested and evaluated taking into account the above aspects and under the same attacks. A block diagram of the attack process is shown

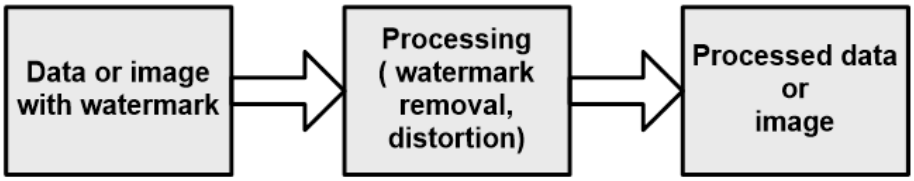


Fig. 2.2: Watermarking attacking process.

in Fig. 2.2.

Some attacks that may disturb the information embedding systems are listed below:

- **Additive Noise**

  Additive noise can be added to the cover message to disturb the hiding process. This noise normally degrades the host data and tries to destroy the hidden message. Robust embedding techniques should survive this attack. Examples of additive noise are white Gaussian and salt-and-pepper [34].

- **Linear Filtering Noise**

  Linear filtering is used in images to sharpen or soften their appearances. This type of attacks has large artifacts when using space domain information hiding systems. It is preferred to use transform domains in embedding to survive this attack. Examples of this distortion are low-pass and high-pass filtering [1].

- **Non-linear Filtering**

  Nonlinear filtering is used to enhance the filtered data, especially in images. In image watermarking, for example, it can destroy the watermark without affecting the visual quality to a great extent. Example of this kind of attacks is the median filtering [35].

- **Local and Global Transforms**

  Some transforms are applied to multimedia, such as images, to change the geometrical characteristics of them. The purpose of that is to enhance the visual appearance. Examples of these transforms include scaling, rotation and translation. These transforms can be used by attackers to destroy the embedded data [36]. Robustness to this kind of attacks is important especially in transform domain techniques. In contrast to removal attacks, geometric attacks do not actually remove the embedded watermark itself, but intend to distort the watermark detector synchronization with the embedded information [37]. The detector can recover the embedded watermark information when perfect synchronization is regained. However, the complexity of the required synchronization process might be too big to be practical. Alignments of images and videos are generally used to counteract the effect of these transforms and regain the detection sensitivity.

- **Data Reduction**

  Images can be reduced by cropping or histogram equalization depending on the requirement of users [38]. Other ways of data reduction include frames dropping in videos. In 3D meshes, reduction can be done using mesh simplification process. Information hiding techniques should have the ability to detect these reductions and survive them.

- **Lossy Compression**

  Many compression techniques for multimedia are currently available. Compression is used to reduce the transmission and the storage requirements of the data. Some standards that are being used:

  o For still images JPEG, JPEG2000, PGF.

  o For videos H.264/MPEG-4 AVC, H.265/HEVC.

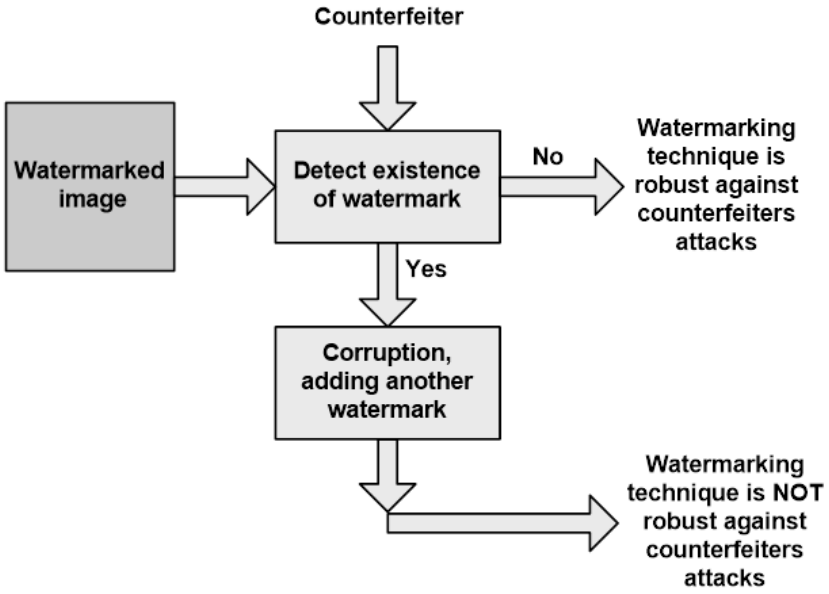


Fig. 2.3: Counterfeiter attack.

o For Audio MP3, AAC, Vorbis, FLAC,

- **Counterfeiting Attacks**

  Most of data hiding techniques store the data in a manner or a location which is kept secret. Any attacker who tries to destroy the hidden data tries to look for the location, or put unreasonable amount of data to the host message [39, 40]. In watermarking systems that are used for ownership verification, the attacker can add his own watermark and claim the ownership of the marked content; this can be seen in Fig. 2.3. Robust watermarking techniques should survive these multiple watermarking attacks, and the watermark should be detected with reasonable quality. These kinds of attacks are normally called protocol attacks, and they are considered when robustness is required.

2.2.3 Computational Complexity

Among the most important designing issues that are being addressed in any embedding system, are the total time and the number of operations being performed. This, of course, depends on the techniques and the applications of the information hiding system. Computational complexity, in this regard, gives us information of how much of a resource (such as time, space, parallelism, or randomness) is required to perform some of the computations that interest us the most. The most basic resource studied in computational complexity is running time which depends, of course, on the number of basic steps taken by an algorithm. Other resources, such as space (i.e., memory usage) are affected directly by this complexity. When the input is given as a string of 0's and 1's, typically, the running time is measured as a function of the input length and the number of operations being performed. To design an optimum information hiding system, a trade-off between perceptual transparency, robustness, and computational complexity should be achieved.

Chapter 3

Hybrid Pyramid-DWT-SVD Dual Data Hiding Technique for Videos Ownership Protection

3.1 Introduction

The delivery and distribution of various types of multimedia became easier as the new advances in digital communications and information networks are reaching new limits and capabilities every year. These forms of digital information can be easily stolen and exploited when the appropriate precautions are not put in place. These concerns motivated significant research in image and video watermarking fields [41]. Watermarking is a type of data hiding and it is used primarily for authentication and ownership protection. New innovations in video processing techniques, such as compression and transcoding techniques, has brought new challenges to watermarking. For instance, High efficiency video coding (HEVC) or H.265 standard was introduced

officially in 2013, it needs on average only half the bit rate of its predecessor, ITU-T H.264 — MPEG-4 Part 10 'Advanced Video Coding' (AVC), which was considered the most deployed video compression standard worldwide [42]. The new standard is expected to be phased in gradually as the new display technologies and networks capabilities outgrow their current limits [43].

Various watermarking schemes that use different techniques have been proposed over the years [44, 45, 46, 13, 47, 48, 49, 50]. To be effective, a watermark must be imperceptible within its host, extracted with ease by the owner, and robust in the face of both intentional and unintentional distortions [13, 51, 52]. Multi-resolution decompositions of images are very popular in the areas of images and videos codings, specifically compression and data hiding [9, 53]. The pyramid scheme, for instance, which was introduced by Burt and Adelson [54] is a form of multi-resolution analysis and proved to be very useful in images compression, it can use linear and nonlinear interpolation and decimation operators. On the other hand, Discrete Wavelet Transform (DWT) is another multi-resolution process; it has wide applications in the different areas of image and video processing such as compression, noise reduction and watermarking [55, 56]; this is attributed to its characteristics in : space-frequency localization, multi-resolution representation and superior Human Visual System (HVS) modeling [45]. Furthermore, the Singular Value Decomposition (SVD) is a powerful technique in many matrices computations. It has the advantage of being more robust to numerical errors [49]; this property of SVD analysis besides others made it useful in image and video watermarking [57, 58]. To have a robust watermarking technique, both the hiding and the extraction processes should be optimized. This in turn necessitates the need for enhanced detection process with no significant extra cost in terms of visual quality or computational complexity.

In this research, our target is to develop a dual watermarking technique using mainly

a hybrid pyramid-DWT-SVD analysis. The overall technique will be used for data hiding in encoded videos to meet the requirements of imperceptibility, robustness, security, and computational complexity. The hiding process will be composed of two stages for security reasons; furthermore, a new algorithm will be derived to enhance the extraction and detection processes. A great deal of randomness will be used in many aspects including but not limited to: the filter banks generation and the hiding process to ensure high level of security. The overall performance of the proposed technique will be measured when common aggressive attacks are applied to the test videos. Moreover, special robustness tests against H.264 nad H.265 compressions and trasnscoding are performed to illustrate the performance of our system against these attacks.

3.2 Proposed Watermarking Technique

3.2.1 Pyramid, DWT and SVD Processes

Multi-resolution analysis is very important in images and videos processing. They enable flexible processing of the input data with lower computational complexity; this is the reason that they are used in image compression for instance. There are many techniques that rely on the multi-resolution analysis such as Pyramid Transform and DWT; a theoretical background is introduced here.

If $x_0(n_1, n_2)$ is the original image of size $L1 * L2$ pixels, then the pyramid structure can be done as shown in Fig. 3.1 [59]. For decimation by a factor of 2, the image will be filtered using analysis lowpass filter H, and then it will be decimated by a factor of two. This results in an image $x_1(n_1, n_2)$ which is 1/4 of the size of $x_0(n_1, n_2)$ and it is called the first-level image of the pyramid. The second level image $x_2(n_1, n_2)$

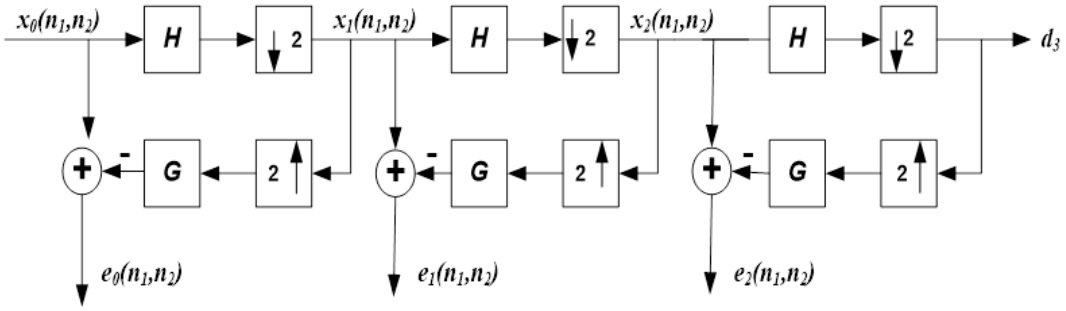


Fig. 3.1: Three-level pyramid decomposition of an image $x_0(n_1, n_2)$.

can be obtained from $x_1(n_1, n_2)$ by the same process, and this process is repeated for the higher levels. The image $x_1(n_1, n_2)$ can be interpolated by a factor of 2 and then filtered using synthesis filter G. The resulting image will be $I[x_1(n_1, n_2)]$. Where $I[.]$ is the spatial interpolation and filtering operation. The synthesis filter G is a time reversal version of the analysis filter H. The difference (error image) $e_0(n_1, n_2)$ is given by:

$$e_0(n_1, n_2) = x_0(n_1, n_2) - I[x_1(n_1, n_2)] \tag{3.1}$$

This process can be done for the higher levels and we will have the error images $e_1(n_1, n_2)$, $e_2(n_1, n_2)$ ... etc. The optimization of the analysis and synthesis filters plays the major role in the perfect reconstruction of the images. For watermarking purposes, random filters will be used. Quadrature Mirror Filter (QMF9) is an example of optimum filters [60]; it satisfies symmetry, normalization, unimodality and equal contribution. The frequency response of this filter is shown in Fig. 3.2.

The error image that results from this process will be used for our DWT hiding process. The reason for this pre-DWT process is that the resulting error image has

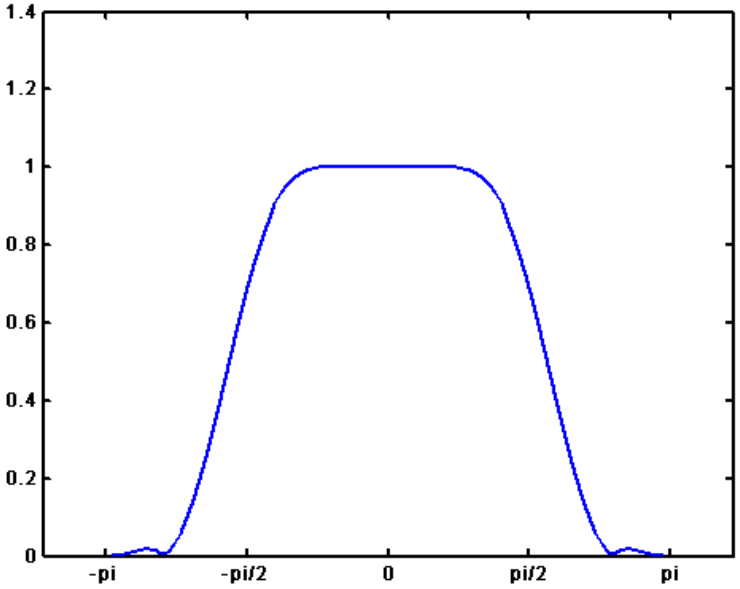


Fig. 3.2: Frequency response of QMF9 filter.

a broader frequency spectrum than the original image, and hence this would give more space in the frequency domain for hiding our watermark. This, in turn, will contribute less visual artifacts to the host image which is important to achieve perceptual imperceptibility; this can be shown in Fig. 3.3. To give a whole picture of the pyramid decomposition process; the Y-frame of Akiyo was decomposed using pyramid decomposition for three levels; this time another filter was used, this would result in three error bands e_0, e_1, e_2 and a decimated image d_3 [61]. The frequency responses of these bands are shown in Fig 3.4. It can be seen that the frequency response of the decimated version is similar to the frequency response of the original image, where most of the power is concentrated in the lower frequency band. On the other hand, the spectrums of the error images show more flat distribution of the power all over the frequency bands. Moreover, changing the pyramid decomposition filter changed the shape of the spectrums of the bands; this is expected since each filter has its own frequency response. Furthermore the histograms of the errors and decimated bands

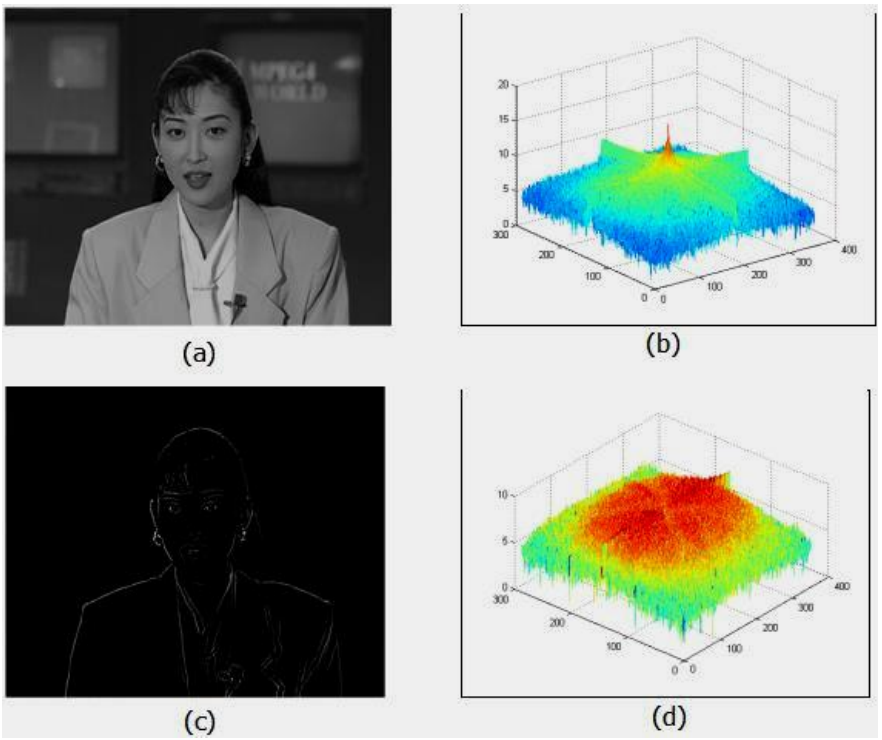


Fig. 3.3: (a) the original image (b) the frequency spectrum of the original image (c) the error image 1 (d) the frequency spectrum of the error image 1

and the original frame are shown in Fig. 3.5; It can be seen that the decimated image and the original image have similar histograms; the error images pixel values are concentrated around zero. These characteristics of pyramid decomposition made it useful in compression. Minimal distortion to the error images like in the case of the watermarking process can be tolerated more than the distortions in the original or decimated images.

The wavelet transform on the other hand, has the advantage of achieving both spatial and frequency localizations. Wavelet decomposition depends mainly on filter banks, typically the wavelet decomposition and reconstruction structures consist of filtering, decimation, and interpolation. Fig. 3.6 shows two-channel wavelet structure for one-dimension signal; where H_0, H_1, G_0, and G_1 are the low decomposition, high decomposition, low reconstruction and high reconstruction filters respectively. For the perfect reconstruction (i.e. x_0=$\hat{x_0}$), these filters are related to each other according

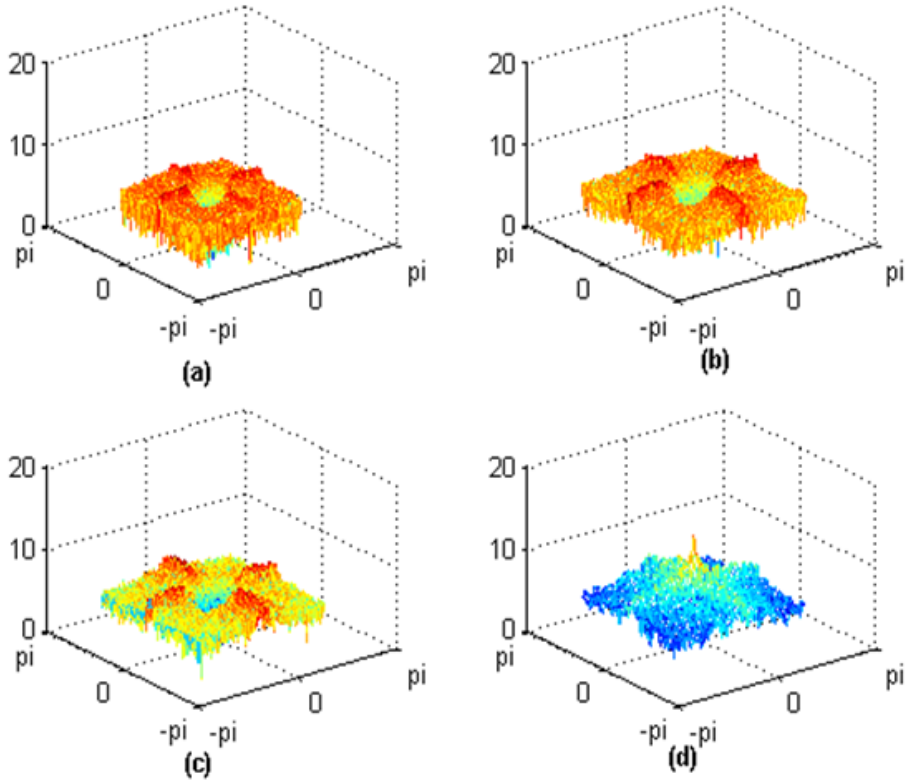


Fig. 3.4: Frequency responses of the pyramidal components of Akiyo frame: (a) e_0 image, (b) e_1 image, (c) e_2 image, (d) decimated image d_3.

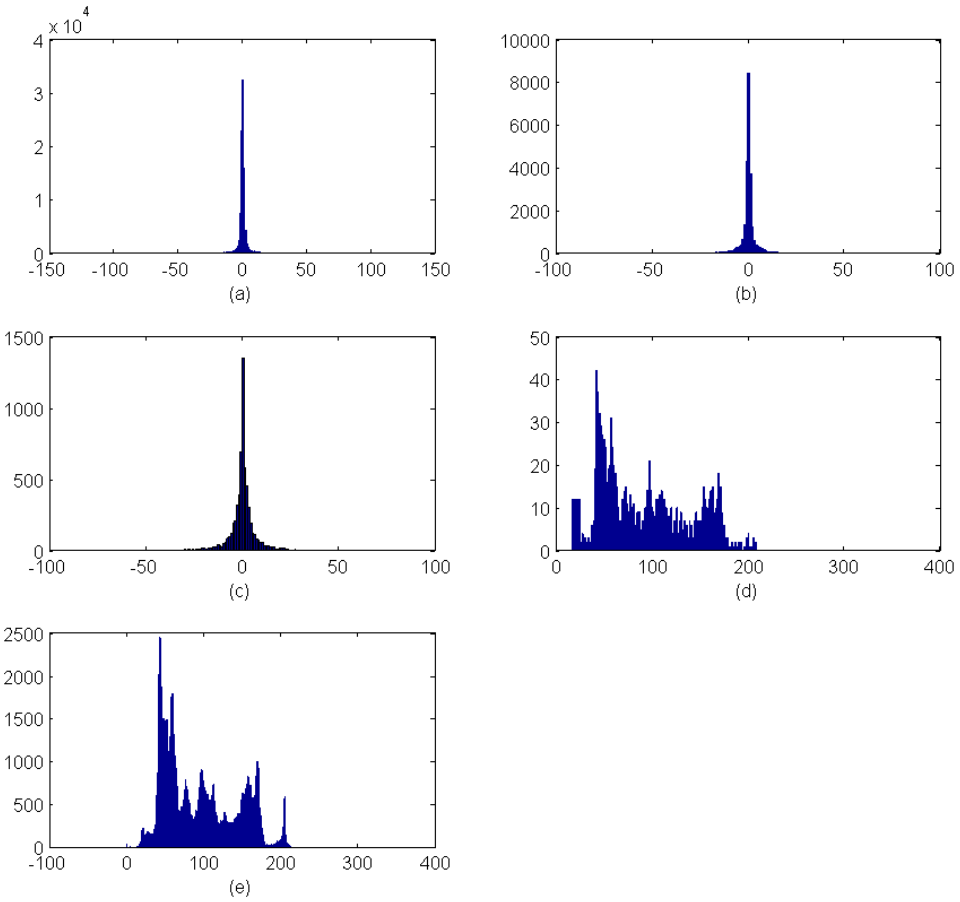


Fig. 3.5: Histograms in space domain of Akiyo image and its Pyramidal decomposed images (a) e_0 image, (b) e_1 image, (c) e_2 image, (d) decimated image d_3, (e) original image

.

to the relations given below in Equations 3.2 and 3.3 [61]:

$$H_0(z)G_0(z) + H_1(z)G_1(z) = 2 \tag{3.2}$$

$$H_0(-z)G_0(z) + H_1(-z)G_1(z) = 0 \tag{3.3}$$

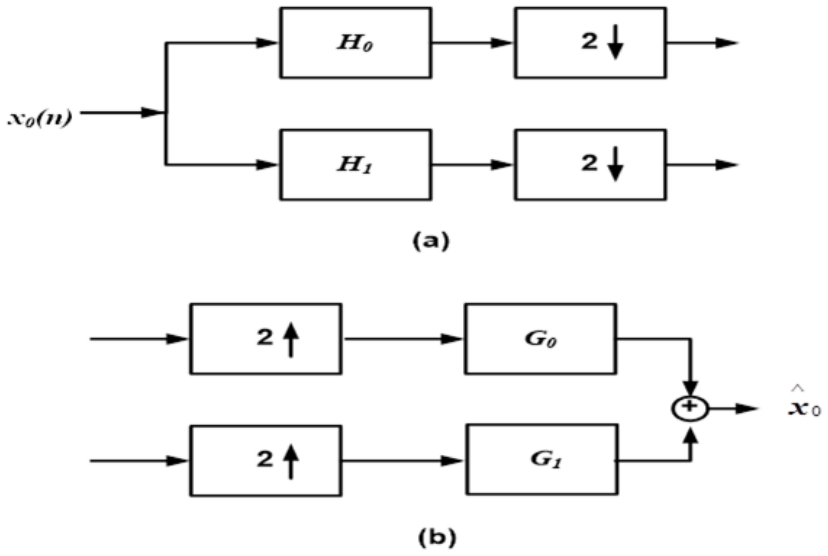


Fig. 3.6: Two-channel wavelet transform structure: (a) decomposition, (b) reconstruction.

The shape of the generated filters is controlled by the generating polynomials , so that, they can be wide-band or narrow-band, with large or small sidelobes. Fig. 3.7 shows the analysis and synthesis filters that were generated depending on the generating polynomials; these filters are orthonormal filters that have large sidelobes.

The two dimensional decomposition can be extended to images by performing the filtering horizontally and vertically. This means that the filtering is done on rows and

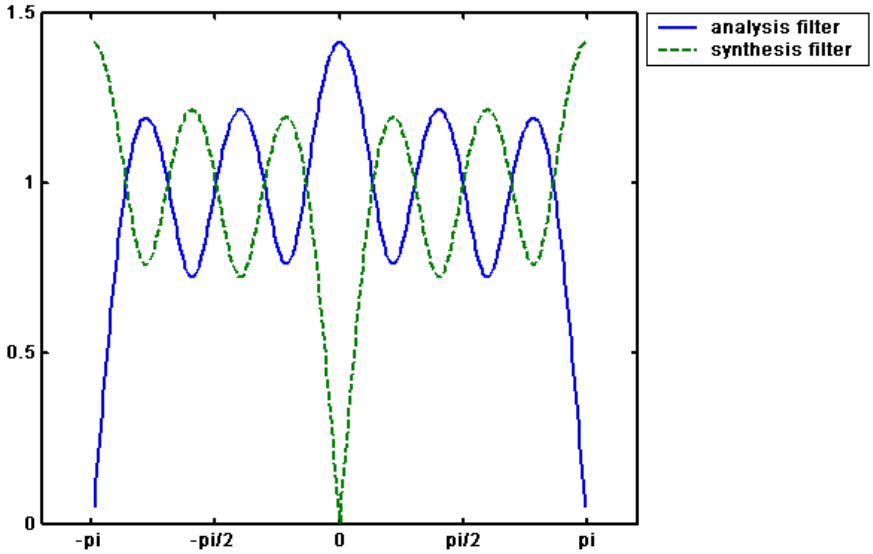


Fig. 3.7: Wavelet analysis and synthesis filters.

columns using the analysis filters in the decomposition stage, and the synthesis filters in the reconstruction stage as shown in Fig. 3.8.

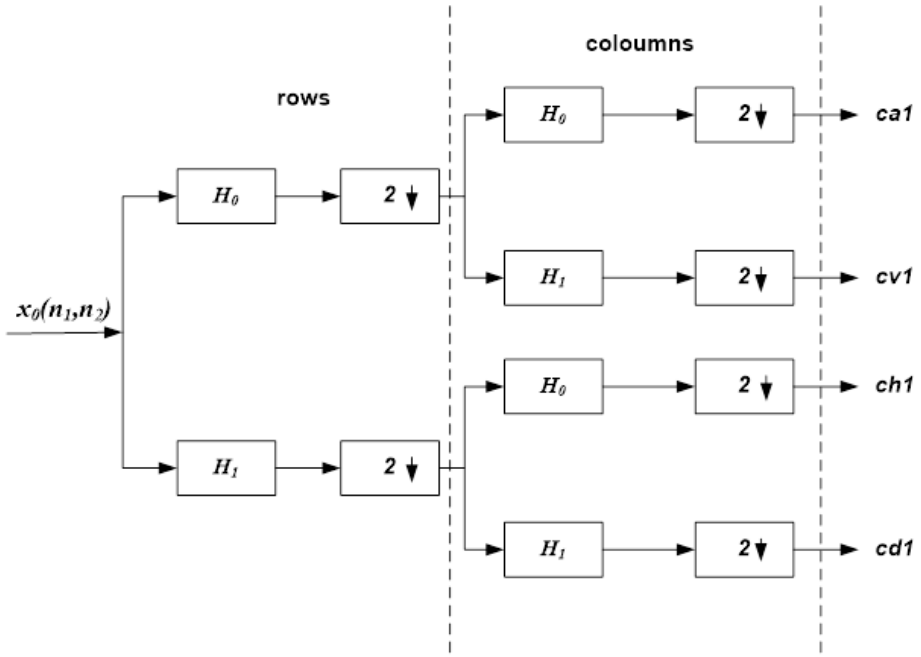


Fig. 3.8: One-level wavelet decomposition of two-dimensional signal.

Where:

- *ca*: lowpass-lowpass filtered subband,
- *cv*: lowpass-highpass filtered subband,
- *ch*: highpass-lowpass filtered subband,
- *cd*: highpass-highpass filtered subband.

The numbers accompanied with these subbands denote the level of decomposition. For multi-level decomposition, this process should be done successively; moreover, the decomposition can be done in a variety of structures. A possible decomposition structure for five levels is shown in Fig. 3.9. Furthermore, a 2-level DWT decomposition of UCI logo image is shown in Fig. 3.10. It can be seen that the high frequency bands preserve the edges and textures, while smooth areas are lost in these bands. On the other hand, the low frequency band ca_2 has most of the information of the original image.

In fact, the DWT orthonormal analysis and synthesis filters can be constructed in such a way that they have large sidelobes. This allows higher energy in the medium frequency bands of the spectrum of the images, to avoid as much as possible the effects of different images' processing techniques that are applied at one stage or another. Depending on the number of the decomposition levels, each filter bank can be used for one level of the DWT decomposition and reconstruction. Furthermore, full control on both the structure and the number of levels of the decomposition process can be established to address the security concerns. The bands in the middle frequencies will be used for hiding in general, this in turn would avoid the use of the lower frequency bands, where most of the energy is in, and the higher frequency bands which are susceptible to compression and other image processing attacks such as low-

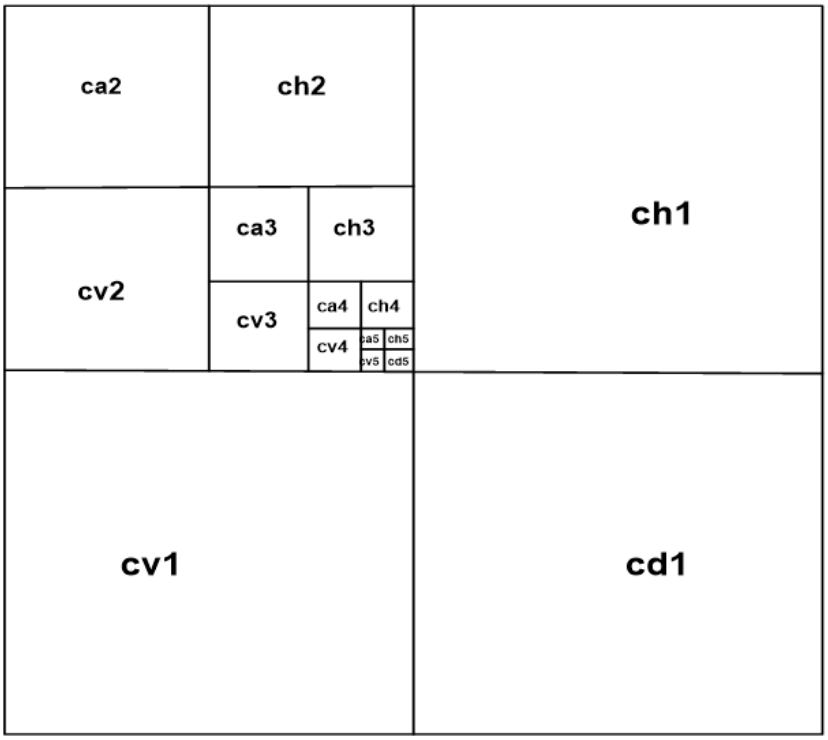


Fig. 3.9: Five-level wavelet decomposing structure.

pass filtering. These facts are illustrated in Fig. 3.11 which shows the spectrums of the 1st level DWT bands of the Y-frame of Akiyo video. The watermark might be distributed across many subbands. This scenario is helpful in counteracting the Non-linear Collusion Attack. However, to find the best band to hide in, the *directive contrast* will be used as will be shown later in this research.

Fig. 3.10: A 2-level DWT decomposition of UCI logo image.

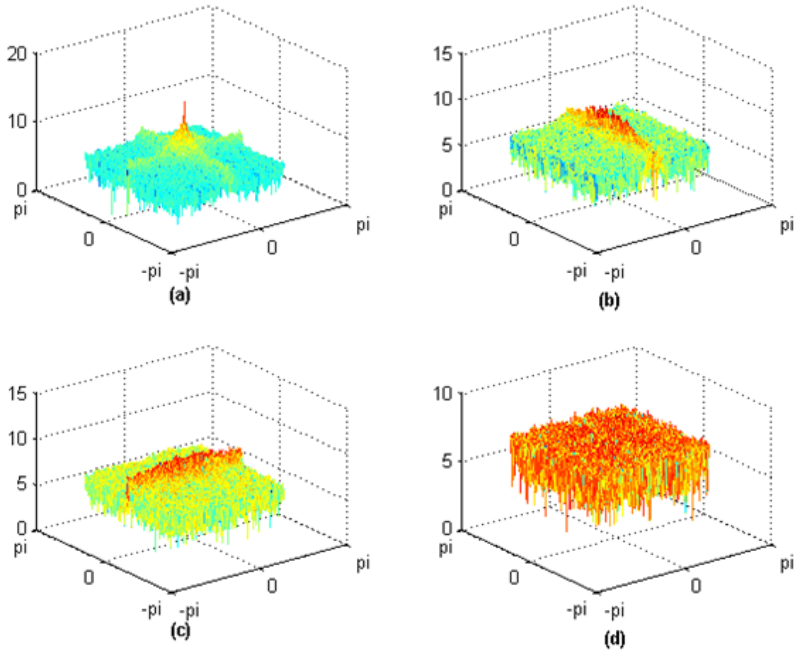


Fig. 3.11: Spectrums of the 1st level DWT bands of a Y-frame of Akiyo video.

Singular Value Decomposition (SVD) was used extensively in images and video watermarking; the image A can be decomposed according to this relation:

[U S V]=SVD(A) where:

$$A = U * S * V^T \tag{3.4}$$

U and V are orthogonal matrices, while S is a diagonal matrix containing the singular values; these singular values are distributed in a descending order. Like the Eigenvalues analysis, the highest singular values comprise the greatest amount of information in the decomposed matrix; this is the reason that SVD can be used in the images' compression processes. Due to numerous image processing attacks, especially aggressive compression, these singular values could change dramatically, which would affect the reconstructed matrix. Most watermarking techniques that use the singular values decomposition depend on hiding the singular values of the watermark in the host image; these methods, however, usually require the original image during the extraction process, hence these methods are semi-blind methods [49, 57, 62, 63]. Our proposed method does not require the original images or video frames; the watermark will be hidden in the host image using the pyramid-wavelet hiding process and then in the extraction process, an approximated watermark will be estimated, and by performing other processing and doing singular value decomposition, optimal singular values can be established that have the minimum root-mean-square error to the original singular values. These established singular values can be used in establishing the reconstructed watermark.

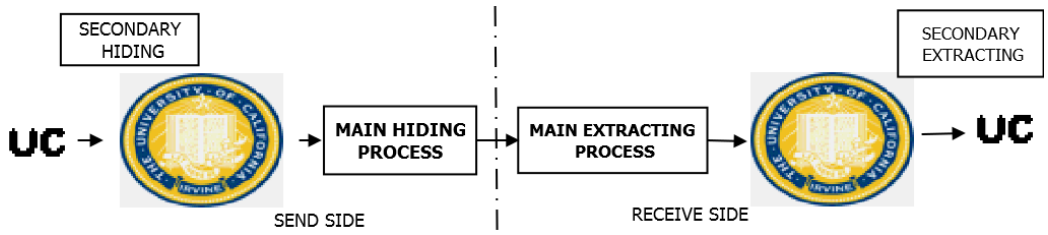


Fig. 3.12: A general illustration of our dual hiding process

3.2.2 Embedding Method

In this subsection, we introduce our digital video watermarking technique for the purpose of authentication and ownership protection. The proposed technique is aimed at achieving reasonable degrees of robustness, imperceptibility and security. The embedding technique consists of two stages: the first stage is the decomposition process and the second stage is the hiding process. The hiding process is a dual one. The main watermark that will be used in the videos is a color RGB image; moreover, another smaller B&W watermark will be hidden in this color watermark. The reason for this duality is to establish a high degree of security; furthermore, this would help in generating a built-in random spread spectrum sequence that will be used in the main hiding process. A general illustration of this dual hiding process is shown in Fig. 3.12.

The watermark can be a logo color image of size N*M*3 pixels. The encoded videos are primarily in the YUV color space; this space is more efficient in representing the images than the traditional RGB space. The watermarking process can take place in any of the three components Y, U or V. Our proposed algorithm will use the luminance Y frames as host images for the multi-resolution watermarking process; that is, the watermark will be inserted or distributed in one or more of the subbands that result from the hybrid pyramid-wavelet decomposition process. Choosing the analysis and

synthesis filters is an important aspect in the efficiency of the reconstruction process. For the wavelet decomposition, special type of filters known as the orthonormal filter banks will be used [61]. These filter banks can be generated randomly depending on the generating polynomials; hence, by generating random numbers for the polynomial coefficients, it's possible to build multiple filter banks that are used for the different stages of our decomposition processes. The filters for the pyramid transform are unconstrained, typically, zero-phase FIR filters are used [61].

The watermark used in our technique is an RGB color image; the three components: the Red, the Green and the Blue will be extracted, then Bit-Slicing process will be performed on each component, i.e, the slices corresponding to the bits from the least significant to the most significant will be established. Since the most significant bits (from 5 to 8) contain most of the information, these bit-slices will be used only [49]. This is illustrated in Fig. 3.13; these individual binary slices will be used as separate watermarks in the hiding process, and then, they will be reconstructed afterwards.

A method to embed the binary watermark using pseudo-random sequence is proposed in [64]. This method establishes the watermarking embedding process by converting the original watermark image Q to a binary sequence S of length M where the data pixels are valued as +1 and the background pixels are valued as -1. Moreover, a pseudo-random sequence P that has the same length M as the watermark sequence is generated using a secret key and, similarly, is represented as binary bits that are valued either +1 or -1. The DWT coefficients of the subbands which will be used for the embedding process can be represented as a matrix Q_1 of the same size as the watermark, and it can be converted to a vector T of length M. The watermark is embedded into the vector T to obtain a new vector T' according to the following

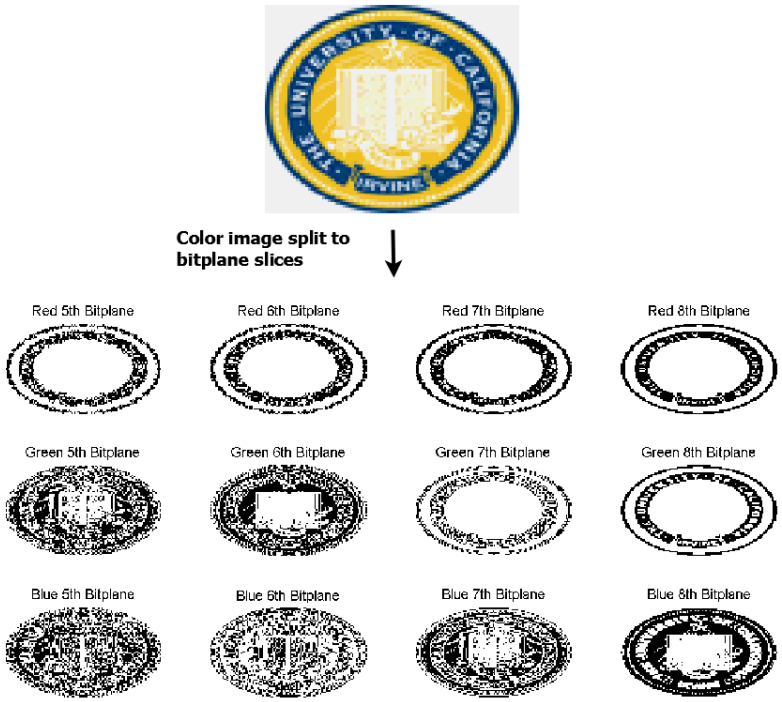


Fig. 3.13: Color image and its most significant bit-slices.

rule:

$$t'_i = t_i + \alpha * p_i * s_i, \;\; for \;\; i = 1, 2...M \tag{3.5}$$

Where α is a magnitude factor which is a weighting constant that controls the strength of the processed watermark. The value is selected to offer a trade-off between robustness and visual quality. Furthermore, choosing the weighting factor should take into account many issues such as the compression ratio, the smoothness of the image, and the detection process. Moreover, the energy content in the wavelet subbands should be taken into account; one way to get the magnitude factor is to compare the original coefficients of the host DWT subband Q_1 and that of the original watermark image

Q according to this empirical formula:

$$\alpha = 2 * \sqrt{\frac{E(Q_1)}{E(Q)}} \tag{3.6}$$

Where *E(Q_1)* denotes the energy of the original wavelet coefficients, while *E(Q)* denotes the energy of the watermark matrix Q which are the sum of the square elements. The enhanced wavelet coefficients are used then in their respective places to reconstruct the watermarked frame. The overall hiding process with color watermarks is shown in Fig. 3.14. It can be shown, and this depends on the decomposition structure, that the Low-Low frequency version (LL) of the decomposed image is avoided since this sub-band contains most of the information of the original image; the other images represent the Low-High (LH), High-Low (HL) and High-High (HH) bands, and they can be used for hiding.

3.2.3 The Directive Contrast

To choose the best band for hiding, *directive contrast* can be used [63]. The various directive contrasts for any DWT decomposition level i are defined as:

- Horizontal Contrast: $C_i^H = LH_i/LL_i$
- Vertical Contrast: $C_i^V = HL_i/LL_i$
- Diagonal Contrast: $C_i^D = HH_i/LL_i$

Directive contrast depicts the high frequency information of an image and the relative intensity of high frequency to the background. To choose the best band to hide in,

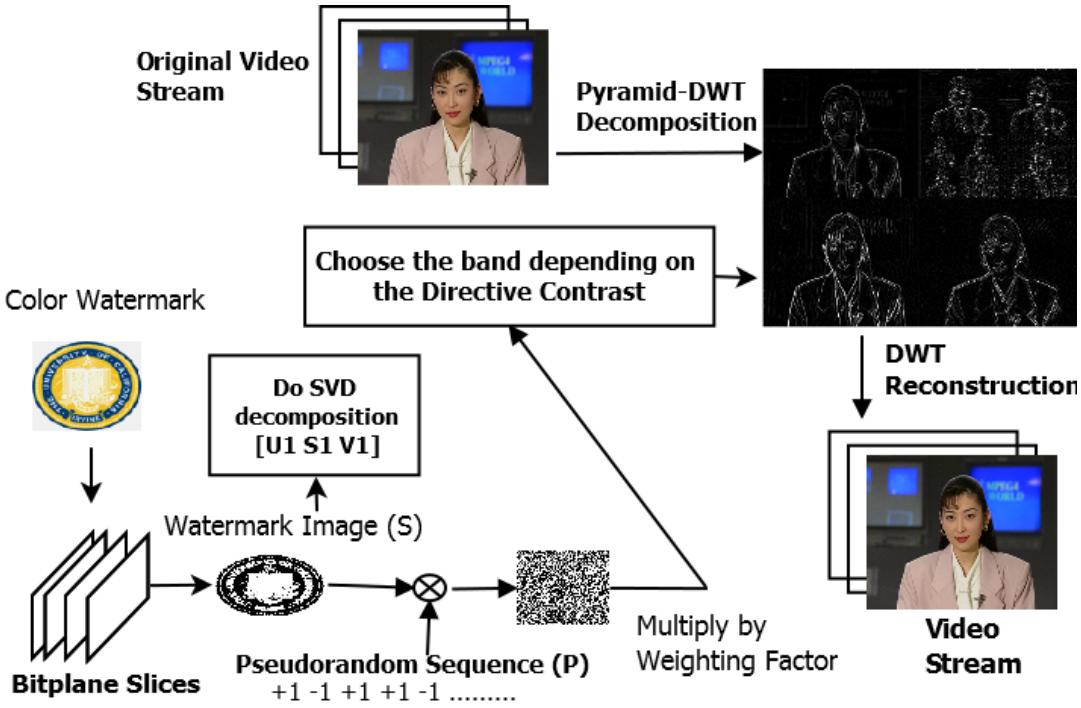


Fig. 3.14: The main color watermark hiding process

the band with the highest directive contrast can provide the highest capacity. One way to compare is to use the norm of matrices; this can be shown in Fig. 3.15 where one level of DWT was performed on the Akiyo video; It can be shown that *HL* band (the left bottom one) is the band with the highest contrast, and so it is the best band for hiding; experimental results are accordant with these remarks.

3.2.4 1-D Discrete Fourier Transform

Security issues always arise when designing a robust and reliable hiding system. Applying a fixed watermark to each frame in the video leads to the problem of maintaining statistical invisibility [65]. Moreover, applying independent watermarks to each frame also presents a problem if these frames have few or no motion regions; these motionless regions in successive video frames may be statistically compared or

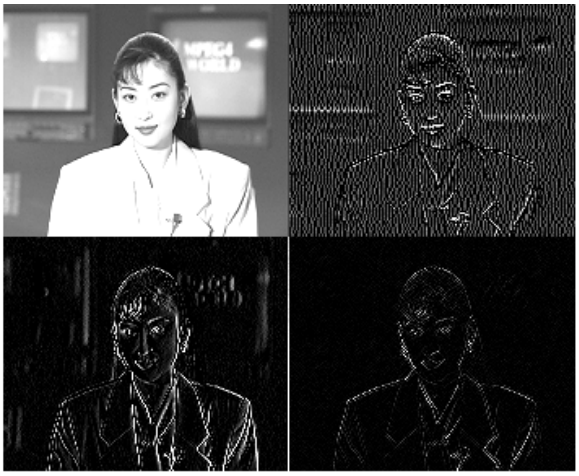


Fig. 3.15: 1-level DWT of Akiyo Y-frame showing the directive contrasts of the bands

averaged to remove independent watermarks. Attacks of such natures and scopes are normally referred to as collusion attacks [66]. The Inter-frame collusion attacks, for instance, exploit the inherent redundancy in the video frames or in the watermark to produce an unwatermarked copy of the video; two types of these attacks can be distinguished: the so called Watermark Estimation Remodulation (WER) attack, and the Frame Temporal Filtering (FTF) attack [67]. Classifying the video frames into motion and motionless frames is useful in this regard. The motion issue, in fact, is a relative one; since most of the time there is some sort of motion in the videos, but what interest us here are: the amount of the motion, how fast is the motion, and the distribution of this motion allover the space domain of the frames. Most of the video compression techniques use Inter-frame motion estimations to encode the frames; however, a useful and simpler method other than these to detect static and dynamic scenes in videos can be developed using the *1D Discrete Fourier Transform* (DFT). The 1D DFT in temporal direction transforms a group of pictures (GOP) into a temporal frequency domain; in this domain, the spatial information and temporal frequency information exist in the same frame. Higher frequencies correspond to the fast motion from one frame to other frames [50]. The 1D DFT of a video *f(x,y,t)* of

size MxNxT, in which, MxN is the size of each frame and T is the total number of frames in the Group of Pictures (GOP), is given by:

$$F(u,v,\tau) = \sum_{t=0}^{T-1} f(x,y,t)e^{-j2\Pi(t\tau/T)} \tag{3.7}$$

where u and v represent the spatial domain while τ represent the temporal domain. Taking the GOP as 5, a series of spatio-temporal frames can be established for the Akiyo video. 30 frames of Akiyo video were transformed using the 1D DFT, and since the DFT is a symmetric process in one GOP, so it is sufficient to show the first spatio-temporal frame of each group of pictures. Fig. 3.16 shows the 6 temporal frames of Akiyo video that correspond to the original 30 frames, and their norms. The edges shown in these frames represent high frequencies which correspond to motion in temporal domain, and the distribution of this motion in each frame, and hence the value of the norms represent the amount and speed of motion in each group of pictures; for instance, the intensity of the edges shows the blinking of the eyes and the movement of the head or the whole body; it can be seen that the background is motionless which is what we expect. Setting a threshold that classify video frames into dynamic and static frames can be done, and this would help us in establishing a hiding process that is more secure and reliable and can counteract the aforementioned collusion attacks. Depending on this analysis, different watermarks will be hidden in motion frames and the same watermark will be hidden in motionless frames. In fact, our color watermarks were split into bit-slices and this solved the problem in motion frames partially; furthermore and since our watermarking process is a dual one, and the hiding process takes place in transform domain rather than the space domain, and the bands being used are not confined to the high frequency ones, the

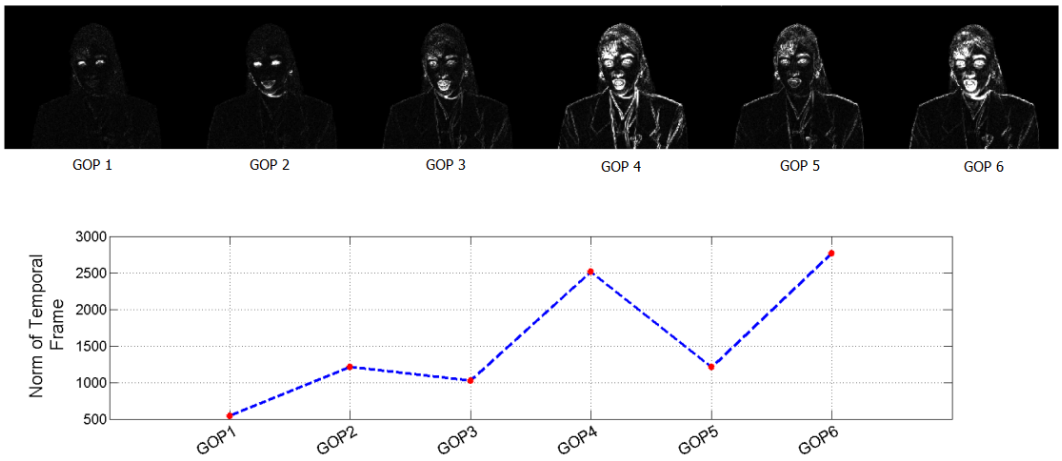


Fig. 3.16: The 1D DFT of 6 GOP's of 30 frames of Akiyo video and their corresponding norms.

effect of averaging and collusion attacks is reduced as well. This will be shown in the experimental results later on. Using 1D-DFT to establish motion information is not the only way that can be used. 3D DWT, for instance, can be used to establish Spatio-temporal components of videos [68]. Choosing the best way to establish motion in frames depends mainly on the application and other factors such as computational complexity. Since we are concerned only with estimating motion but not in precise way, using 1D-DFT is sufficient at this stage.

The pseudo code of the hiding process is illustrated down.

Hiding stage:

1. *Convert the color watermark to 12 bit-plane slices* W_j, *j=1,2 ...12*
2. *Perform SVD on the bit-slices,* $[\mathbf{U1_j\ S1_j\ V1_j}]$*=SVD(*$W_j$*).*
3. *Multiply the bit-plane slice by a pseudorandom sequence (*P_j*):* $W'_j = W_j * P_j$
4. *Multiply the scrambled bit-slice by a weighting factor k,* $W''_j = k * W'_j$
5. *Read input video frames* F_i, *i=1,2...n*

For i=1:n {

Get the YUV components. $F_i \Rightarrow Y_i, U_i, V_i$

Perform 1D Discrete Fourier Transform (1D-DFT) on Y_i *frames.*

Perform Pyramid decomposition on the Y_i *frame.*

Perform one or two levels of DWT on the error image e_0 *based on the video resolution.*

Use 1D-DFT values to choose groups of pictures (GOP) for hiding.

Use Directive Contrast (DC) to find the best DWT-subband for hiding, where: {*Choose the band with Max(DC)*}.

Hide the scrambled watermark W_j'' *in the band found in previous step.*

Perform Pyramid and DWT reconstruction of the modified Y_i' *frame.*

Reconstruct the Y_i', U_i, *and* V_i *components to get the watermarked video stream* F_i'

6. *Store or Transmit.*

3.3 Secondary Hiding Process

As mentioned beforehand, this watermarking process is a dual one, so smaller binary watermarks can be hidden in the Red , Green and Blue components of the color watermark. A method getting benefits of some properties of the Singular Value Decomposition in Grey scale images was proposed in [57]. The method which is shown in Fig. 3.17 depends on dividing the image into 4x4 smaller matrices, then SDV process is performed on each matrix, to get the U, S and V matrices. By taking the U matrices, it was realized that there is big correlation between the elements of the first columns of these matrices, so two other matrices named M_1 and M_2 can

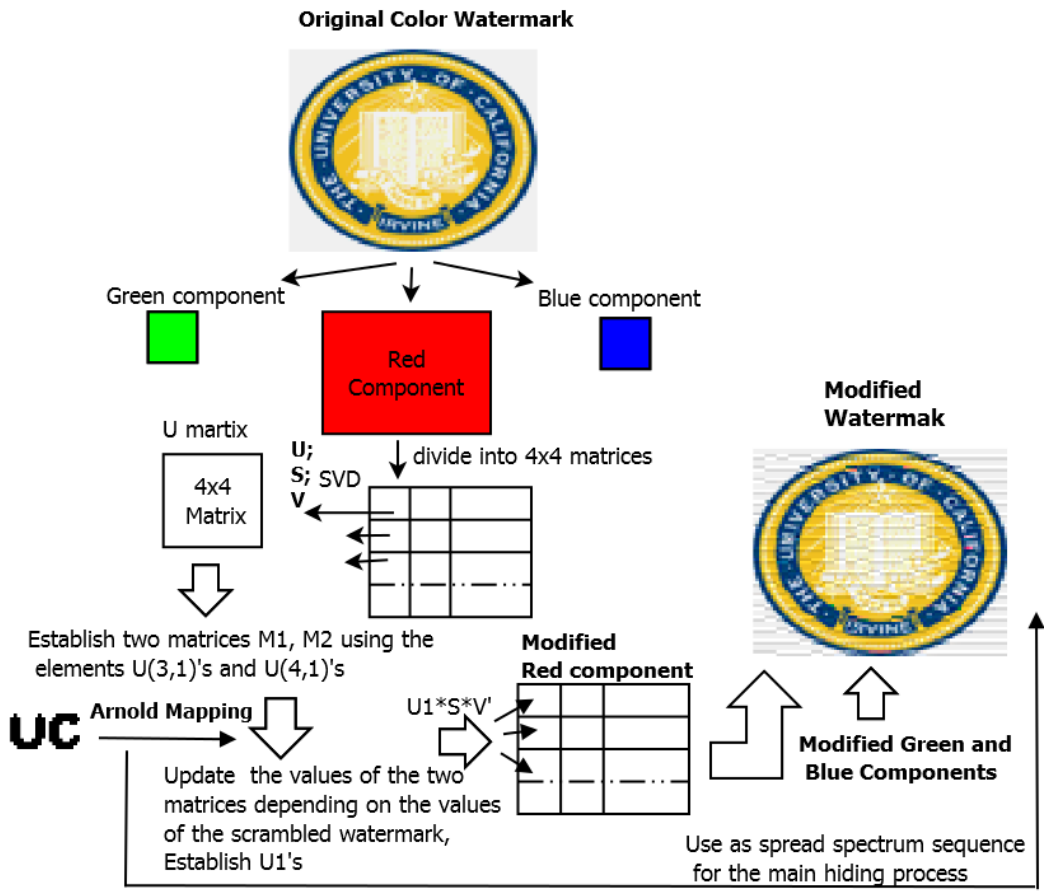


Fig. 3.17: Secondary hiding process

be established using the 3rd and the 4th elements of the first columns of the 4x4 U matrices. Experimental tests showed that these two elements provided the greatest correlations, but this does not exclude the possibility that other elements could be used. A binary watermark W of the same size of M_1 can be hidden using this relation:

$$\text{if} \quad W = 1, \begin{cases} u'_{31} = sign(u_{31}) * (u_{avg} + T/2) \\ u'_{41} = sign(u_{41}) * (u_{avg} - T/2) \end{cases} \tag{3.8}$$

$$\text{if} \quad W = 0, \begin{cases} u'_{31} = sign(u_{31}) * (u_{avg} - T/2) \\ u'_{41} = sign(u_{41}) * (u_{avg} + T/2) \end{cases} \tag{3.9}$$

where u_{avg} is the average of u_{31} and u_{41}, and T is a weighting factor. The whole secondary hiding process for RGB color watermarks is shown in Fig. 3.17. The resulting color watermark can be used for the main hiding process shown in Fig. 3.14. Chaos mapping in the shape of *Arnold Transform* is used to scramble the watermark before hiding for security reason. The point (x, y) in a square image of width N is mapped to another point (x', y') using the Arnold Transform according to this relation [69]:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 1 \\ 1 & 2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} (modN) \tag{3.10}$$

Furthermore, many iterations can be performed. Fig. 3.18 shows the original 256x256 Lena image, and the Arnold transformed versions of it using 1 to 5 iterations respectively. The resulting chaos mapped matrix can be used as a pseudo-random sequence for the embedding process shown in Fig. 3.14. This requires of course a process of rearrangement and combining of the B&W watermarks and performing multiple Arnold Transforms, since the color watermark has larger size, and Arnold Transform is applied on square images only. This is shown in Fig. 3.19, where rearrangements and multiple Arnold Transforms are performed on our B&W watermark to get a bigger watermark that can be used for our main watermarking process.

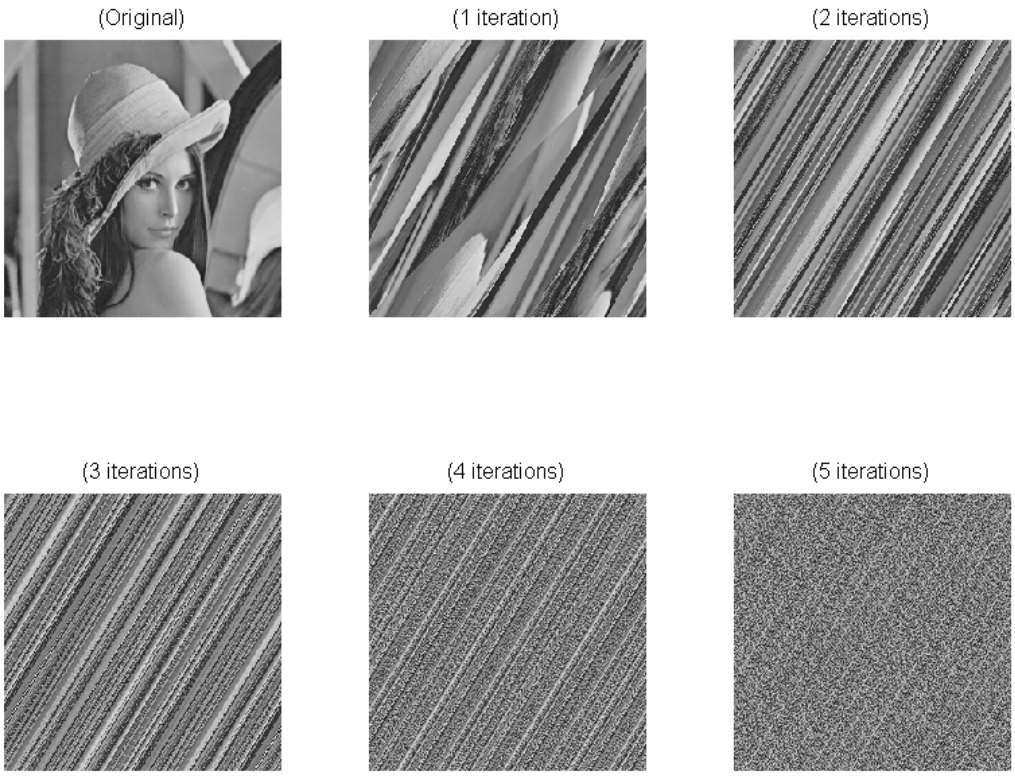


Fig. 3.18: The original 256x256 Lena image, and the Arnold transformed versions of it using 1 to 5 iterations respectively.

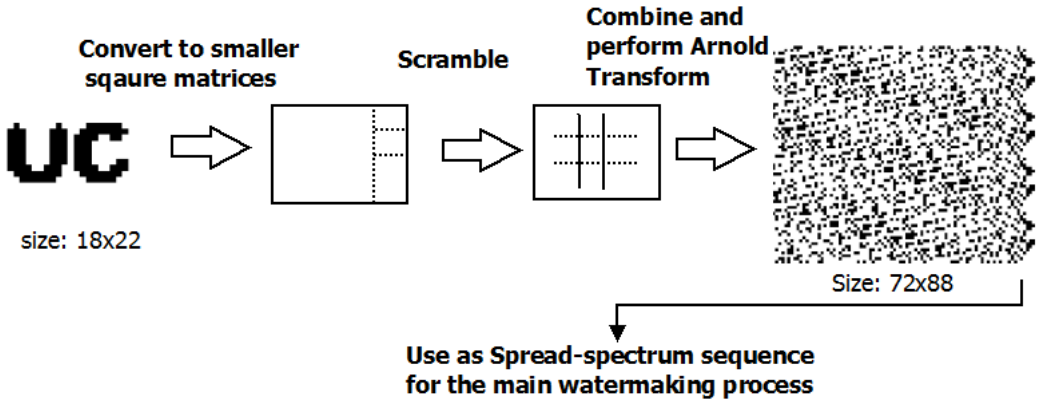


Fig. 3.19: Performing Arnold Transform on our B&W watermark.

The detection process for this secondary hiding process is the reverse of the hiding one. At the receive side and after extracting the bit-slices and the RGB components, SVD process is performed the same way shown in Fig. 3.17, then the 4x4 U matrices are established, and the elements u_{31} and u_{41} of each matrix are compared as shown in Equation 3.11. Then reverse Arnold transform can be done to reconstruct the original watermark. This secondary hiding process has no effect on the quality of the original video since it modifies our color watermark only; so that it added to the security of the system without significant extra cost. Moreover, these watermarks have relatively small sizes, so any decompositions performed on them won't add too much to the computational complexity of the system.

$$W' =, \begin{cases} 1, \text{ if } & u_{31} > u_{41} \\ 0, \text{ if } & u_{31} < u_{41} \end{cases} \tag{3.11}$$

3.4 Color Watermark Detection Process

3.4.1 General Extraction Process

The extraction process is the reverse of the hiding process. The original video is not required, but still, the knowledge of the synthesis filter banks and the pseudorandom sequence is required. To extract the watermark, a prediction of the original values of the pixels is needed [64]. The watermarked image may be considered to be the original image that is disturbed by the pseudorandom noise. Due to the effect of the lossy compression, the additive noise and the numerous video processing operations that are being performed most of the time, the watermark detection process become a challenging one to overcome the false alarm detections on one hand, and the loss of hidden data on the other hand. Moreover, security arises as a critical issue that should be taken into consideration. The attacker would try to detect and extract the watermark or at least destroy it intentionally. This practice might result in the debilitation of the whole watermarking process and the loss of its effectiveness. To overcome these situations, an enhanced detection process is proposed. The detection process is aimed at extracting the binary bit-slices for further processing. The overall extracting process is shown in Fig. 3.20. The pseudo code for the extraction process is illustrated down.

Reconstruction stage:

1. *Read the incoming video stream F_i': i=1,2...n*

2. *For i=1:n* {

   Get the YUV components $F_i' \Rightarrow Y_i', U_i', V_i'$

   Perform 1D Discrete Fourier Transform (1D-DFT) on Y_i' frames.

Perform Pyramid decomposition on the Y_i' frame

Perform one or two levels of DWT on the error image e_0 based on the video resolution.

Choose the proper frame based on the (1D-DFT).

Choose the proper DWT band based on the Directive Contrast (DC).

Get the modified DWT coefficients, Denote them Q_j where $j \in$ [1,12] based on the (1D-DFT).

Perform for each Q_j {

Q_j'=*median filter*(Q_j)

$L_j = Q_j' - Q_j$;

$LL_j = sign(L_j)$;

$Q_j'' = -1 * LL_j * P_j$}

Use 5X5 smoothing median filter (SMF): $Q_j''' = SMF(Q_j'')$

Perform SVD process on Q_j''' to get the singular values:

[$\mathbf{U2_j}$ $\mathbf{S2_j}$ $\mathbf{V2_j}$]=$SVD(Q_j''')$

Use the approximated singular values $\mathbf{S2_j}$ *as:*

$\mathbf{U1_j} * \mathbf{S2_j} * \mathbf{V1_j} \Rightarrow$ *reconstruct the bit-plane.*

}*end*

3. *Use the bit-planes to reconstruct the color watermark.*

3.4.2 Enhanced Detection Process

A noise-elimination technique can be used to extract the hidden watermark pixels; to achieve that, a spatial convolution mask of size 5x5 can be used to smoothen the extracted coefficients. Experimental results showed that the 5x5 mask gave superior performance compared to the 3x3 mask under different circumstances such as noise

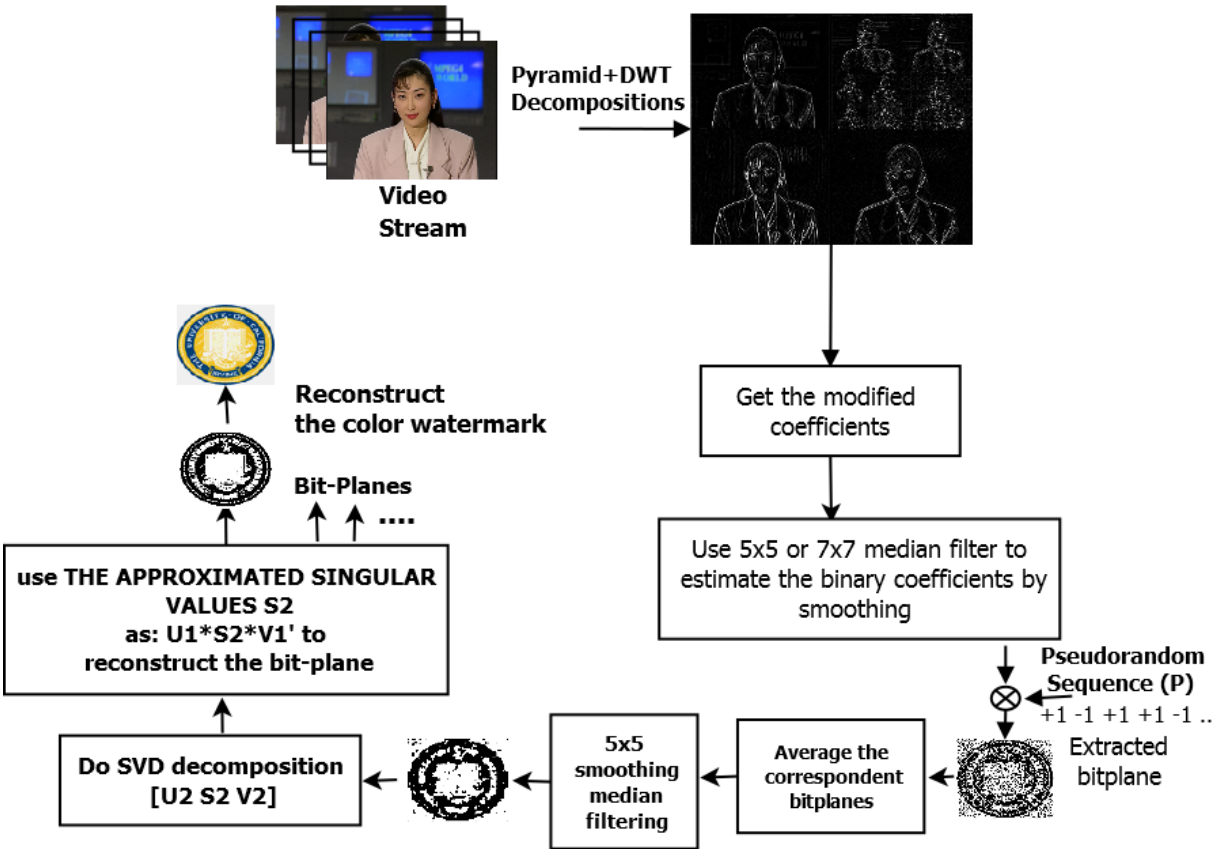


Fig. 3.20: The color watermark extraction process

addition and compression processes. The enhanced detection process is then set to use multiple extracted watermarks, which were embedded in different video frames in the first place, for our final estimation process.

Let's assume that the extracted watermarks are grouped in a set W where $W = w_1, w_2, \ldots w_n$. To choose the set of watermarks that can be used in the final estimation process, cross-correlation test can be performed between every two extracted watermarks w_i and w_j. The cross-correlation between two matrices A and B is given according to Equation 3.12:

$$R = \frac{\sum_m \sum_n \left(A_{mn} - \overline{A}\right) \left(B_{mn} - \overline{B}\right)}{\sqrt{\left(\sum_m \sum_n \left(A_{mn} - \overline{A}\right)^2\right)\left(\sum_m \sum_n \left(B_{mn} - \overline{B}\right)^2\right)}} \tag{3.12}$$

Where $\bar{A}$ is the mean of A and $\bar{B}$ is the mean of B. The cross correlation test is used primarily because of the fact that the extracted watermarks are the original watermarks that were embedded and thereafter corrupted due to the numerous video processing operations that were performed, and the intentional and unintentional attacks that the watermarked videos were subjected to. This can include geometric, statistical, and other types of attacks. Hence the extracted watermarks can be considered as noisy versions of the original ones, or in other words noisy signals. The cross correlation is a measure of similarity between two signals, and hence, the extracted watermarks would have a certain amount of similarity. Using this analogy, new set of extracted watermarks W_1 can be established.

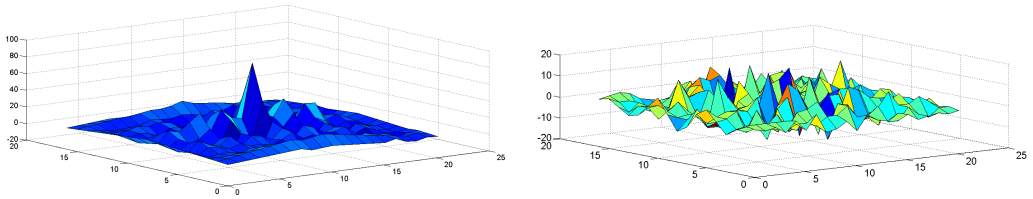
When the cross-correlation results in a significant peak at the center, this means that the two sets of extracted coefficients are useful and could be used for our final estimation process. Hence they could be included in the final watermarks set W_1. On the contrary, if the cross-correlation process did not result in significant peak at the center, this means that one or both of the sets of extracted coefficients are highly corrupted and hence one or both of them will be excluded from the final watermarks set. This can be shown in Fig. 3.21 where Fig. 3.21(a) shows a plot of the cross-correlation matrix between two extracted watermarks that are highly correlated and can be used in the final estimation process, while Fig. 3.21(b) shows the cross-correlation matrix between two extracted watermarks that one or the two of them are corrupted and therefore are discarded from the final estimation process. A threshold value for our estimation process is to be defined and set later depending on many factors such as the number of embedded watermarks and the intensity of the expected attacks.

The cross correlation between binary images as mentioned afore is a measure of sim-

ilarity between them; this means that the flipping of any pixel would result in a reduction in the similarity parameter. If $w_i \in W$, then w_i is cross-correlated with all the other extracted watermarks in the set, the average cross-correlation parameter is then computed. This process is repeated for the other watermarks in the set. A new set of average cross-correlation parameters is established that corresponds to each extracted watermark. By establishing a threshold h for the average cross-correlations, each extracted watermark that does not achieve the threshold test is excluded from the new set W_1. To get the final extracted watermark w_e, an averaging process is performed on the watermarks in the set W_1, where:

$$w_e = Ave\{W_1\} \tag{3.13}$$

Using the averaging process is attributed to statistical analysis. The correlation coefficient R between two matrices A and B which is used to measure the final performance is given in Equation 3.12; the mean value of a binary image A is the expected value of A or E(A). Assuming that the probability of 1's at the input is p_1 and that the probability of flipping is p as shown in Fig. 3.22, and by taking into account that the average or mean value of a binary image is the expected value, then:



(a) **(b)**

Fig. 3.21: 3-D plots of the Cross-correlation matrices of two extracted watermarks.

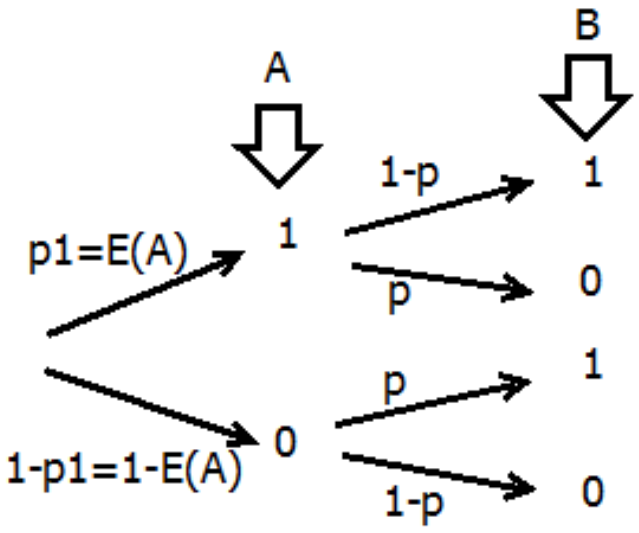


Fig. 3.22: Expected values of the input and output binary images

$$\bar{A} = E(A) = p_1 \tag{3.14}$$

Furthermore, the probability of getting 1 at the output $= \bar{B} = p_1 * (1-p) + (1-p_1) * p$, and by taking Equation 3.14 into account it can be written as:

$$\bar{B} = E(B) = E(A) + (1 - 2 * E(A)) * p \tag{3.15}$$

Assuming that the input watermark A corresponds to a constant matrix during our watermarking process, the flipping probability of the pixels p is the only variable in the above equation. Furthermore, By comparing Equations 3.12 and 3.15, it can be shown that the correlation between the two matrices A and B is a function of the flipping probability of the pixels, and hence by averaging the extracted watermarks,

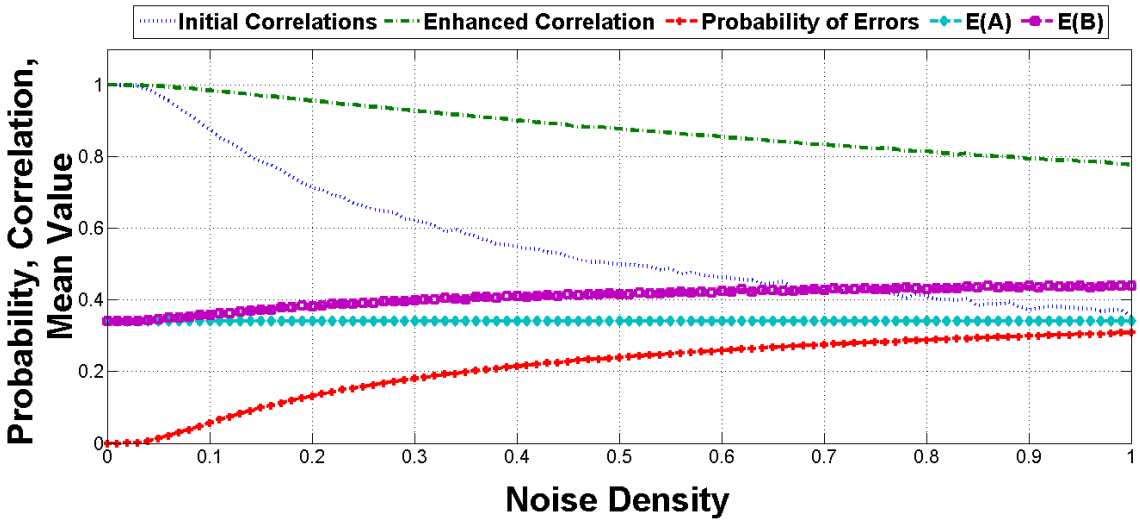


Fig. 3.23: The enhanced correlations vs. noise density

the flipping effect is eliminated to some extent and an enhanced version can be reconstructed as far as p is not equal to 0.5 which corresponds to an entropy value of one. To illustrate this analysis, Fig.3.23 shows the relationships between these parameters when Gaussian noise with zero mean and different variances is applied to a random binary watermark; here the term density corresponds to the variance of noise for illustration purposes.

3.4.3 Estimating the Singular Values

The next step in our algorithm includes estimating the singular values that best approximate the original singular values for each slice. The singular values of the extracted watermark so far provided good approximation, but further processing is done to minimize the errors. The extracted bit-slices are smoothened by applying 5x5 or 7x7 median filtering on them, then, the SVD process is performed on each one. Fig. 3.24 shows the approximated singular values for our 72x88 watermark, the 8th slice of the red component. It can be seen that the error was minimized which helps in getting near perfect reconstruction. Moreover, Fig. 3.25 and Fig. 3.26

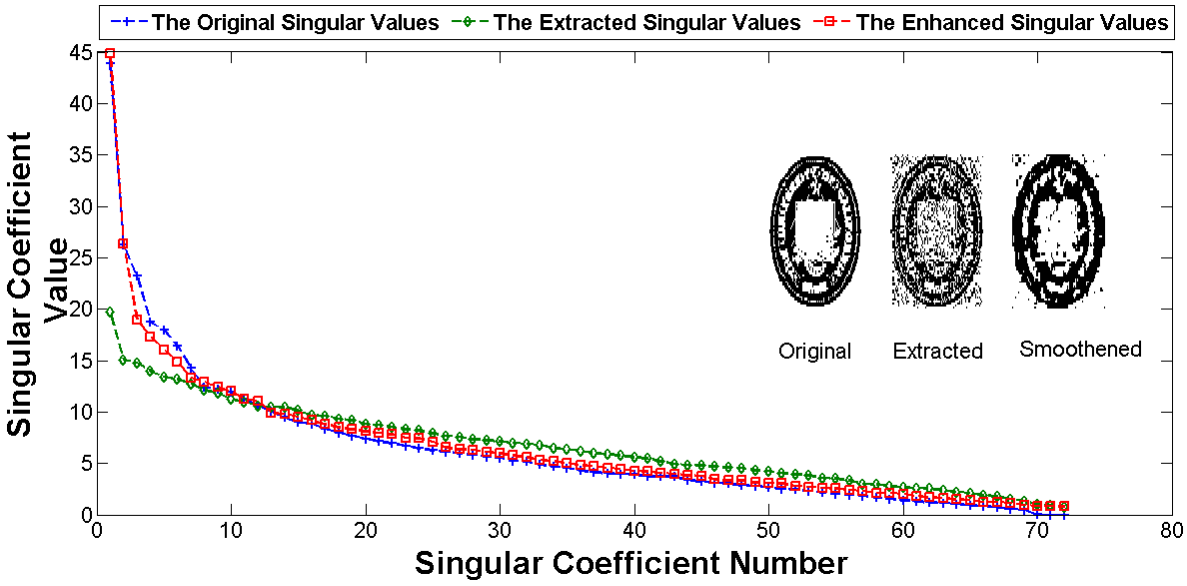


Fig. 3.24: Original and estimated Singular values of one Bit-Slice of the color watermark

show the original and filtered extracted bit-slices of the color watermark respectively. These bit-slices correspond to the slices in Fig. 3.13. It can be seen that the original extracted bit-slices were noisy; while the noise was reduced when filtered using the median filter. Reducing the level of noise in the bit-slices enhanced the values of the singular values and hence reduced the errors in these values. This in turn helped in getting near perfect extracted watermarks. The original and approximated singular values for these bit-slices are shown in Fig. 3.27.

Furthermore, Table. 3.1 shows the Root-Mean-Square-Error (RMSE) values between the original singular values and the extracted and enhanced singular values that we've got using our approach. It can be shown that our approach enabled us to get singular values with RMSE's in the limits of 2.3. The enhancement after smoothening was at least 25%. It can be seen that the enhancement increases with the increase in the resolution of the videos; for HD (1920x1080) video, the enhancement reached 60%; this is expected since the larger resolution videos have more available spectral bands and hence more capacity of hiding in the DWT coefficients and this enabled us to

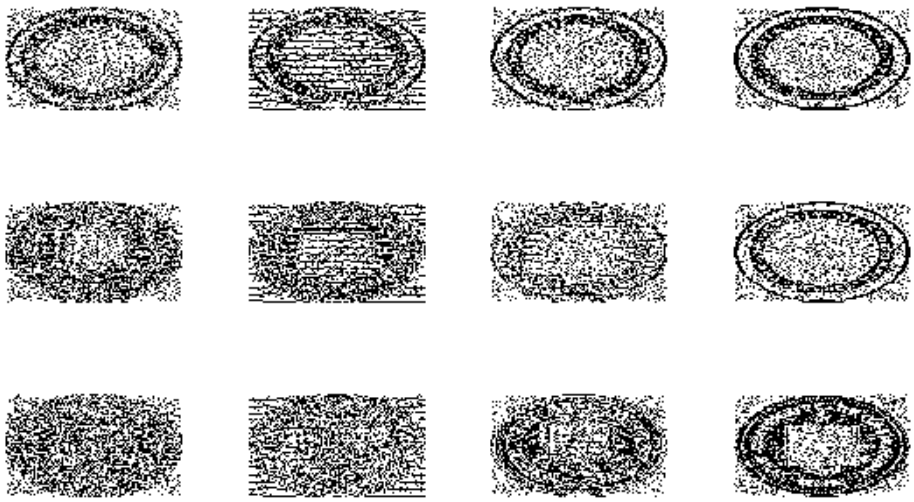


Fig. 3.25: Extracted bit-slices using Pyramid-DWT method.

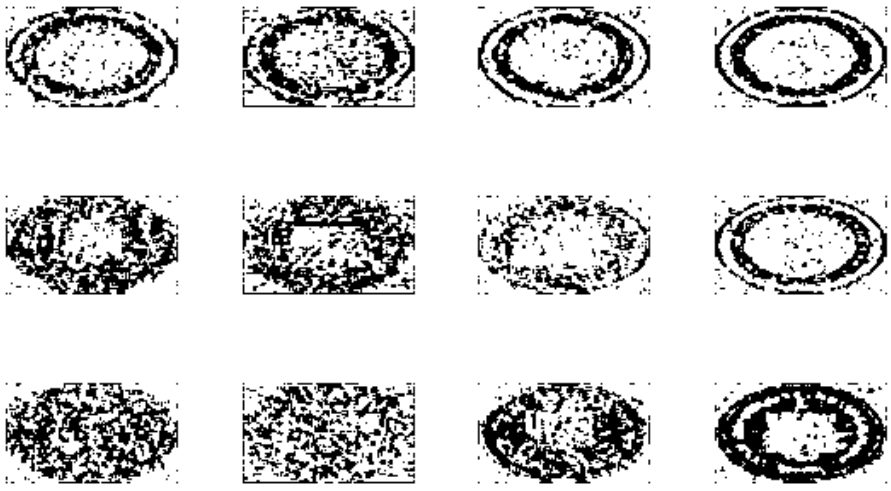


Fig. 3.26: Filtered extracted bit-slices using Pyramid-DWT method; the slices correspond to slices of Fig. 3.13.

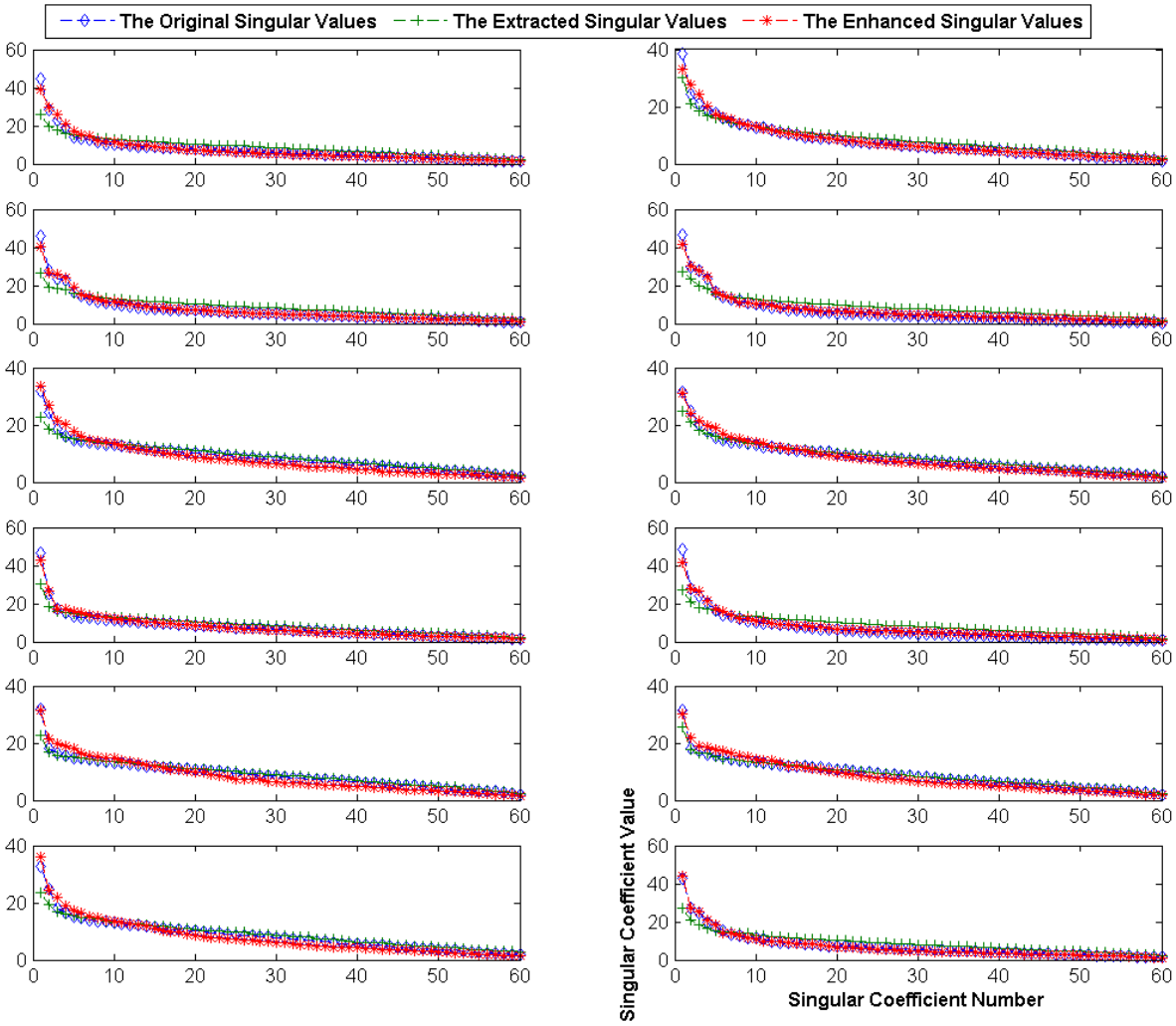


Fig. 3.27: Original and estimated Singular values of the Bit-Slices of the color watermark.

Table 3.1: RMSE between original and extracted singular values

Video	**RMSE (original and extracted))**	**RMSE (original and enhanced**	**% Enhancement**
Akiyo	3.16	2.35	25.6%
Foreman	3.37	2.44	27.6%
Mother-daughter	3.21	2.39	25.5%
BasketballDrill	4.03	2.52	37.4%
BasketballDrive	2.97	1.19	59.9%

have better establishment of the hidden watermark.

The proposed method can perform full reconstruction of the hidden data under no attacks to moderate attacks circumstances. The *stability of singular value theorem* indicates that, when there is a little disturbance with a matrix A, the variation of its singular value is not greater than 2-norm of disturbance matrix, where 2-norm is equal to the largest singular value of the matrix [70]. From Fig. 3.24 and Table. 3.1, it can be seen that by using our method, the variation in singular values were further minimized which would help in getting full reconstruction of the hidden watermark. The secondary B&W watermark can be extracted after getting the three RGB components as was shown in Section 3.3.

3.5 Experimental Results

In this section we demonstrate the performance of our algorithm using our proposed method on different standard videos under different attacks. Furthermore, it will be compared with many proposed algorithms in the field. Watermarked and unwatermarked versions of a frame of *Mother-daughter* video (352x288 pixels) are shown in Fig. 3.28. The secondary B&W watermark of size 18x22 and the original and modified color watermarks of size 72x88x3 are shown in Fig. 3.29. The secondary watermark

Fig. 3.28: (a) Original mother-daughter frame, (b) watermarked frame

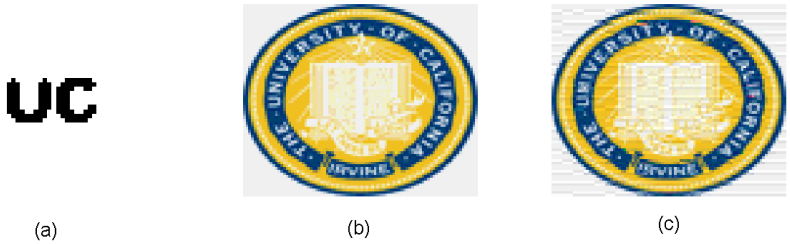


Fig. 3.29: (a) 18x22 B&W original secondary watermark, (b) 72x88x3 Original color watermark (c) 72x88x3 Modified color watermark

and the extracted version are shown in Fig. 3.30. Furthermore, the modified hidden color watermark and the extracted one are shown in Fig.3.31. The first frames of the other watermarked standard YUV videos: *Akiyo, Foreman, BasketballDrill* and *BasketballDrive* are shown in Figures 3.32, 3.33, 3.34 and 3.35 respectively; it can be seen the good quality after performing our data hiding process.

Fig. 3.30: (a) Original B&W secondary watermark, (b) The extracted watermark

Fig. 3.31: (a) Modified hidden color watermark (b) Extracted color watermark

Fig. 3.32: First frame of the watermarked Akiyo video.

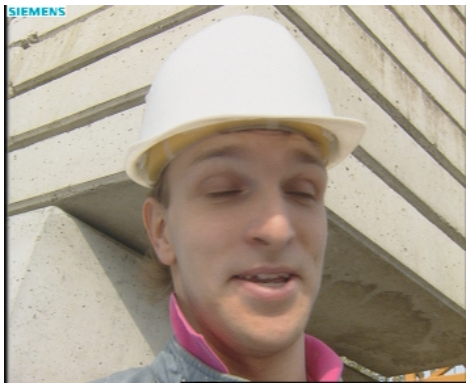


Fig. 3.33: First frame of the watermarked Foreman video.

Fig. 3.34: First frame of the watermarked BasketballDrill video.

Fig. 3.35: First frame of the watermarked BasketballDrive video.

3.5.1 Visual Quality and Extraction process

Our algorithm performance will be evaluated in terms of PSNR between the original and the watermarked videos, and the Bit-Error-Rate (*BER*) between the original and the extracted watermarks for the standard YUV videos: *Akiyo, Foreman, Mother-daughter, BasketballDrill* and *BasketballDrive*. The YUV space was selected for the raw videos because it is more efficient than the traditional RGB space and provides flexibility in choosing the resolutions required. For the CIF (352x288) videos, the watermark shown in Fig. 3.31 was used, while for the other two videos a color watermark of size 60x104x3 was used. The secondary watermarks are of sizes 18x22 for the smaller CIF videos and 15x26 for other videos. In these tests, multiple number of frames were watermarked. Moreover, since the metric measurements of the videos qualities are not always enough because the Human Visual System (HVS) perceives the quality in other ways, the Structural Similarity Index (SSIM) was evaluated for the watermarked frames of these videos. Table 3.2 shows these evaluated values for the videos. It can be seen that our proposed method achieved full reconstruction of the hidden color watermarks, moreover, the PSNR's were above 50 dB for all videos and sometimes over 60 dB which indicates good qualities. These good qualities were assured by the high values of the SSIM indices; all the SSIM indices were over 99% which indicates excellent visual perceptual quality and minimal distortions due to the hiding process.

Perceptual transparency test of the watermarked videos according to ITU-R Rec.500 and ITU-T Rec. P.910 was performed using 25 people; this test is called the Mean Opinion Score (MOS), and it reflects how humans react to images and videos qualities, and how they evaluate them. The videos were shown to them without telling if the displayed video is watermarked or not. Table 3.3 shows the quality ratings of the

Table 3.2: The BER, PSNR and SSIM for the main watermarking process

Video Stream	**BER**	**PSNR**	**SSIM**	**Extracted Water-mark**
Akiyo	0	54.4	0.9959	
Foreman	0	53.2	0.9967	
Mother-daughter	0	69.8	0.9985	
BasketballDrill	0	52.4	0.9994	
BasketballDrive	0	64.4	0.9997	

Table 3.3: Mean Opinion Score (MOS) of the perceptual transparency of the watermarked videos (number of persons is 25).

Rating	**5**	**4**	**3**	**2**	**1**	Average
Original Akiyo	23	2	0	0	0	4.92
Watermarked Akiyo	22	3	0	0	0	4.88
Original Foreman	23	2	0	0	0	4.92
Watermarked Foreman	23	2	0	0	0	4.92
Original Mother-daughter	25	0	0	0	0	5
Watermarked Mother-daughter	24	1	0	0	0	4.96
Original BasketballDrill	23	2	0	0	0	4.92
Watermarked BasketballDrill	23	2	0	0	0	4.92
Original BasketballDrive	23	2	0	0	0	4.92
Watermarked BasketballDrive	22	3	0	0	0	4.88
Impairment	Imperceptible	Perceptible, not annoying	Slightly annoying	Annoying	Very annoying	
Quality	Excellen	Good	Fair	Poor	Bad	

videos; the table shows the number of individuals who gave their ratings for each video. It can be seen that most of the people did not notice any difference between the original and watermarked videos and they gave the same ratings for them. The evaluation process was almost unbiased, since the individual does not know if the shown video is watermarked or no. On the other hand, the human visual perceptual response to videos is a subjective issue; Table. 3.4 shows the bias in this subjective test; it can be seen that the bias is very small in these kind of tests. Moreover, the table shows that the average perceptual MOS difference between the original and watermarked videos is 0.49%. Thus, the watermarking process is imperceptible; also, the qualities are mostly excellent.

Table 3.4: The average MOS's and the human visual bias for the standard test videos.

Test videos	Optimal Rating	Mean Opinion Score (MOS)	Percentage Bias
Original Videos	5	4.936	1.28%
Watermarked Videos	% change in MOS=(4.936-4.9120)/4.936=0.49%		

Table 3.5: The BER's for the secondary hiding process in the three color components

Video Stream	**BER (red)**	**BER (green)**	**BER (blue)**	**Average %BER**
Akiyo	0.0379	0	0	1.26
Foreman	0.0379	0	0	1.26
Mother-daughter	0.0379	0	0	1.26
BasketballDrill	0.0333	0	0	1.2
BasketballDrive	0.0333	0	0	1.2

Our watermarking process is a dual one, so the BER's for the secondary watermarking process in the three components: the Red, the Green, and the Blue ones are shown in table 3.5. It can be seen that the BER's were zeros for the green and the blue components while they were in the limits of 3.7% in the red component. The average values were in the limits of 1.2% which indicates high reliability of the system, especially that this is a secondary hiding process aimed at increasing the security of the overall system.

3.5.2 Robustness Against Attacks

To evaluate our proposed method, several attacks were performed to measure the degree of robustness of the system. The attacks that were used are: additive noise which include (Gaussian, Poisson and salt-and-pepper noise), contrast adjustment,

histogram equalization, median filter, rotation, frame dropping, frame averaging, frame swapping, transcoding, and H.264 and H.265 compressions. Moreover, the algorithm will be compared with other works in the field of video watermarking; it will be compared with the methods of [49], [50] and [58] under numerous attacks mentioned before.

Since the compression attacks and transcoding are among the most serious attacks that any video is subjected to, we will start our tests with them. The compressions being used are H.264 and H.265 under different data rates. High compression rates affect the perceptual quality of the videos; and since this work is concerned with protection of the ownership, so it would be useful to show the degradation in video quality with different compression rates before showing the algorithm performance. Two metrics will be used to show the quality; one is the Structural Similarity Index (SSIM), and the other is the PSNR while the compression being used is H.265. Table 3.6 shows the degradation in quality according to the data rates being achieved. One frame of each of these videos under the lowest data rate is shown to illustrate the downgraded quality. This table also shows that objective tests are not always in line with the HVS perception of images or subjective tests.

Fig. 3.36 shows the performance of our system with the use of H.264, while Fig. 3.37 shows the performance of our system using the H.265. It can be shown from these two figures that our system is robust against compression process. Of course H.265 is more efficient than H.264, so that it can be seen that lower data rates in H.265 result almost in the same BER's as those of H.264.

The transcoding process for videos is a common process in multimedia fields. Depending on the platform that the video is used in, numerous multi-media containers are used; some popular containers are:

Table 3.6: Data-rates and the corresponding SSIM's (the upper values) and PSNR's (the lower values) of test videos after applying H.265 compression

Video Stream	**Bit-Rate: (kb/s)**	**SSIM PSNR (dB)**	**Bit-Rate: (kb/s)**	**SSIM PSNR (dB)**	**Bit-Rate: (kb/s)**	**SSIM PSNR (dB)**	**One frame with last Bit-Rate**
Akiyo	1093	0.9813 49.4	75.5	0.9479 40	32.6	0.7370 30.7	
Foreman	2670	0.9745 45.7	162.0	0.9068 36.7	42.5	0.7282 28.3	
Mother-daughter	1426	0.9873 48.1	70.8	0.9675 40.9	29.4	0.8124 31.2	
Basketball-Drill	19444	0.9532 42.31	553.2	0.8983 36.0	85.5	0.6847 27.7	
Basketball-Drive	90963	0.9679 45.9	1389.2	0.8970 38.8	268.2	0.7987 31.6	

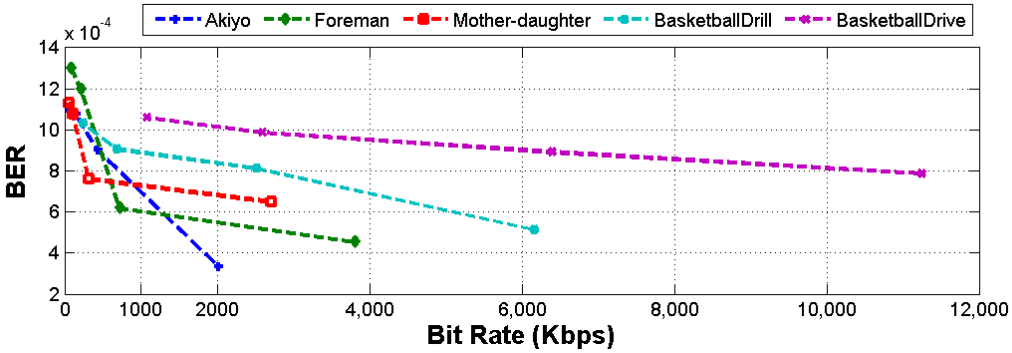


Fig. 3.36: BER of the hiding process when applying H.264 compression

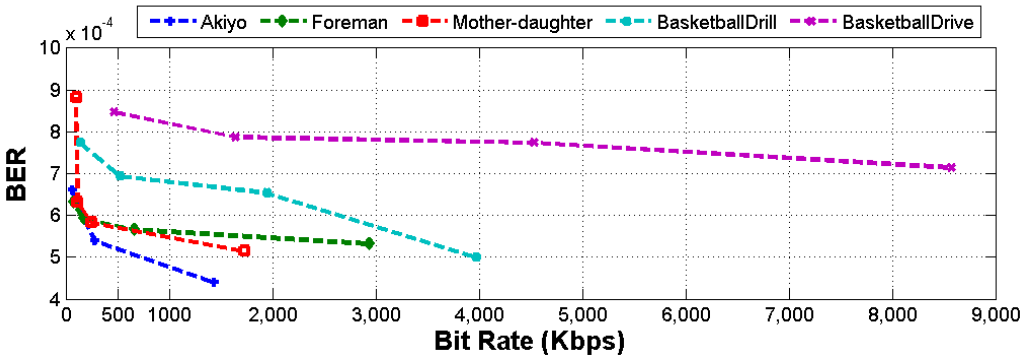


Fig. 3.37: BER of the hiding process when applying H.265 compression

- MP4: which is the standard audio and video container for the MPEG-4 multimedia portfolio.
- 3GP: which is used by many mobile phones.
- AVI: which is the standard Microsoft Windows container.

Of course this would result in many features such as variable frame rates, streaming capabilities and different data rates depending on the compression technique that is used and the format that is adopted. Moreover, Transcoding involves as well rescaling the videos to fit different display devices. Fig. 3.38 shows the first frame of Mother-daughter video using the three formats: MP4, AVI and 3gp rescaled to different resolutions. The raw video is originally YUV 4:2:0 352x288. The watermarked videos were transcoded to these formats and rescaled, then the watermarks were extracted. Table. 3.7 shows the BER's and the data rates for these videos. Also the extracted color watermarks are shown for the 3gp transcoding process. Table. 3.8, however, shows the secondary hiding performance with the application of the transcoding processes. It can be seen that our dual hiding process is able to survive the transcoding attack with high reliability.

Our proposed method was tested under different other attacks and compared with

Fig. 3.38: The first frame of the watermarked mother-daughter video transcoded to: (a) AVI 832x480, (b) 3gp 480x320, (c) MP4 320x240

Table 3.7: The BER's and Data-rates of the color watermark extraction from different transcoded videos

Video Stream	**MP4, 320x240**		**AVI, 832x480**		**3gp, 480x320**		**Watermark**
	BER	**Bit-Rate: (kb/s)**	**BER**	**Bit-Rate: (kb/s)**	**BER**	**Bit-Rate: (kb/s)**	
Akiyo	0.0299	201	0.0298	890	0.0362	349	
Foreman	0.0290	507	0.0285	2173	0.0359	895	
Mother-daughter	0.0290	247	0.0295	1199	0.0295	409	
Basketball-Drill	0.0179	548	0.0214	1920	0.0115	835	
Basketball-Drive	0.0103	501	0.0119	1914	0.0106	1171	

Table 3.8: The BER's of the secondary watermark extraction from different transcoded videos

Video Stream	**BER: MP4, 320x240**	**Extracted Watermark**	**BER: AVI, 832x480**	**Extracted Watermark**	**BER: 3gp, 480x320**	**Extracted Watermark**
Akiyo	0.0589		0.0589		0.0623	
Foreman	0.0623		0.0598		0.0640	
Mother-daughter	0.0606		0.0614		0.0606	
BasketballDrill	0.0615		0.0615		0.0564	
BasketballDrive	0.0547		0.0573		0.0590	

methods [49], [50] and [58]; the attacks used were:

- Gaussian noise with variances 0.1 and 0.5.
- Poisson noise.
- Salt & pepper noise with 2% and 6% densities.
- Median filtering 3x3 and 5x5.
- Contrast adjustment.
- Histogram equalization.

Table. 3.9 shows our system performance using the *Mother-daughter* video for both the main and secondary watermarking processes when applying the aforementioned attacks, where BER1 and BER2 denote the bit error rates for the color and B&W watermarks respectively. Moreover, Fig. 3.39 shows the correlation values for the extraction process of the color watermarks under those common attacks using our method and other methods. It can be seen that our method provides almost the best correlation values of all the other methods. Furthermore, our method was able to provide high correlation values; in fact more than 97% for all the attacks.

Geometrical attacks are ones of the most aggressive and serious attacks in multimedia watermarking fields. One of them which is the scaling attack was addressed in the videos transcoding beforehand; the other geometrical attack is the rotation attack; it involves rotating all or some of the video frames around the center point for specific degrees. Most traditional transform watermarking methods fail to survive this attack unless some sort of readjustment and realignment were applied before the extraction process. Because of *the rotation invariant property* of the singular values [70], it is possible to survive this attack with good robustness. Our watermarked videos were

Table 3.9: BER's of our extraction method with several images processing attacks

Attack	Watermarked Frame	BER1	Extracted watermark	BER2	Extracted watermark
Gaussian noise with variance 0.1		0.0114		0.0244	
Gaussian noise with variance 0.5		0.0115		0.0227	
Poisson noise		0.0112		0.0244	
Salt & pepper noise 2%		0.0096		0.0210	
Salt & pepper noise 6%		0.0114		0.0244	
Contrast Adjust-ment		0.0051		0.0160	
Median Filtering 3x3		0.0095		0.0202	
Median Filtering 5x5		0.0111		0.0227	
Histogram Equaliza-tion		0.0048		0.0160	

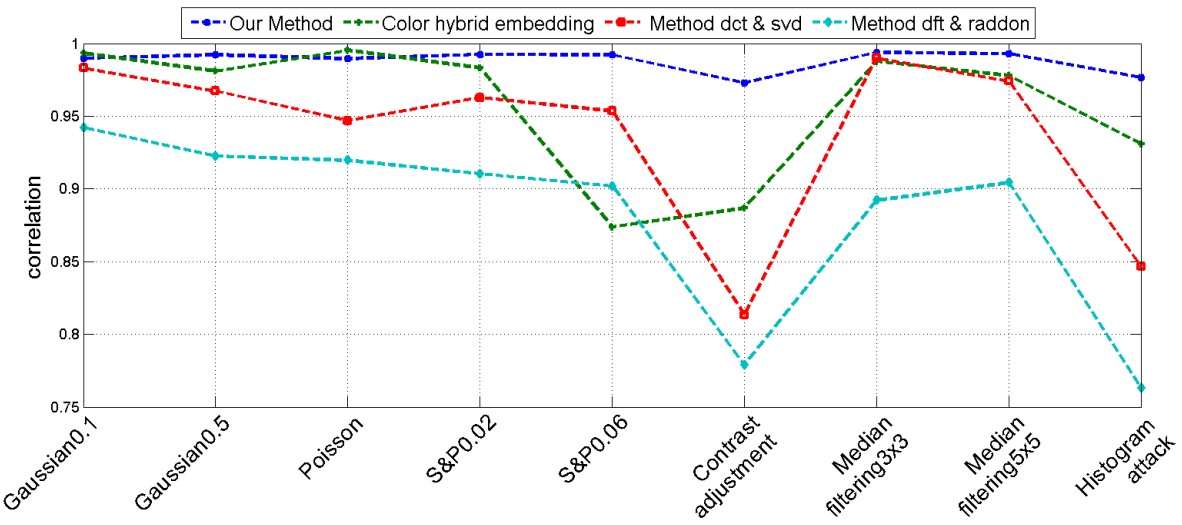


Fig. 3.39: Correlations for the proposed extraction process and methods 'Color hybrid embedding [49]', 'dct&svd [58]', 'dft&raddon [50]' for the color watermarks under some common attacks

subjected to several rotations (1, 2, 5, 10 and 180 degrees), and then our extraction process was applied as mentioned beforehand. Fig. 3.40 shows the BER's at these rotations for both our method and method of [49] which has the best BER's of all other methods. It can be seen that our method outperformed the method in [49] for all the assumed angles of rotations.

The other attacks that are common in video watermarking are the temporal attacks. From their name, it's clear that temporal attacks are more related to multiple frames rather than one frame which in turn can be called space attacks, and which we've addressed so far. Common temporal attacks are: *Frame Dropping*, *Frame Swapping*, and *Frame Inserting*. Since the swapping attack involves inserting process, so we will test our method against frame dropping and frame swapping attacks only. Different numbers of frames were dropped from the standard videos that were used, and afterwards, the average BER's of our extraction process were evaluated. Our results were compared with the method in [49] which outperformed other methods; here, the average BER's that were achieved in the method of [49] were used; these results are

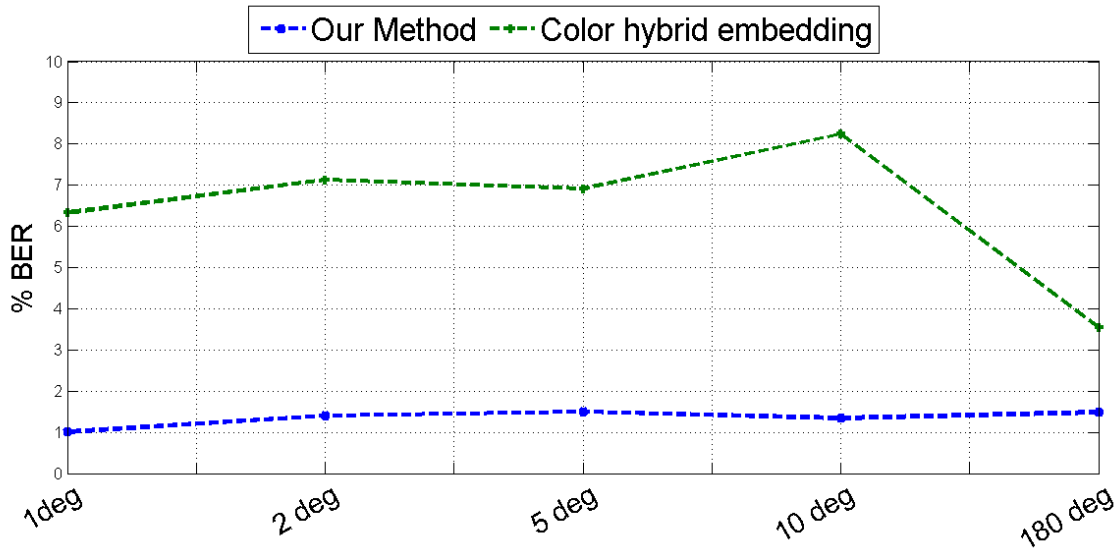


Fig. 3.40: BER's of the extraction process and the method of 'Color Hybrid Embedding [49]' under several rotation attacks

shown in Fig. 3.41. Likewise, different numbers of frames were swapped over the course of running our watermarked videos. In fact, we expect that this attack is less aggressive than the dropping attack, were whole frames are lost and hence important hidden data are lost as a result. Here in frame swapping attacks, the successive video frames especially before scene changing are highly correlated and as a consequence, in general, the hidden data are lost partially only. Fig. 3.42 shows the performance of our method as well as the method in [49] in terms of BER's. For both attacks, our method outperformed the method in [49].

To further demonstrate our system performance, Table. 3.10 shows the performance of our system for both the color watermark and the secondary watermark under specific geometric and temporal attacks for the video of *mother-daughter*. Here the rotation was 10 degrees counter-clockwise, 20% of the frames were dropped randomly for the dropping attack, and 33% of the frames were swapped randomly for the swapping attack. Moreover, cropping attacks were performed for randomly chosen frames, where 25% of the frame was cropped from either the left-upper or the right-

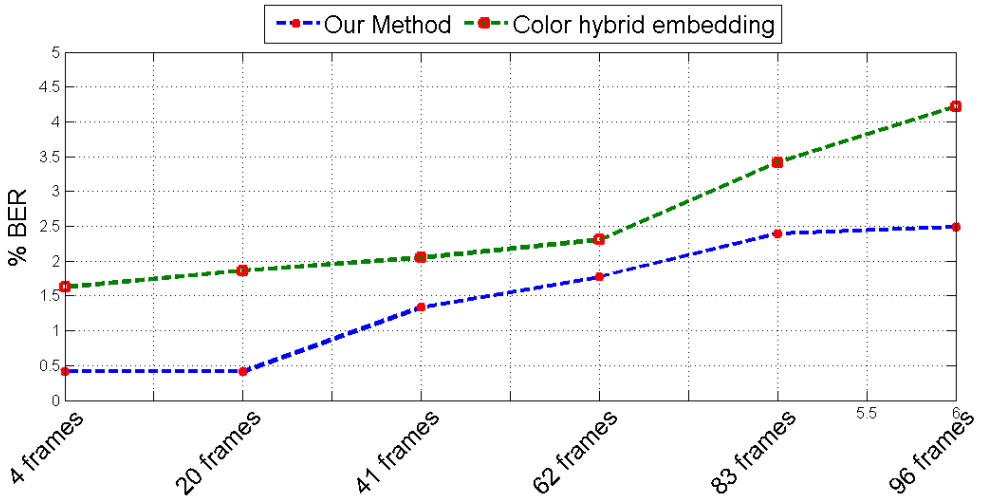


Fig. 3.41: BER's of the proposed extraction process and the method of 'Color Hybrid Embedding [49]' under different frame dropping attacks.

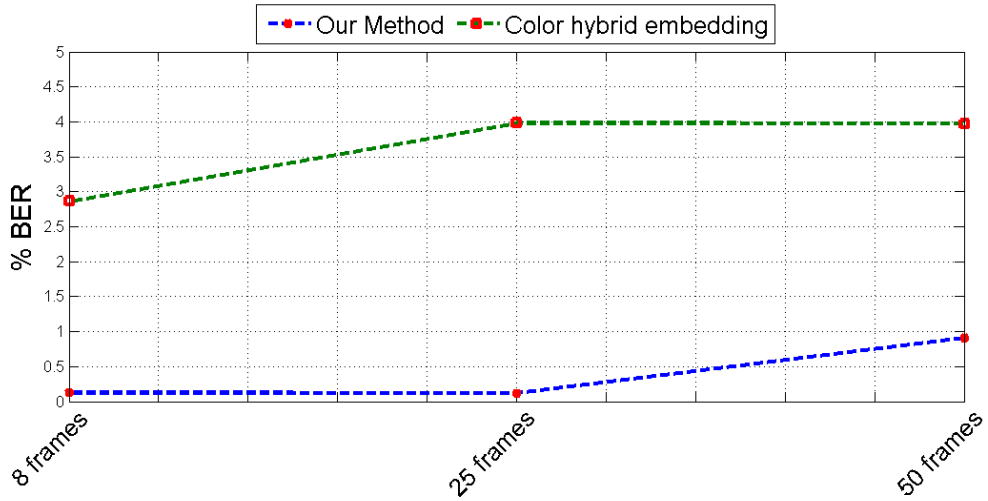


Fig. 3.42: BER's of the proposed extraction process and the method of 'Color Hybrid Embedding [49]' under different frame swapping attacks.

lower sides. It can be seen that our system is able to survive these attacks with high reliability in both the main and the secondary watermarking processes. Here for the frame dropping attack, the BER's depend mainly on the length of the video and hence the total number of frames that are watermarked; this means that the performance can be enhanced dramatically with practical real life videos rather than the test videos which tend to be short.

Our proposed technique, moreover, was tested against averaging attack. We discussed the collusion attacks and the ways they could be implemented in Section 3.2; furthermore, we introduced our method of using the 1D DFT to find the static and dynamic frames. Our data hiding scheme was performed using these frames accordingly. In fact, frame averaging is considered a collusion attack aimed at removing the hidden watermark. We performed frames averaging in the same scenes of the videos to see the results; Table 3.11 shows our results for the five standard videos, and the BER's for both the main watermarking process (BER1) and for the secondary watermarking process (BER2), and the extracted watermarks. It can be seen that our dual watermarking process was able to survive the averaging attack with high reliability.

To ensure the security of the scheme, the watermarking process was tested for false alarms attacks. That's when the system indicates the existence of the inserted watermark while, in fact, no watermark was embedded or another watermark was the one that was hidden actually. For the test videos, 500 different generated random B&W watermarks were embedded and the right watermark was set to be the 200th one. The results are shown in Fig. 3.43. It can be shown that our system responded with low correlations to all the embedded watermarks except the 200th one, which indicates a high reliability from this aspect.

Furthermore, it's useful to check the detection performance of the system. The detection process is more concerned with true or false detections of the presence of a

Table 3.10: BER's of our method with specific geometric and temporal attacks

Attack	Watermarked Frame	BER1	Extracted Watermark	BER2	Extracted Watermark
Rotation (10 deg)		0.0111		0.0253	
Frame dropping (20%)		0.0218		0.0463	
Frame swapping (33%)		0.0078		0.0177	
Frame Cropping (25%)		0.0168		0.0438	
Frame Cropping (25%)		0.0146		0.0320	

Table 3.11: BER's of our method with Frames Averaging attacks

Attack	**Four Averaged Frames**	**BER1**	**Extracted Watermark**	**BER2**	**Extracted Watermark**
Akiyo		0.0054		0.0185	
Foreman		0.0068		0.0202	
Mother-daughter		0.0069		0.0185	
Basketball-Drill		0.0042		0.0214	
Basketball-Drive		0.0039		0.0197	

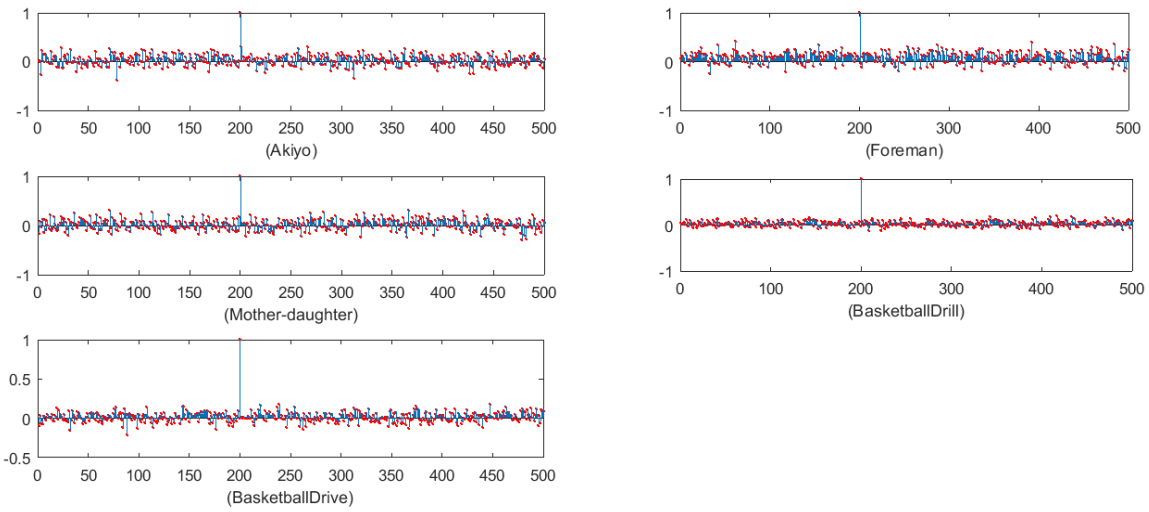


Fig. 3.43: The watermarking process response to false alarm test, the right watermark is the 200th

watermark in an object, rather than extracting the whole watermark. The watermark embedding process is repeated for at least 50 times for each test video, and under aggressive attacks, by using different watermark sequences and keys, then the receiver operating characteristics (ROC) curve is plotted [71]. This curve is a plot of the probability of true positive detection versus the probability of false positive detection [29]. The false positive detection occurs when the detector detects a watermark in an unwatermarked object, while the true positive detection happens when the detector detects the presence of a watermark in a watermarked object (which is in this case the video sequence). There are also the false and true negative detections which are not used in this research since they give the same detection results. Fig. 3.44 shows the detection performance of our method under aggressive attacks (Gaussian noise with variance 0.1, H.265 compression (QP=25), Median Filtering 3x3), and the other methods of video watermarking. Here the ROC curves are built on BER's values rather than correlations, and the average BER's for the test videos is used. It can be seen that our method is superior to other methods in the detection performance.

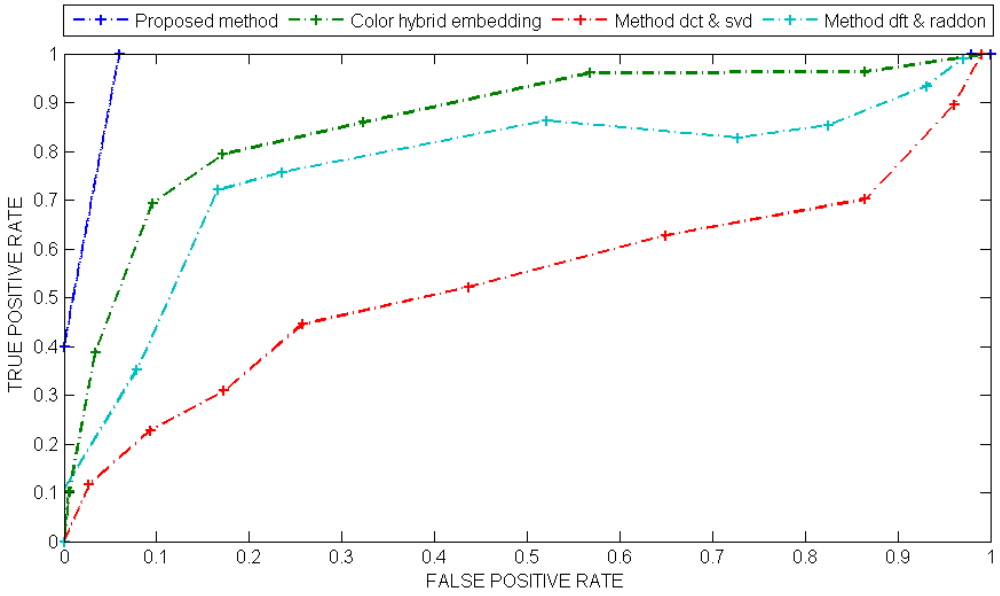


Fig. 3.44: ROC curves for our method and other methods; in our method aggressive attacks are used.

3.5.3 Computational Complexity of the Dual Hiding Process

To perform the watermarking in the pyramid and the wavelet transforms, two requirements should be met. First, the filter banks should be generated randomly; this means that the decomposition structure and the bands being used for watermarking must be determined by the owner in the hiding stage. The second requirement for practical watermarking system is to perform the hiding and the extracting processes in minimum time. Storage requirements are small; the filters can be generated by changing the coefficients only. The running time is related directly to the computational complexity of the pyramid, DWT and SVD processes.

Computational complexity for the Pyramid and DWT processes depends on the number of operations (here multiplications) required to transform an image for a number of levels N. A mathematical derivation of this complexity for pyramid transform is introduced in [59]. The derivation assumes that circular convolution based on

Fast Fourier Transform (FFT) and Inverse Fast Fourier Transform (IFFT) is used for transforming an image pyramidally. It is well known that the overall complexity of conducting convolution for B points via the FFT is: $O(BLog_2B)$ which is lower than $O(B^2)$ of the computation of B-point Discrete Fourier Transform (DFT). The derivation is summarized below for an input image x_0; the pyramid decomposition for this image was shown in Fig. 3.1.

For an image $x_0(n_1, n_2)$ of size $L_1 * L_2$, the number of multiplications needed for the first level $x_1(n_1, n_2)$ with decimation factor M will be:

$$L_1 L_2 Log_2 L_1 + \frac{L_2 L_1}{M} Log_2 L_2 \tag{3.16}$$

The first part of Equation 3.16 results from horizontal filtering and the second part is the number of multiplications needed for vertical filtering after a decimation by M. Let $K_1 = L_1 Log_2 L_1$, and $K_2 = L_2 Log_2 L_2$, then Equation 3.16 can be applied for higher levels. In general, the total number of multiplications needed to get the decimated images $x_1(n_1, n_2), x_2(n_1, n_2), ..., x_{N-1}(n_1, n_2)$ and the difference images $e_0(n_1, n_2), e_1(n_1, n_2), ..., e_{N-2}(n_1, n_2)$ can be written as follows [31]:

$$2\left[L_2 K_1 + \frac{L_1}{M} K_2\right] \qquad \text{for} \quad N = 1 \tag{3.17}$$

$$2\left[\sum_{i=0}^{N-1}\left[\frac{L_2 K_1}{M^{2i}} + \frac{L_1 K_2}{M^{2i+1}}\right] - L_1 L_2 \sum_{i=0}^{N-2}\left[(i+1)\frac{M+1}{M^{2i+3}}\right]\right] \qquad \text{for} \quad N \geq 2 \tag{3.18}$$

Where N is number of decomposition levels, M is the decimation factor which is generally 2; moreover, the hiding and extracting stages are taken into account. The above analysis can be extended to the wavelet transform taking into account that there are four filters for each stage of decompositions and four filters for each stage of reconstructions, and the decimation factor is $M = 2$. Number of multiplications in the wavelet transform is shown in Equations 3.19 and 3.20.

$$8\left[L_2K_1 + \frac{L_1}{M}K_2\right] \qquad \text{for} \quad N = 1 \tag{3.19}$$

$$8\left[\sum_{i=0}^{N-1}\left[\frac{L_2K_1}{M^{2i}} + \frac{L_1K_2}{M^{2i+1}}\right] - L_1L_2\sum_{i=0}^{N-2}\left[(i+1)\frac{M+1}{M^{2i+3}}\right]\right] \qquad \text{for} \quad N \geq 2 \tag{3.20}$$

The singular value decomposition, on the other hand, has a computational complexity of $O(min\{mn^2, nm^2\})$ [72], where n and m are the dimensions of the matrix that is being decomposed. As was shown in Section 3, the RGB color watermark was converted to its three components, the red, the green and the blue; then each component was divided to 4x4 matrices, where the SVD process is performed. According to the complexity limit of the SVD process, the computational complexity is $4 * 4^2 = 64$ for each martix. If the size of the watermark is 72x88, then the total number of multiplications in the hiding process is (72x88/16)x3x64=76032 and the same for the extraction process. The number of multiplications for the three decompositions being used, the pyramid, the DWT, and the SVD for one video frame are shown in Fig. 3.45. Moreover, the number of multiplications in Logarithmic scale are shown in Fig. 3.46 . Here the pyramid and DWT processes correspond to the main hiding process, while the SVD process correspond to the secondary hiding process. The increase in the computational complexities due to the use of the secondary hiding

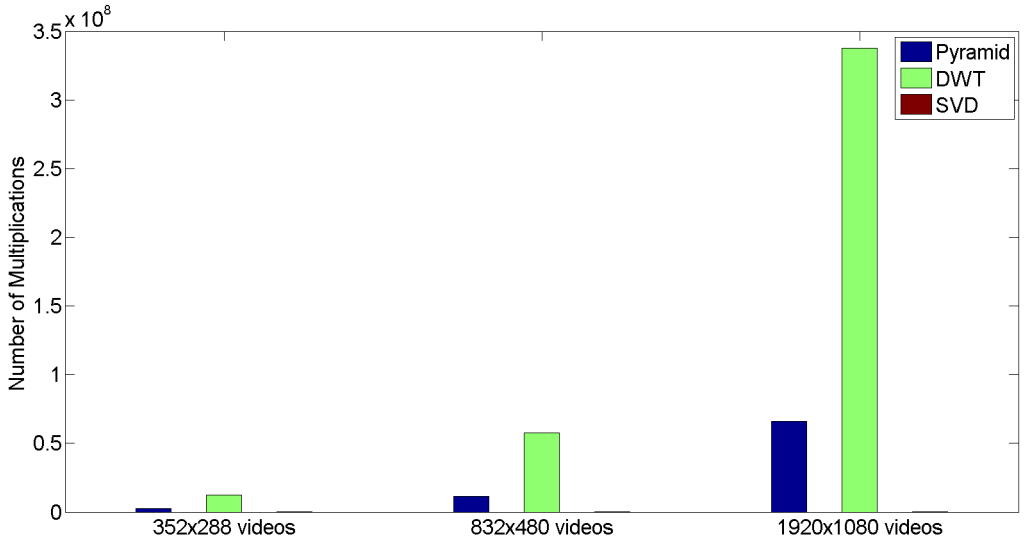


Fig. 3.45: The Number of Multiplications for the three decompositions being used, the pyramid, the DWT, and the SVD for one video frame.

process is 0.51%, 0.11% and 0.018% for the low, meduim and high resolution videos respectively. This increase is marginal given the increase in the security that was achieved due to our dual hiding process. Moreover, the computational complexities evaluated here did not include every aspect of the operations that were performed during our hiding process; we are more concerned with the limiting behavior of the operations $O(operations)$, and the increase due to the duality in our hiding process.

3.6 Noise-Removal Selective Filter

One of the challenging aspects in video encoding and watermarking is the additive noise that result in distorted video streams. The nature of the additive noise depends primarily on the source of this noise. Not only the additive noise tends to distort the visual quality of the video in question, but it also has its noticeable impacts on the watermarking process. One type of noise that is common in video processing techniques is the salt-and-pepper noise which we discussed before as a signal processing

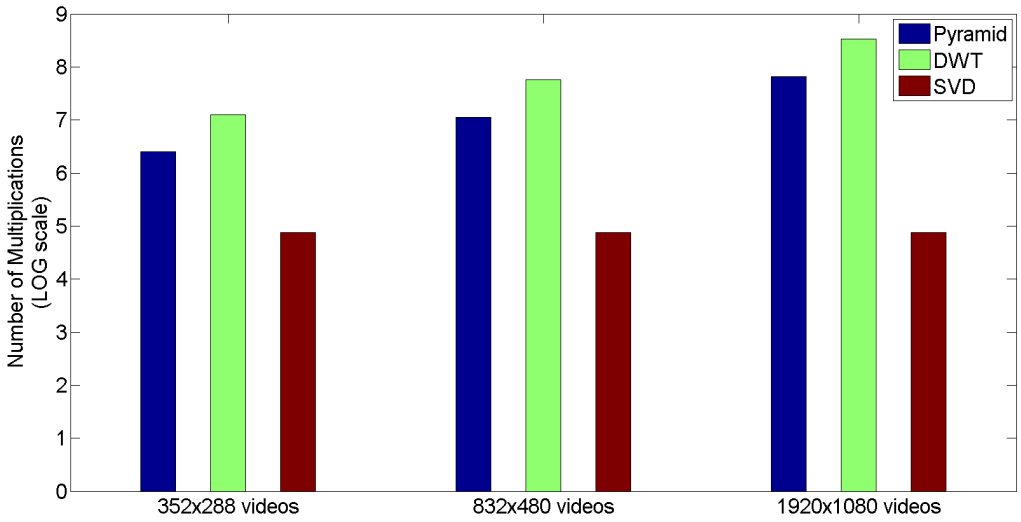


Fig. 3.46: The Number of Multiplications in Log scale for the three decompositions being used, the pyramid, the DWT, and the SVD for one video frame.

attack. This type of noise could be added to the video frames during the transmission process when the communication channels, in a sense, are noisy, or it could be a result of hardware generated errors during the encoding and decoding processes. Removing the noise without disturbing the watermarking process on one hand and preserving the visual qualities on the other hand is a challenging process.

As far as the watermarking process is concerned, it is useful to check the effects of both the additive noise and the removal process on our data hiding process. The normal median filters for example, which are used to eliminate the salt-and-pepper noise in images do in fact filter the whole image regardless of the presence or absence of the noise in a certain area. This process reduces the original resolution of the image to a great extent in such a way that the qualities of high definition (HD) videos are lost. This means that our watermarking process would not achieve the visual quality condition.

In this research, a noise detection process that depends on the absolute differences between a pixel a_{ij} and its surrounding pixels is proposed. In order to enhance the

detection process, the variance of the pixels in the surrounding window is calculated. This step is important because of false detections, especially at edged and textured details of the image where the absolute difference value could be high while the region is noise-free. This method takes into account the fact that such variances are dramatically high at these locations. However, this is not the case around noisy pixels in general where some sort of consistency is there. The proposed method for noise detection and elimination process involves the following steps:

1. For each pixel a_{ij}, a sub window of size 3x3 around this pixel is taken.

2. The absolute differences between the pixel a_{ij} and the surrounding pixels are calculated.

3. The arithmetic mean (AM) of the calculated differences for a given pixel a_{ij} is computed. The arithmetic mean (AM) is then compared with a threshold value t to detect whether the pixel a_{ij} is informative or corruptive.

4. The 3x3 pixels window is converted to an array and then it will be arranged in an ascending order. The largest and the smallest values will be eliminated. This will help in removing other noisy pixels in the surrounding window. The resulting array will be denoted L. The variance of the pixels in the array L is computed and denoted as V.

5. A comparison will be performed between AM and V, and their respective thresholds:

   (a) If AM is greater than t and V is less than the variance threshold, then do the elimination process by replacing the noisy pixel by the median of the surrounding pixels in the window.

(b) Otherwise, do nothing. In this case, either there is no noise, or the pixel in question is on one of the edges of the image, and nothing should be done accordingly.

The arithmetic mean (AM) threshold is a user defined-value between the minimum and maximum pixel values (0,255) which is used to distinguish an informative pixel from a noisy one. On the other hand, the variance V can take larger values, and its threshold value can be determined accordingly. In fact, its value depends on the images themselves whether they were textured or smooth ones. Original frame of Foreman, the same frame after being corrupted with salt-and-pepper (S&P) noise of 1% density and the same frame after the denoising process are shown in Fig. 3.47. Moreover, the same frame after being denoised using median filter is shown in the Figure. It can be seen that our selectivity in noise removal preserved the visual quality to some extent unlike the traditional median filter. Fig. 3.48 shows the peak signal-to-noise-ratio (PSNR) values of the noisy and denoised versions of the standard videos: Akiyo, Foreman, Mother-daughter, BasketballDrill and BasketballDrive.

Moreover, the performance of the watermarking process with the use of the selective denoising filter was examined. Table. 3.12 shows the performance of our system; the videos were attacked with salt-and-pepper (S&P) noise of 2% density, then our selective denoising filter was used to clean these watermarked videos; later on, the two watermarks, the color and the B&W watermarks were extracted. The first frames of the denoised videos are shown in the table as well as the extracted watermarks.

Fig. 3.47: (a) Original Foreman frame, (b) the same frame after adding salt-and-pepper (S&P) noise of 1% density (c) the denoised frame using our selective filter, (d) the denoised frame using median filter.

Table 3.12: BER's of our method with the use of the selective filter

Attack	**Four Averaged Frames**	**BER1**	**Extracted Watermark**	**BER2**	**Extracted Watermark**
Akiyo		0.019		0.040	
Foreman		0.024		0.053	
Mother-daughter		0.018		0.050	
Basketball-Drill		0.010		0.045	
Basketball-Drive		0.014		0.059	

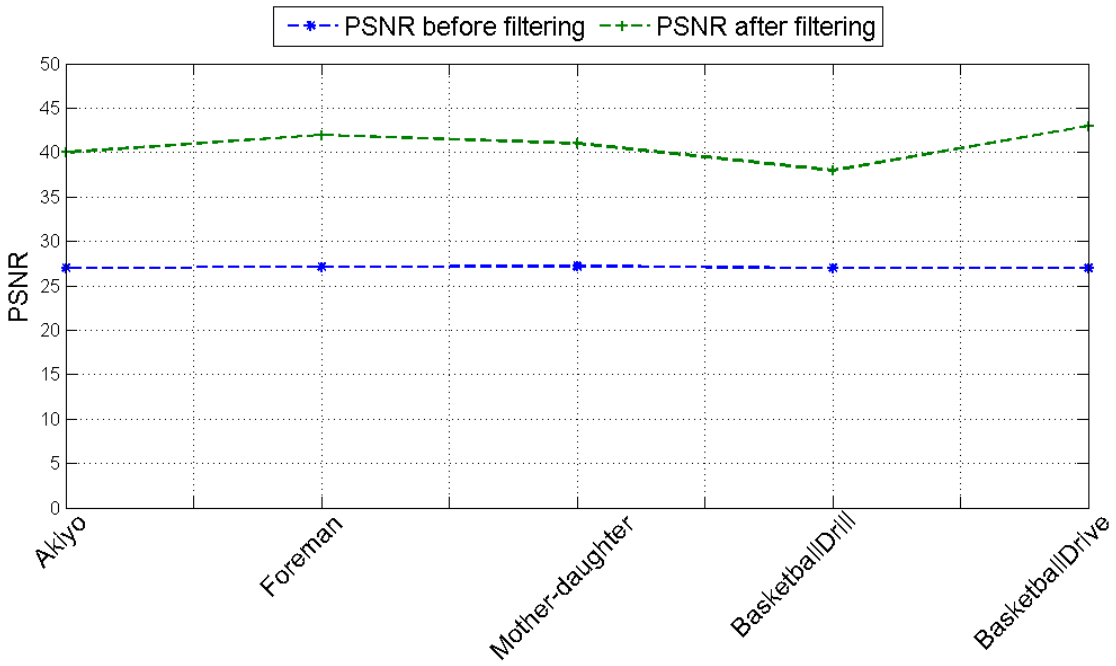


Fig. 3.48: PSNR's of standard videos before and after denoising process

Chapter 4

3D Mesh watermarking

4.1 Introduction

Digital watermarking has evolved in the past years as an important mean for data authentication and ownership protection. The images and video watermarking is well known in the field of multimedia processing. However, 3D objects watermarking techniques have emerged as important mean for the same purposes, as 3D mesh models are in increasing use in different areas of virtual reality, medical imaging, video games and computer aided design [73].

A mesh is a collection of polygonal facets that approximate the 3D objects in real life. It can be described by many elements; the main elements that describe a mesh are the vertices, the faces, and the edges that connect the vertices. Other elements that are used also are the normals to the faces. The elements that are used to describe a mesh depend mainly on the application that the mesh is used in; for instance, the faces' normals are often used in computer graphics to determine the shading in the mesh which reflects the directions with respect to the light. This is extremely important to

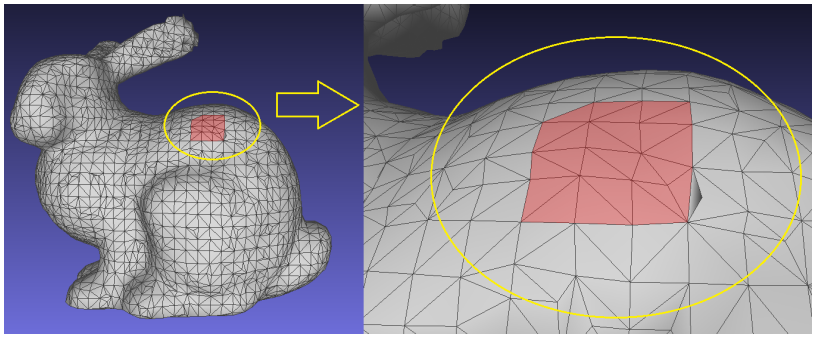


Fig. 4.1: Bunny mesh with a closeup of the vertices and faces.

give the real life human perceptual sense of the 3D object. Fig. 4.1 shows a closeup of the vertices and faces of a down-sampled mesh called Bunny; here the faces' polygons are triangles which is the most popular case.

Moreover, other information that the mesh have are: the geometry information which describes the location of the different vertices in the space, and the connectivity information which describes the connectivity between these points or simply the edges. This will be illustrated more in the watermarking process attacks where the attacks are categorized in different areas depending on the source and the part of the meshes that are most affected.

The watermarking of 3D objects can take place in either space or transform domains, which is similar to image watermarking techniques; moreover, unlike image and video watermarking where the frames have regular structures in both space and temporal domains, 3D objects are represented in different ways as meshes that are basically irregular samplings of surfaces [74]. Moreover, meshes can undergo a large variety of alterations which may be hard to tackle [75]. This makes the watermarking process more challenging. Several 3D watermarking schemes and methods were developed over the past years, they can be classified as blind or non-blind, robust or fragile [73]. Blind watermarking process means that the original object is not required in the course of extracting the hidden data; while non-blind watermarking techniques

means that the original host object is required in the extracting process.

The meaning of robustness in 3D watermarking is no different from its meaning in digital image watermarking. For the watermarking technique to be robust, the hidden data should be extracted with the lowest errors rates under different circumstances; furthermore, the visual and metric qualities of the mesh should remain within specific limits [76, 77, 78]; specific benchmarks to assess the watermarking processes were proposed [71]. While the transform domain watermarking is preferable in images and videos, they are still difficult to implement in 3D meshes due to the huge number of vertices involved and the complicated topology and geometry, and hence the difficulty to perform the spectral decomposition even though significant work was done in the field [75, 79, 80]. Transform domain watermarking can be divided to: Direct frequency analysis such as the Laplacian transform and Multi resolution analysis such as the wavelet transform. The main problems with the Laplacian decomposition are focused in two points; first, the computation time increases rapidly with mesh complexity, and this forced the process of mesh segmentation, i.e, dividing the mesh to several regions and performing the decomposition process after that; and this in turn necessitated the process of performing mesh registration to do a resampling at the extraction process; this means that the watermarking process is no longer blind. These problems are faced in most transform domain mesh watermarking techniques.

Spatial domain watermarking have attracted significant attention in the past years; they can either act on the topology or on the geometry of the model [81]. Moreover, they can be categorized as deterministic or statistical [82, 83]. Exploiting the statistical characteristics in the 3D mesh models from both geometrical and topological aspects was useful in hiding data. However, doing that with minimal surface distortions to the mesh attracted significant research in the field [84, 85]. An oblivious watermarking method for 3D mesh models, which alters statistically the distances

between all vertices and the center of mass of the mesh was proposed in [83]. Significant work was done to optimize the statistical methods. These methods are mostly concerned with: optimal placement of the vertices, the causality issue, and the imperceptibility or surface distortions issue [74, 86]. In this research, a blind optimized method that further exploits the statistical characteristics of the 3D mesh models and hides the watermark bits in them is developed. The hiding process aims at introducing the minimal surface distortion, enhancing the extraction process, and reducing the computational complexity of the system. Several tests will be performed to demonstrate the robustness and efficiency of the technique.

4.2 The proposed method

In this section, the method of 3D mesh watermarking is introduced. Our watermarking method depends on modifying the vertices positions with respect to the center of the object. An optimal method will be developed to reduce the errors, minimizing the distortions that the 3D object may experience due to the watermarking process, and reducing the computational complexity due to the iterations and other factors. First of all, the Cartesian coordinates of the vertex $v_i(x_i, y_i, z_i)$ are converted to spherical coordinates (ρ, θ, ϕ). The center of the mesh $\mathbf{o}$ can be evaluated by many methods, one of them is [82]:

$$\mathbf{o} = \frac{\sum_{v_j \in O} A(v_j) v_j}{\sum_{v_j \in O} A(v_j)} \tag{4.1}$$

where $A(v_j)$ represents the area of all triangular faces of the mesh object O containing the vertex v_j. Another way to compute the center is to use the volume and surface

body moments [87]. Vertex norms p_i's then are evaluated; then, they are divided into N distinct bins according to their magnitude, each bin will be used to hide one watermark (W) bit. The maximum and minimum vertices norms will be determined p_{max}, p_{min}; the difference will be divided into N different bins B_n's where each bin is defined as follows [83]:

$$p_{n,min} = p_{min} + ((p_{max} - p_{min})/N).n \tag{4.2}$$

$$p_{n,max} = p_{min} + ((p_{max} - p_{min})/N).(n+1) \tag{4.3}$$

where $p_{n,min}$ and $p_{n,max}$ are the lower and upper limits of the nth bin. The angles θ, ϕ for each vertex in the spherical coordinates are also needed to convert back to the Cartesian coordinates later on. The next step will be to normalize each element $p_{n,j}$ in the bin n. These normalized vertex norms can be modified in two methods of histogram mapping functions [83]. Assuming that the elements in each bin are uniformly distributed, the first histogram mapping function changes the mean value of the statistical variables in each bin that correspond to the vertices, while the other method changes the variance. In this research we will use the second method which changes the variance; the mapping here will be to the range [-1 +1]. The expected variance of the uniformly distributed variables is 1/3, so the variance ${\sigma_n}^2$ will be

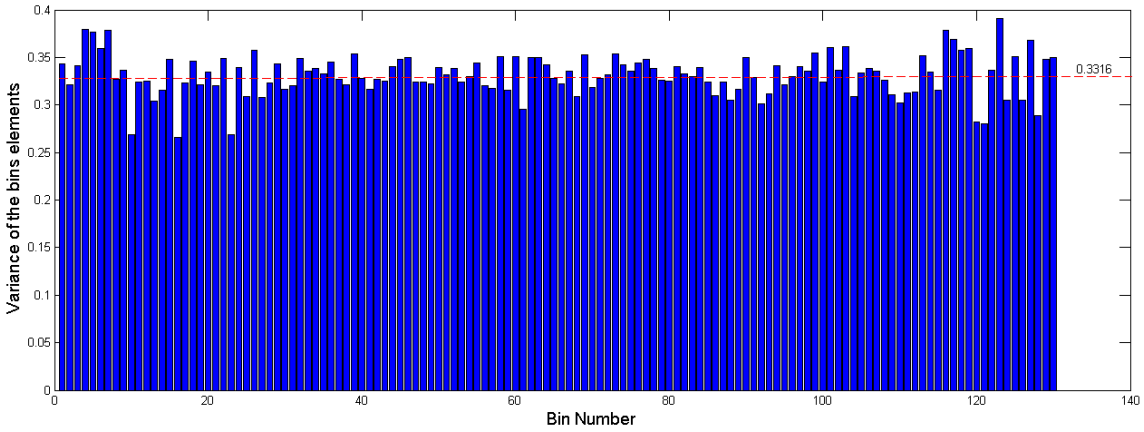


Fig. 4.2: Variances of the bins elements of the Bunny mesh

modified according to:

$$\sigma_n'^{\,2} = \begin{cases} 1/3 + \alpha, & if \quad B_n = +1 \\ 1/3 - \alpha, & if \quad B_n = -1 \end{cases} \tag{4.4}$$

Where α represents the shift value. To illustrate the uniform statistical distribution and the degree of this uniformity; Bunny mesh norms (34835 vertices) were distributed to 130 bins according to the previous analysis, then the variance of each bin was evaluated. Fig. 4.2 shows that the variances tend to be 1/3 (actually the average variance was 0.3316).

In fact a power function will be used to evaluate the modified vertices [83], where the normalized norms $p_{n,j}$ are modified according to this equation:

$$p_{n,j} = sign(p_{n,j}).|p_{n,j}|^{k_n} \tag{4.5}$$

where n represents the bin number and j is the vertex index. The power function

was used to restrict the vertices normalized norms to remain in the range [-1,+1]. Furthermore, the iteration process proposed in [83] can be performed with specific steps to get the final modified vertices's norms. This will be performed in combination with the optimization process proposed later on in this chapter. The extraction process of the watermark bits w'_n's is quite simple; the variance of each bin (σ''^2_n) is calculated and compared using this relation:

$$w'_n = \begin{cases} +1 \quad if \quad \sigma''^2_n > 1/3 \\ -1 \quad if \quad \sigma''^2_n < 1/3 \end{cases} \tag{4.6}$$

The hiding process using the VARIANCE method is shown in Fig. 4.3.

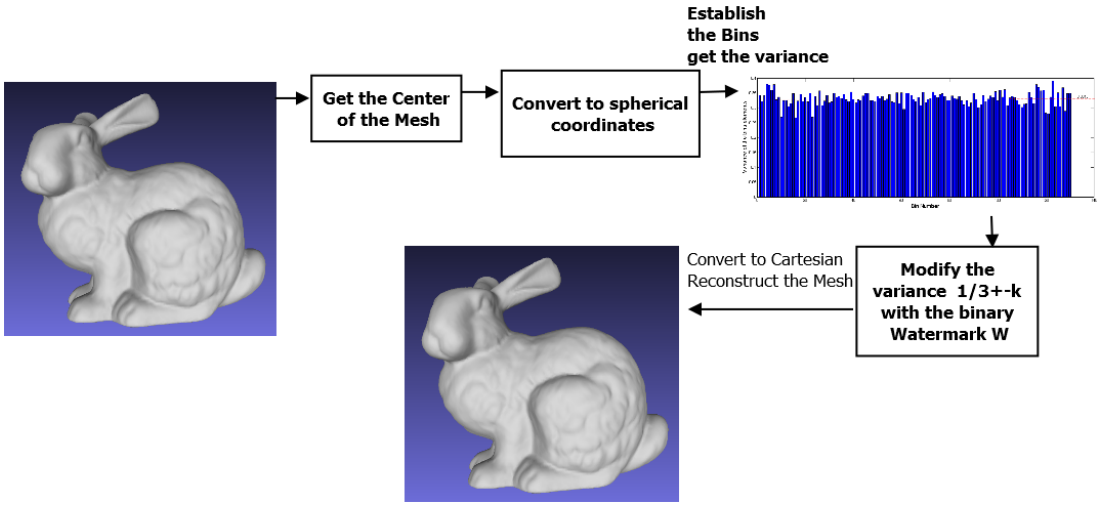


Fig. 4.3: 3D mesh watermark hiding process.

4.2.1 Optimization Process

In this work, statistical analysis was done to establish the relations between the number of elements in each bin, the variance values of the bins, the variances of the variance values of the bins, the width of the bins and the errors rates that resulted. In [86], several constraints were put in place to minimize the distortions and enhance the

detection process, and the problem was converted to an optimization problem that relies on quadratic programming. The optimization in our case can be statistically established in another way; to illustrate that, and using Bunny mesh (34835 vertices); first, the interval [p_{min},p_{max}] was divided to 400 equal segments, and hence 400 bins. Then the variances of these bins were evaluated; and a watermark binary sequence of 400 bits was hidden in the 3D mesh using method II [83] which shifts the variance of the bins elements. The 400 bins were grouped in 40 groups to evaluate the variances of variance values in these groups (V) as well as the numbers of errors in them (E). Furthermore, (B'') which corresponds to the mean of the bins sizes divided by their actual sizes was computed. Then, the cross correlations between (E) on one side, and (V) and (B'') on the other side were computed; this is shown in Fig. 4.4, it can be seen that these parameters are well correlated .

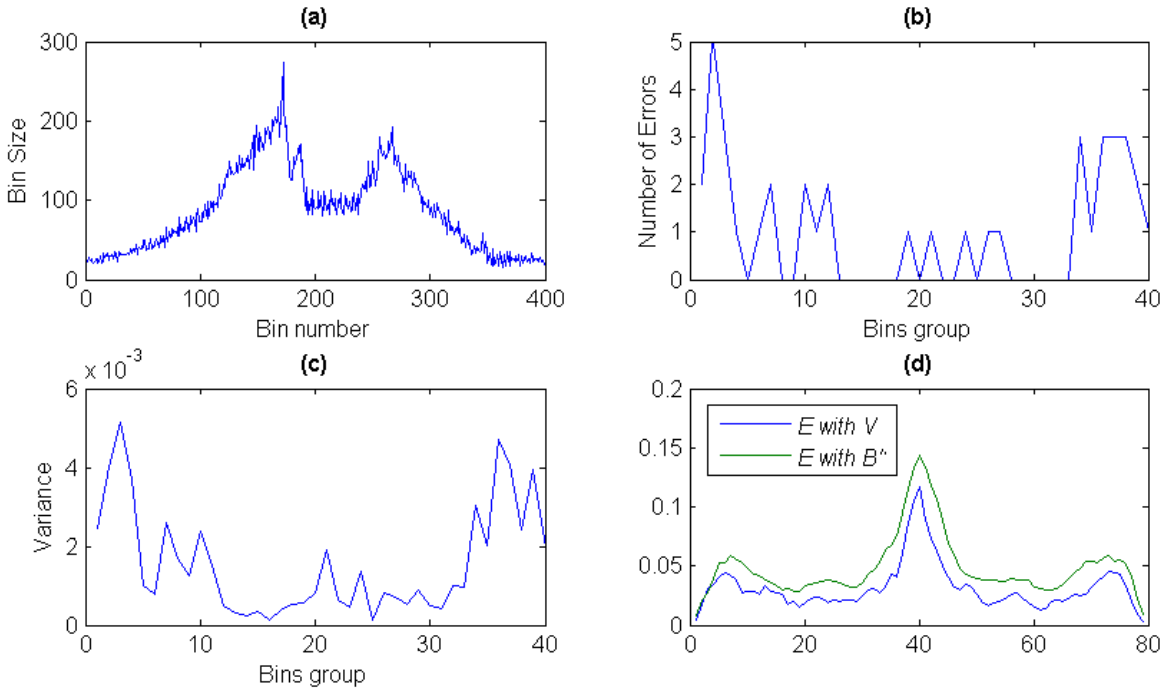


Fig. 4.4: (a) Bins Sizes, (b) Errors in bins groups (E), (c) Variances of the variance groups (V), (d) Cross correlations (E with V, E with B''),

It can be seen in Fig. 4.4(a) that the distribution of the vertices in the bins is not equal; upper and lower limits have fewer elements than other regions. Experimental

results showed that most of the 3D mesh objects behave in this manner. Getting approximate equal numbers of elements in each bin can enhance and smoothen the variance values, and hence the bins elements can behave in a more uniformly distributed manner. Dividing the interval $[p_{min}, p_{max}]$ to segments with widths that are proportional to B'' can solve part of the optimization problem, the resulting weights are denoted K_d. Fig. 4.4(a) shows that the relation between the number of elements in each bin and the widths of the bins is not linear. Since the nearby bins are correlated, then, taking each five nearby bins as a group and applying the same weight to them can enhance the performance. To deal with the non-linearity issue and prevent unrequired spikes in values, a power function can be applied for a watermark of size M, where:

$$K'_d = (K_d)^q \tag{4.7}$$

where $q \in [0.1,0.6]$ under constraint $Sum(K'_d)=M/5$, and K'_d is the new bin weight; this would result in the final weighting coefficients of the intervals. To illustrate this relation, the mesh objects were watermarked using 400 bits, and this is a relatively big watermark for 3D meshes; then different power function factors (q) for establishing the intervals of the bins were used. Fig. 4.5 shows the BER's values, it can be seen that the uniform devision is not always optimal; moreover, the best values of q could be anywhere between 0.2 and 0.5. Using the Bunny mesh and hiding 200 bits; the original weighting factors, the enhanced weighting factors K_d and the final normalized weighting factors K'_d are shown in Fig. 4.6. Here each nearby 5 bins are given the same weighting due to the aforementioned correlation fact.

The weighting process as well was applied on standard meshes: Cow, Casting, Bunny

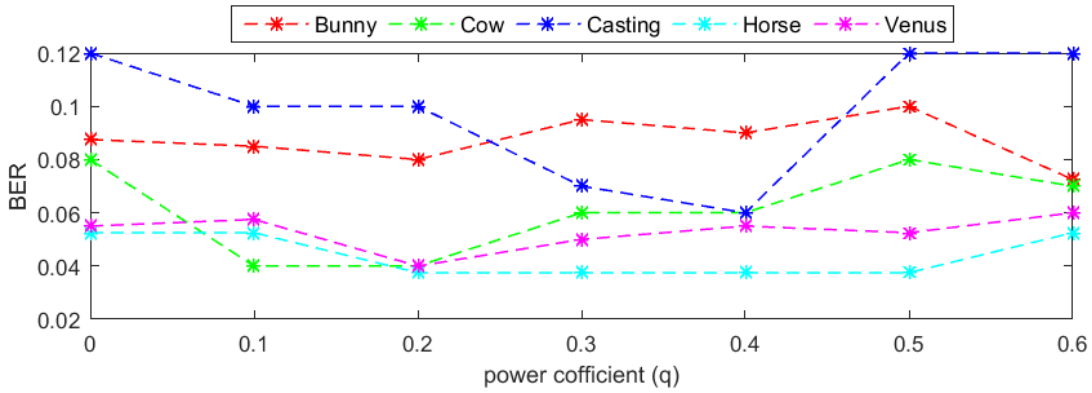


Fig. 4.5: The BER's vs. the power coefficent value q for different Mesh objects, (k_n for Equation 4.5 small, watermark size: 400 bits, no iterations)

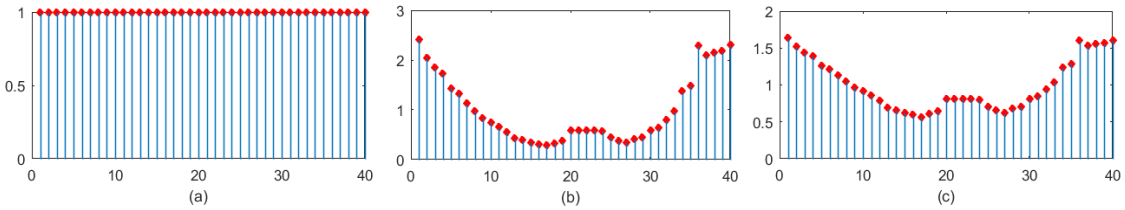


Fig. 4.6: The bins weighting factors, (a) uniform (b) using K_d, (c) using K'_d

and Dragon. The size of the watermark was 65 bits for the first two meshes and 130 for the latter two meshes. Fig. 4.7 shows the variances of the bins before the weighting process–*i.e.*,the uniform divisions of the bins and after using our weighting process for the standard meshes; the red line represents the ideal variance which is 1/3. To show the enhancement in the variances of the bins, the average variance of the bins were evaluated for the same meshes before and after the weighting process; Fig. 4.8 shows the results where the pink line represents the ideal variance. Moreover, the variances in these bins variance groups were evaluated; Fig. 4.9 shows the changes and the modifications that we've got as a result of using our power weighting function of the bins.

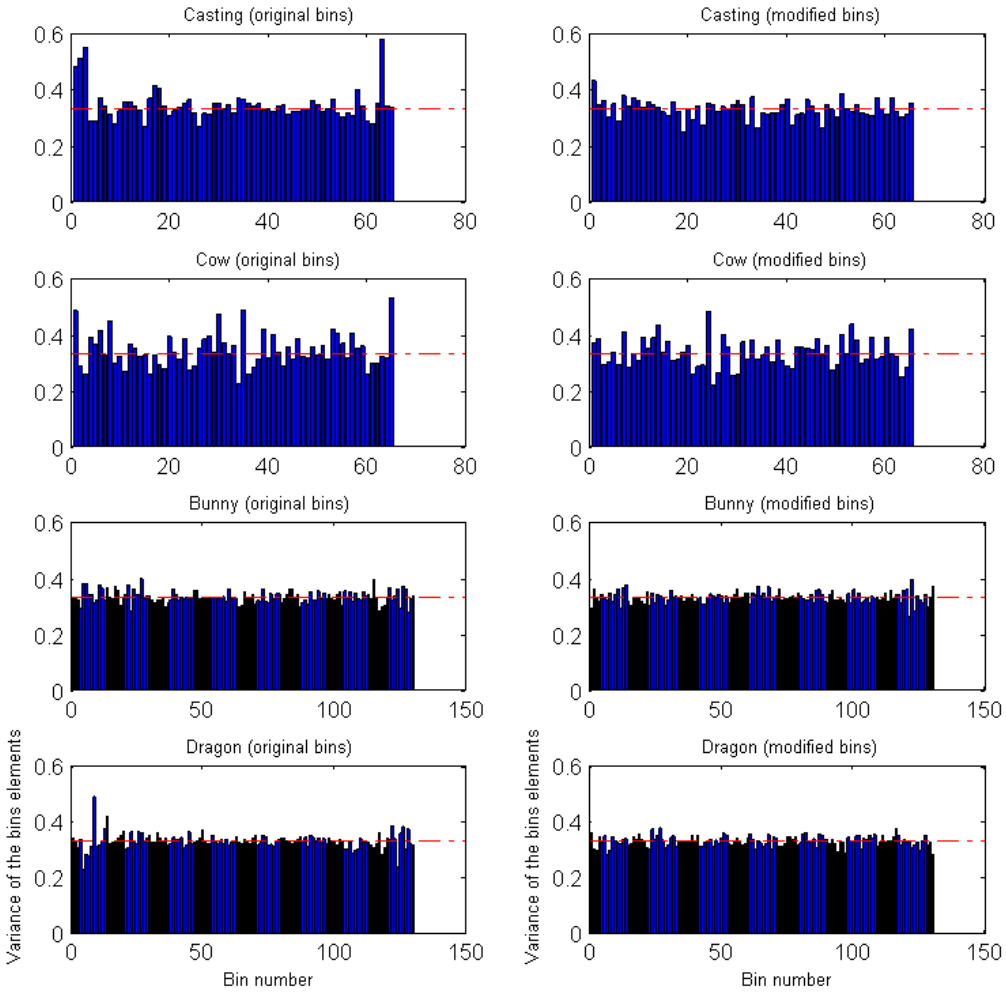


Fig. 4.7: Variances of the bins before and after applying our weighing function.

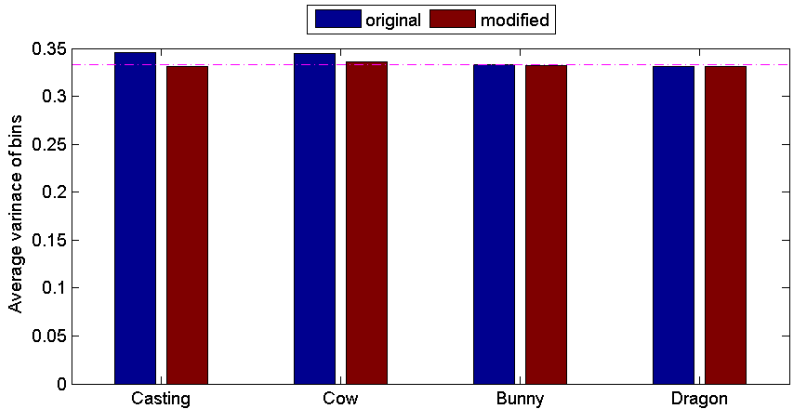


Fig. 4.8: Average variance of the bins before and after applying our weighing function.

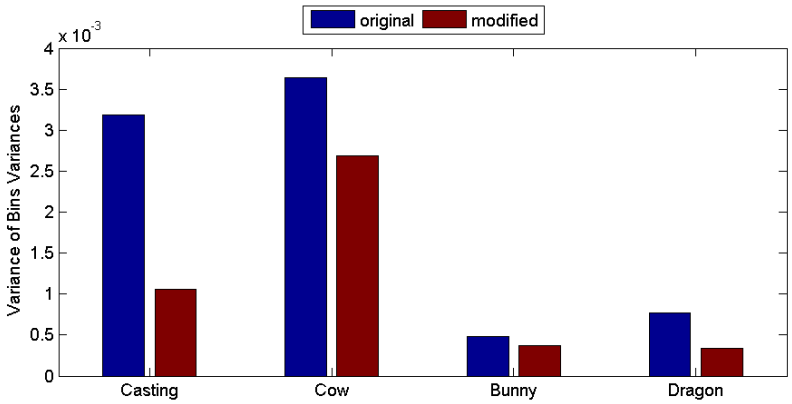


Fig. 4.9: Variances of the bins variance groups before and after applying our weighting function

4.2.2 Minimizing the Distortions

Due to the vast areas that 3D meshes can be used in, then visual qualities as well as geometric distortions should be taken into account depending on the application. In the entertainment fields for instance special interest is payed to the perceptual visual qualities. Using the Local Roughness (LR) approach proposed in [88], the mesh can be divided into regions according to the roughness in the surface. To preserve the surface visual quality, smaller weighting factors according to Equation 4.5 are applied to the smoother areas and higher weighting factors are applied to the rough surface areas. Fig. 4.10 shows the surface local roughness of several standard 3D objects where hot colors represent higher roughness. The roughness from high to low is represented in colors as [Red, Yellow, Green, Blue]. For simplicity, it is possible to use the the coefficients [1, 0.80, 0.60, 0.50] of the original weighting factors with respect to the aforementioned colors and consequently the levels of roughness.

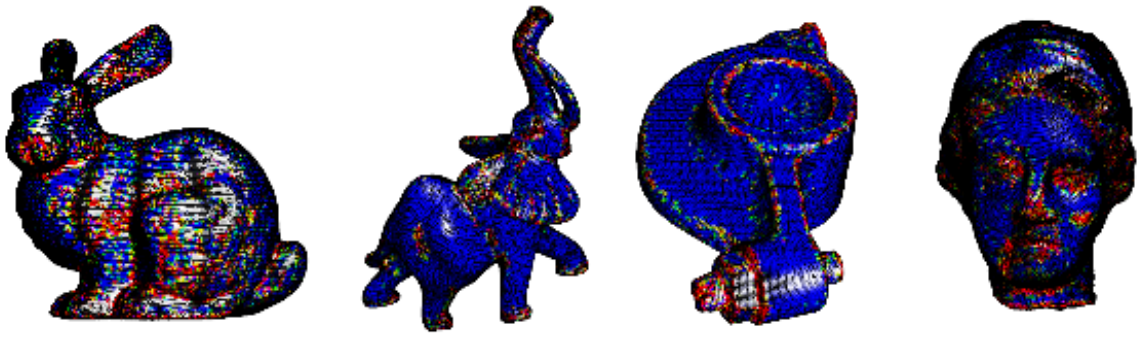


Fig. 4.10: The local roughness of some mesh objects: Bunny, Elephant, RockerArm, Venus.

4.2.3 Effect of the Center Shift

The center of the 3D mesh can be evaluated in many ways, and it has many meanings depending on the way of evaluation and the application. The centroid and the center of mass for instance can be considered centers of the 3D mesh object; one is a geometric center and the other is an arithmetic weighted mean. In this research, the center of the mesh object O was evaluated using the method in [82] and it was shown in Equation 4.1. This method is more efficient than just taking the average of the vertices.

When applying specific attacks on the watermarked mesh, it was found that a shift in the center will result. Furthermore, it was found that the shifts were oriented in the same direction for a certain mesh and a specific attack and constitute a cluster of points. Fig. 4.11 shows the shifts in the center when applying Laplacian Smoothing (Deformation factor λ=0.03) and Simplification attacks on Dragon Mesh; the first attack is a geometry attack that preserves the initial number of vertices but modifies their locations, while the other is a connectivity attack that destroys edges and hence the final number of vertices are reduced. The simplification algorithm used in this research uses Lindstrom-Turk implementation from CGAL and the canonical vertex removal algorithm [89]. The new centers are shown for each case when the attacks

were performed multiple times; 10, 30, 50 correspond to the number of iterations in the case of Laplacian smoothing and the simplification ratios in the case of mesh simplification. It can be noticed that the shifts in the case of the simplification process was larger; moreover the deviation in each group was larger. Furthermore, The points were more split when the simplification ratio was increased.

Mahalanobis distance as a mean of cluster analysis is used to measure the distance between the original center and the new centers, where each new group of centers corresponds to a specific attack. The Mahalanobis distance is a measure of the distance between a point P and a distribution D. The Mahalanobis distance of an observation $\vec{x} = (x_1, x_2, x_3..., x_N)^T$ from a set of observations with mean $\vec{u} = (u_1, u_2, u_3..., u_N)^T$ and covariance matrix S is defined as [90]:

$$D_M(\vec{x}) = \sqrt{(\vec{x} - \vec{u})^T S^{-1} (\vec{x} - \vec{u})} \tag{4.8}$$

Larger Mahalanobis distances represent higher dissimilarity and deviation. From Table. 4.1, it can be seen that the Mahalanobis distances are large; this means higher correlation between the observed samples and higher dissimilarity with the old center. Moreover, Table. 4.2 shows the means and variances of the Mahalanobis distances inside each group of centers; it can be established when comparing with Table. 4.1 that the original center is an outlier of the other groups and the high correlation between the elements in each group. It can be noticed, however, that the Mahalanobis distance does not match the Euclidean distance always; for instance in the case of the 50% simplification, the points seem to be more split even though the mean Mahalanobis distance was smaller for this group as shown in Table. 4.2.

In fact, preserving the original location of the center was found to be highly helpful in extracting the watermark. To establish the new center location, multiple attacks were

Table 4.1: Mahalanobis distance between original center and new centers as a result of Laplacian Smoothing and Simplification Attacks

Smoothing Attack		Simplification Attack	
Iterations	Mahalanobis Distance	Simp. ratio	Mahalanobis Distance
10	$3.47 * 10^4$	10%	$1.46 * 10^3$
30	$2.88 * 10^5$	30%	$2.48 * 10^3$
50	$6.14 * 10^5$	50%	$1.49 * 10^3$

Table 4.2: Mahalanobis distances within each group of centers when Laplacian Smoothing and Simplification Attacks are applied

Mahalanobis Distance: Laplacian Smoothing Attack			Mahalanobis Distance: Simplification Attack		
Iterations	Mean	Variance	Simp. ratio	Mean	Variance
10	3.87	17.15	10%	3.80	19.69
30	3.84	15.90	30%	3.70	16.03
50	3.86	16.70	50%	3.54	7.30

performed on the 3D mesh and a library of the averaged shifts experienced according to each attack was built. By using this new look-up library table, it was possible to compensate for the shifts and enhance the robustness of the algorithm against many attacks especially the simplification and smoothing attacks. The percentage enhancement in BER's for Dragon mesh case is shown in Fig. 4.12, using the aforementioned process of center shift compensation.

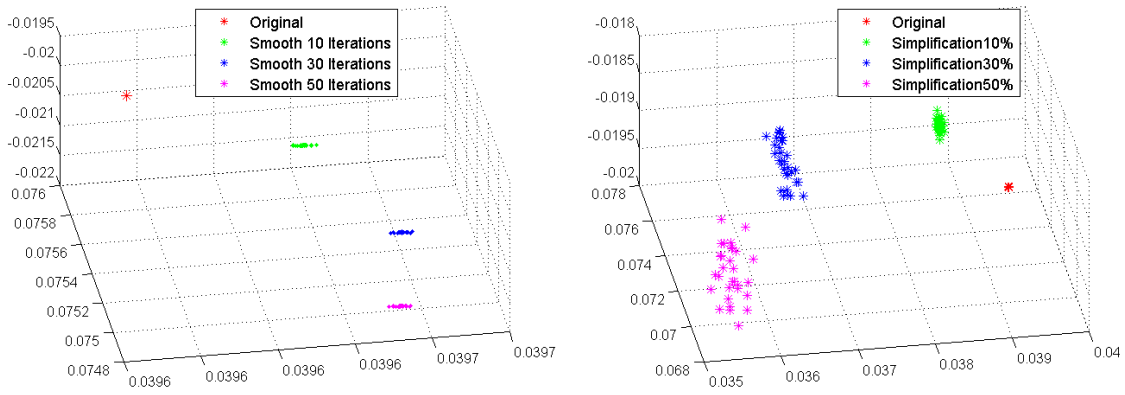


Fig. 4.11: The shifts in the Dragon mesh centers in two cases, Left: Laplacian Smoothing, Right: Mesh Simplification.

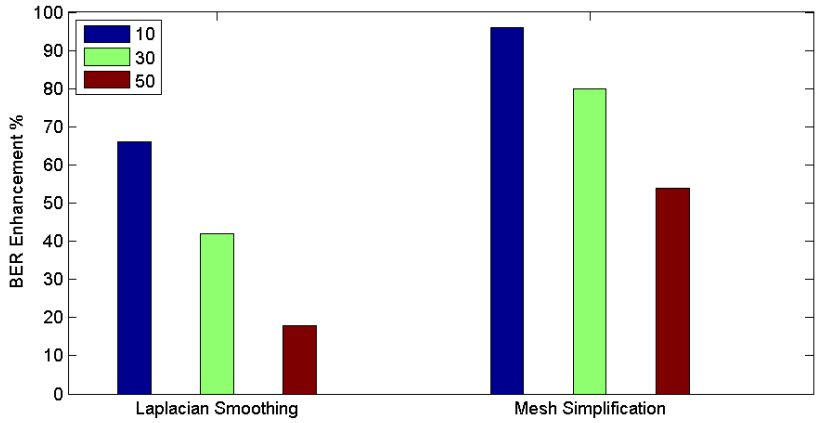


Fig. 4.12: BER enhancement when applying center shift compensation process for Dragon mesh; 10, 30, 50 correspond to iterations or simplification ratios.

4.3 Experimental results

In this section, the performance of our data hiding approach is illustrated. Two protocols for the evaluation of robust mesh watermarking schemes are defined [71]; the first protocol is called *perceptual-quality-oriented* and the second one is called *geometric-quality-oriented*. Several meshes were used from [71]; they are: Bunny (34835 vertices), Venus (100759 vertices), Horse (112642 vertices), Dragon (50000 vertices), Cow (2904 vertices), Casting (5096 vertices). The length of the watermarks were: 64 bits for cow and casting, 128 bits for Bunny and Dragon and 256 bits for Horse and Venus. Figures 4.13, 4.14, 4.15, 4.16, 4.17 and 4.18 show the original and watermarked models. The visual qualities of the original and watermarked meshes can be seen in a more obvious way. Furthermore, Fig. 4.19 shows the distribution of the difference in quality using Hausdorff distance allover the Bunny mesh as a result of our watermarking process; blue color represents higher quality while red color represent lower quality. It can be seen that our watermarking process took place allover the targeted mesh; this is useful especially against some attacks like cropping and simplification attacks; besides that, the histogram of the errors evaluated between the two meshes is shown, it can be seen that the minimum error is zero while the maximum error or distance is 0.000136; here the distance used is the actual distance not the relative distance with respect to the bounding box of the mesh, which is used sometimes.

As in the images watermarking techniques, human eye or the subjective tests are not the only factor in determining the efficiency of adopted technique. Specific objective tests that are adopted in the field of mesh watermarking were used. Table. 4.3 shows the objective test results using the MSDM and MSDM2 (Mesh Structural Distortion Measure) and MRMS (Maximum Root Mean Square Error). The human

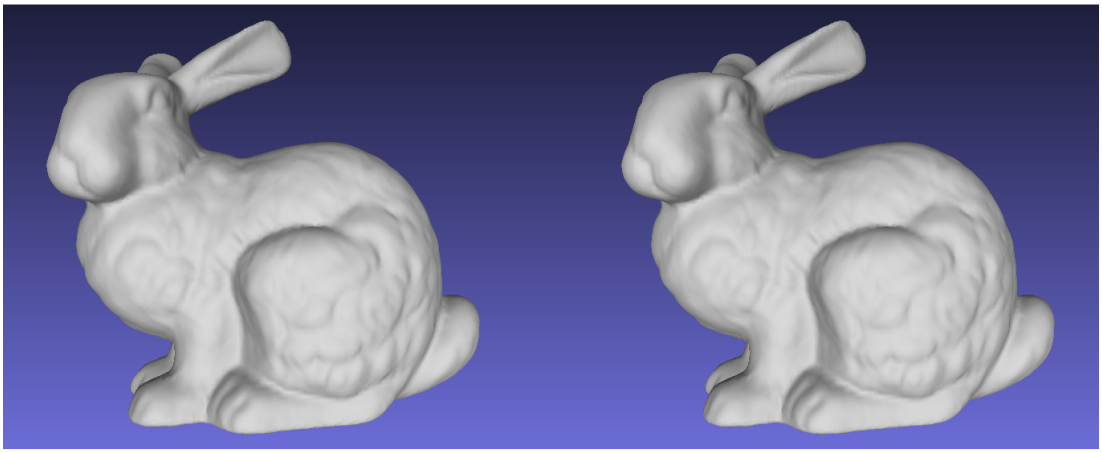


Fig. 4.13: Original Bunny on left, Watermarked Bunny on right.

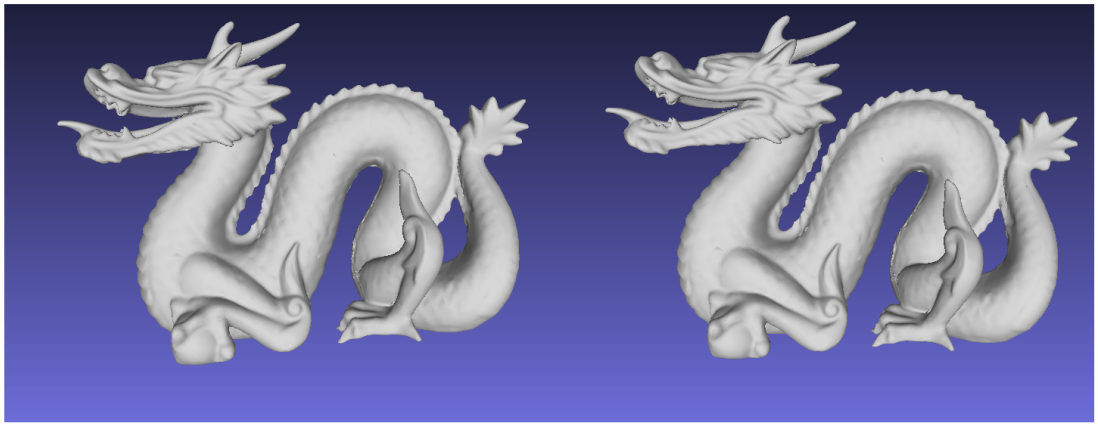


Fig. 4.14: Original Dragon on left, Watermarked Dragon on right.

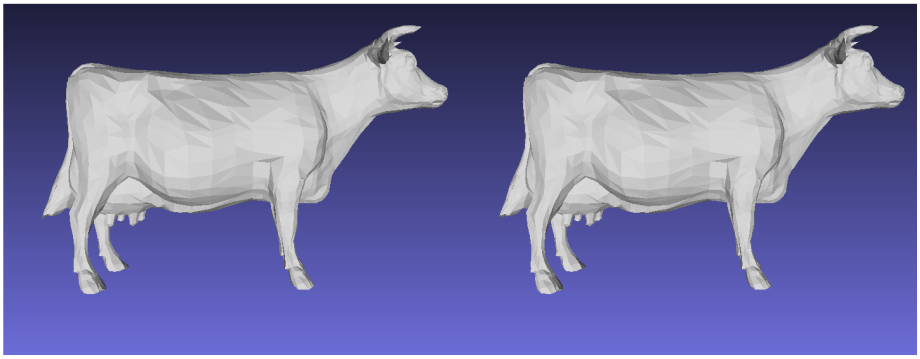


Fig. 4.15: Original Cow on left, Watermarked Cow on right.

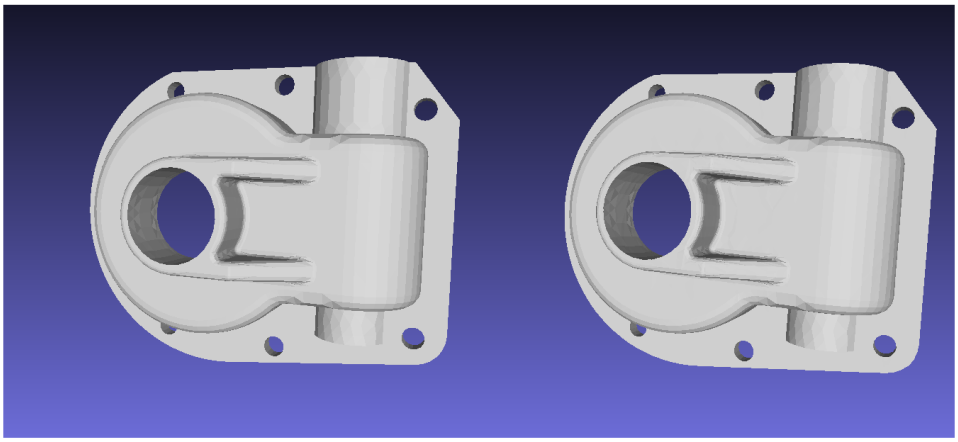


Fig. 4.16: Original Casting on left, Watermarked Casting on right.

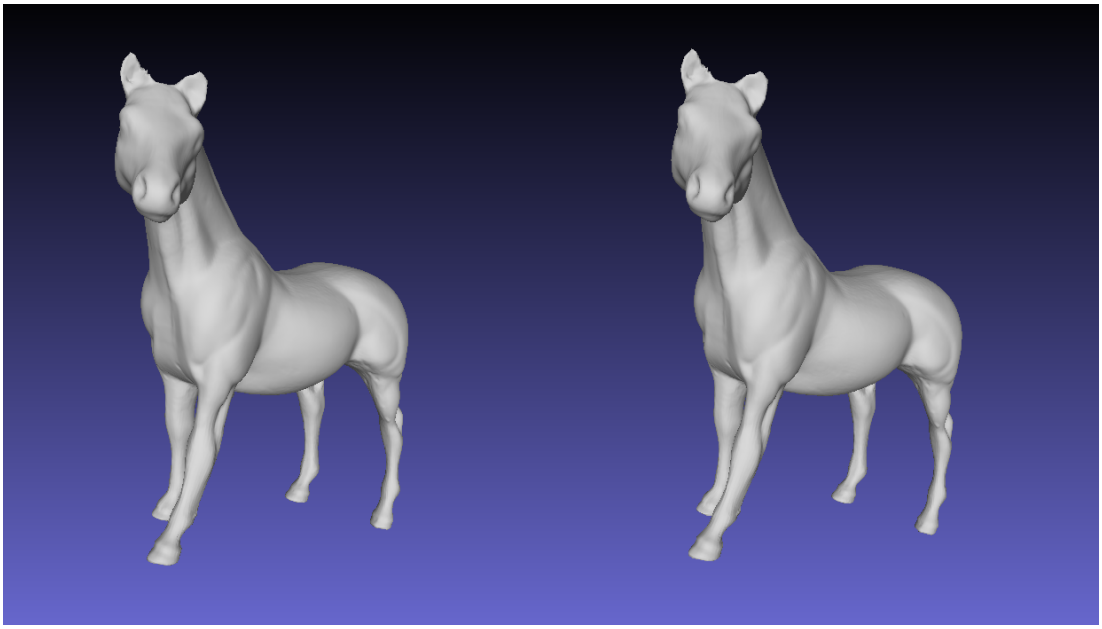


Fig. 4.17: Original Horse on left, Watermarked Horse on right.

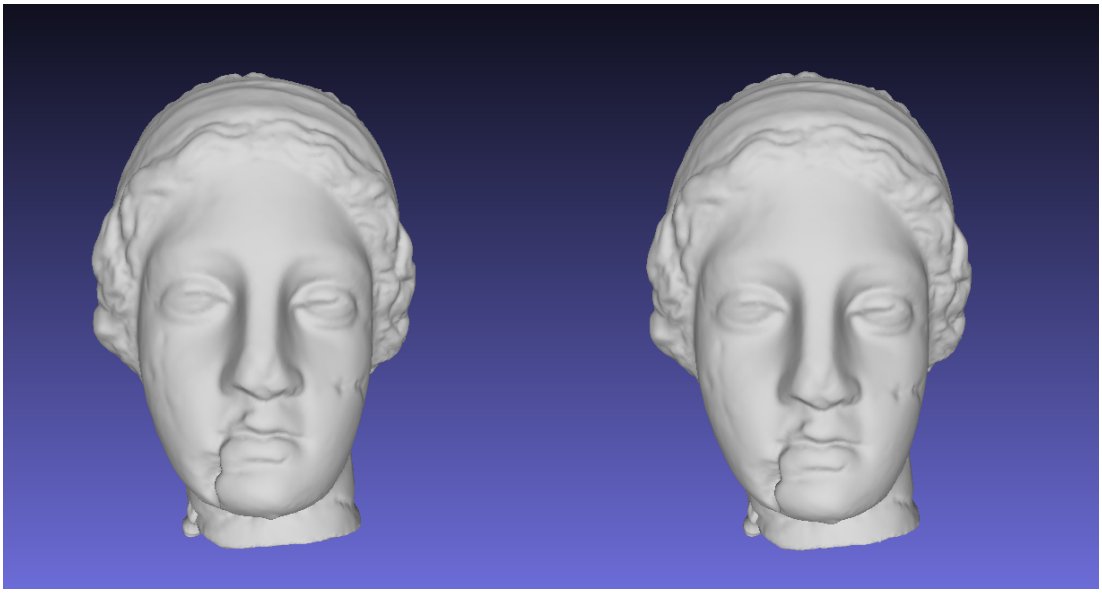


Fig. 4.18: Original Venus on left, Watermarked Venus on right.

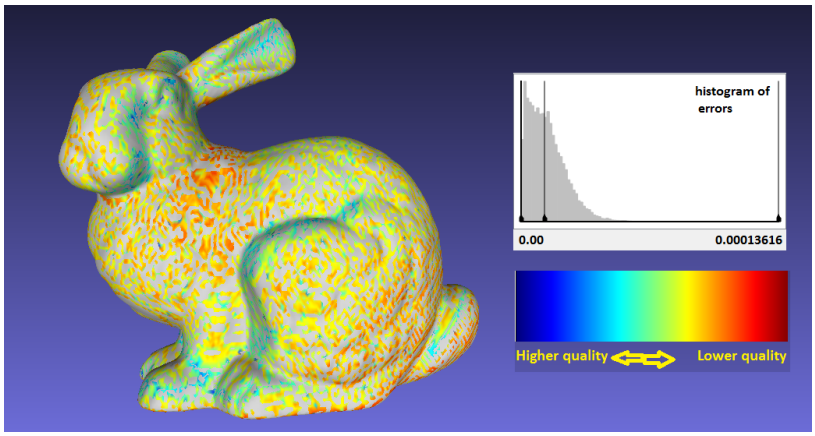


Fig. 4.19: Quality color mapped version of the watermarked bunny mesh, colors represent quality, histogram represents errors.

Table 4.3: Perceptual and Geometric quality of the watermarked Meshes

Model	MSDM		MSDM2		MRMS($*10^{-4}$)	
	Our	Li [85]	Our	Li [85]	Our	Li [85]
Cow	0.135	0.14	0.10	0.10	2.17	4.01
Casting	0.187	0.19	0.158	0.17	1.51	3.85
Bunny	0.159	0.16	0.120	0.12	0.64	1.46
Dragon	0.134	0.15	0.06	0.07	0.96	1.23
Horse	0.170	0.17	0.188	0.18	0.26	0.71
Venus	0.119	0.12	0.098	0.10	0.27	0.64

Table 4.4: BER's of the watermarking process with and without some Attacks

Model	No Attack	Similarity Transformation	Mesh Subdivision
Cow	0	0	0.43
Casting	0	0	0.12
Bunny	0	0	0.09
Dragon	0	0	0.05
Horse	0	0	0.02
Venus	0	0	0.01

visual perception are well integrated in the MSDM and MSDM2 measurements [30, 31]. It can be shown that both the subjective and the objective tests are within the acceptable limits of [71]. The BERs for the extraction process are shown in Table. 4.4; moreover, the BERs when the similarity transform attacks were applied are shown in this table, these attacks include translation, rotation, uniform scaling and their combination. The system is invariant to these attacks since the similarity transformation always keeps the mesh shape intact. Another attack which is the mid-point subdivision attack which in turn is considered a Connectivity attack was applied and the results are shown as well in Table. 4.4. The system performs well in general especially with larger meshes; the high BER for the Cow mesh is due to the relatively small number of vertices in this mesh; the subdivision transform causes significant changes in both geometry and connectivity. Practical meshes today tend to be of large number of vertices and edges and hence this problem is not of significant importance practically.

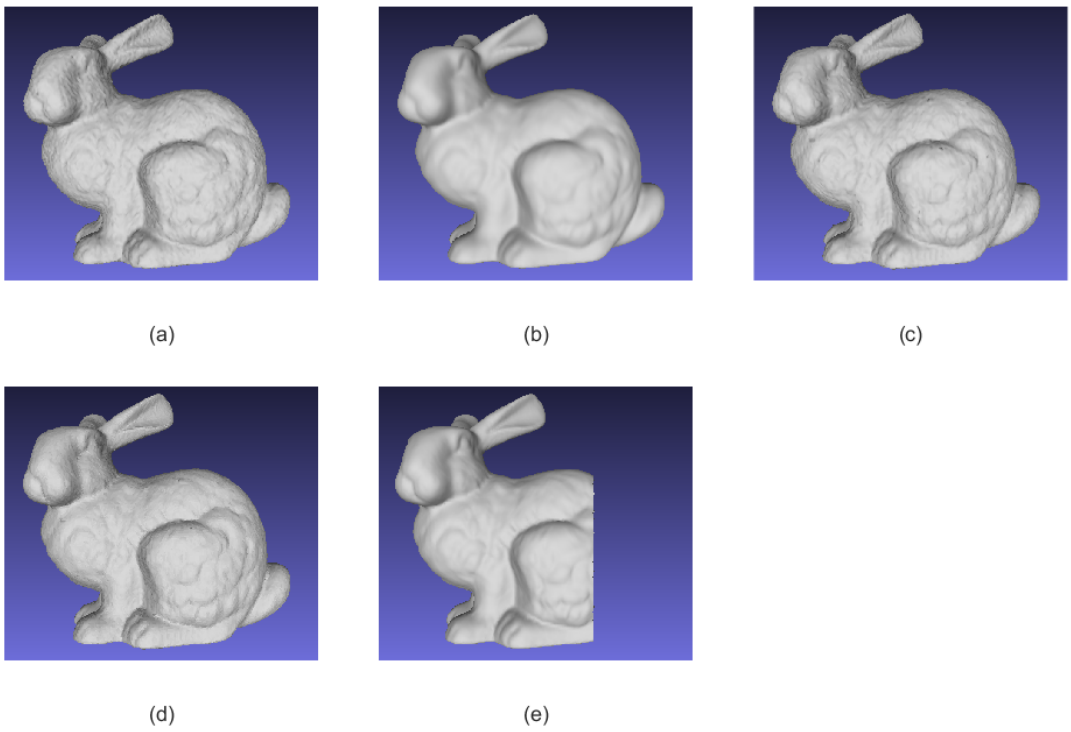


Fig. 4.20: The bunny mesh after several attacks: (a) 0.005 uniform noise, (b) Laplacian smoothing (λ=0.03, 50 iterations), (c) Coordinate quantization (8 bits) (d) Mesh simplification (50%) (e) Mesh Cropping (17%).

To evaluate the algorithm robustness against other common geometry and connectivity attacks, the watermarked objects were subjected to : uniform noise, Laplacian smoothing, vertices quantization, simplification and cropping. The results were compared with Li [85]. The Bunny mesh after these attacks is shown in Fig. 4.20; in addition, the results in terms of BER are shown in Fig. 4.21. Our results are in solid lines, while Li [85] are in dashed lines. It can be seen that our algorithm provided better performance in terms of robustness for almost all the cases. The realignment process of the center enhanced the performance especially for the smoothing, simplification and cropping attacks. For the cropping attack, the re-alignment process as well as the bins steps $[p_{min}, p_{max}]$ are required to extract the watermark.

To test the detection performance of the system, the watermark embedding process is

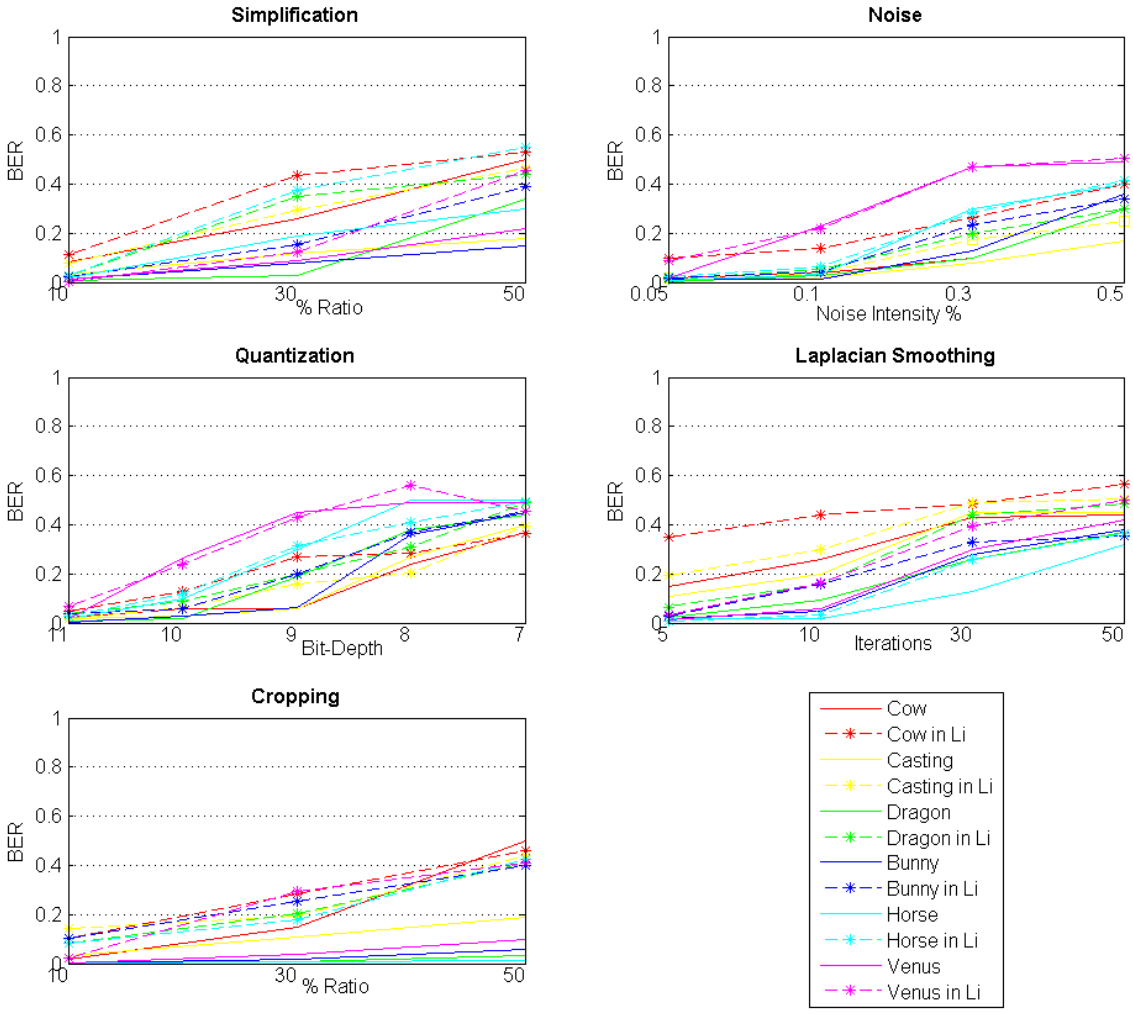


Fig. 4.21: The Robustness of our algorithm against common 3D mesh attacks and the results in Li [85].

repeated for at least 50 times by using different watermark sequences and keys, then the receiver operating characteristics (ROC) curve for each 3D Mesh is plotted [71]. This curve is a plot of the probability of true positive detection versus the probability of false positive detection [29]. The false positive detection occurs when the detector detects a watermark in an unwatermarked object. Since there are many attacks that the meshes are subjected to, a combination of attacks were used, The additive noise (intensity=0.2%), the Laplacian smoothing (λ =0.03, 30 iterations), and coordinate quantization (8 bits). These attacks are of aggressive nature, *i.e,* mean BER is 0.40 which is higher than the mean BER's as shown in Fig. 4.21. The receiver operating characteristics (ROC) curves are shown in Fig. 4.22. To validate the accuracy of the test cases, the area under curve (AUC) is used; the value of the area under the curve determines the level of accuracy [49]. Normally, this value varies between 0 and 1; the area of 1 represents perfect test, whereas the area of 0.5 represents random detection or worthless test. Fig. 4.23 shows the AUC's for each mesh. From Figures 4.22 and 4.23, it can be shown that the AUC's are more than 0.80 and in some cases more than 0.95 which indicates good detection process. Furthermore, different watermarked meshes have different robustness against other attacks as shown in Fig. 4.21; this will as well affect the ROC curves for them and hence the detection process as illustrated in Fig. 4.23.

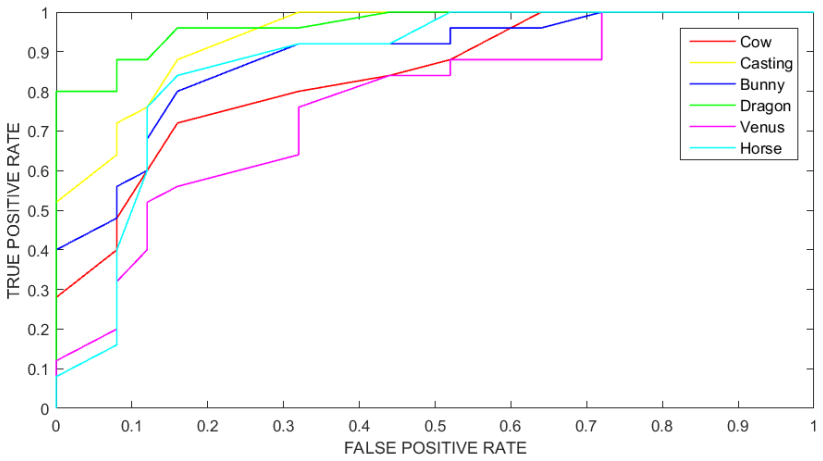


Fig. 4.22: ROC curves for different watermarked meshes

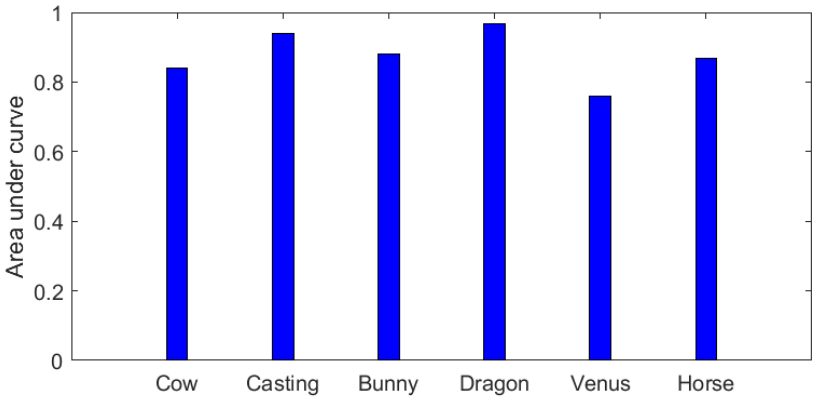


Fig. 4.23: Area Under Curve (AUC) for each mesh under aggressive attacks detection process.

Chapter 5

Conclusion and Future Work

This dissertation introduced some new techniques of data hiding in different multimedia types, the images, the videos and the 3D meshes. Investigating the different aspects of data hiding in more than one medium is important nowadays since the different types of multimedia are being integrated together. For instance, the 3D videos and images are establishing increasing popularity. This of course emanates from the wide areas that these types of multimedia are being used in; the applications on the other hand determine the quality and use of them. The medical applications, for instance, have different requirements from the entertainment fields when it come to 3D images and videos. The security and authentication processes are very important when exchanging and using data; this is the reason why watermarking is being used and different techniques are being established.

This research proposes a dual hybrid Pyramid-Wavelet-SVD watermarking process using Gaussian filters and randomly generated orthonormal filter banks. The hiding process is a dual one that uses two different hiding processes. The purpose of this duality is to increase the security of the system without significant cost in the com-

putational complexity on one hand or the visual quality of the original videos on the other hand. Our main color watermark method does not require the original videos for the extraction process like most of the SVD-based watermarking techniques. The singular values were estimated and approximated using the enhanced detection process. The errors in the singular values were reduced to the point that perfect reconstructions of our watermarks were achieved. Furthermore, the RGB color watermark was split into Bit-Plane slices of binary images for the hiding process; the binary spaces are more efficient in hiding when it comes to security, estimation and extraction; this would enable us to use other means of security such as spread spectrum sequences. Other color spaces such as YUV can also be used in the same manner. The B&W watermark matrix partition SVD-based hiding was proposed in a previous work, but we used it as a secondary hiding in the color watermark. This didn't affect the visual quality of the original frames, and at the same time, the Arnold transformed image was used as a random sequence for our main color watermarking process. Our method proved to be robust against the H.264 and H.265 compression attacks and the Transcoding attack. Moreover, it proved to be robust and superior to previous work under the traditional image processing attacks, geometrical attacks and temporal attacks. The visual quality achieved was excellent since it was possible to control the weighting factor of the hiding process. Throughout the course of the hiding, extracting and estimating processes, it was possible to reduce the weighting factor to the point that the visual qualities were maintained at more than 99% for the SSIM parameter, and the PSNR values above 55 dB on average. Different other helpful techniques were used to increase the robustness of the system such as the *directive contrast*, DFT, adaptive hiding, and smoothening processes.

Moreover, an approach for optimizing the 3D mesh statistical watermarking technique was introduced. The watermarking process originally takes place in the spatial domain, and exploits the statistical characteristics of the vertices norms. To

further enhance the robustness and the overall performance, specific schemes and methods were presented. Since the 3D meshes have many applications ranging from entertainment fields to medical and scientific fields, the two protoclos: *perceptual-quality-oriented* and *geometric-quality-oriented* should be taken into account. The VARIANCE method proposed in [83] proved to provide better perceptual qualities than the MEAN method. Statistical analysis was performed to achieve as much as possible the uniform distribution of the bins elements; power function was proposed to do that. Since the visual perceptual quality is more important in some fields than the geometric characteristics, Local roughness analysis was performed and adaptive hiding according to the roughness and smoothness of the surface was performed. Furthermore, it was found that in some attacks especially the Laplacian smoothing and the mesh simplification, the center of the object was displaced, and for each attack intensity, the new centers form what might be called clusters. Mahalanobis distance analysis was performed to prove this point. The analysis was helpful in compensating for the center's shifts, and the robustness of the watermarking process was increased by using this center shift compensation technique.

The future work will investigate other techniques that can be used in multimedia data hiding. For instance, the transform domains are still not used that much in 3D meshes. Different aspects of 3D mesh hiding processes will be investigated; furthermore, mathematical analysis will be preformed besides the statistical analysis to establish the optimal hiding process. Moreover, data hiding in 3D images and videos will be studied, and this of course will take into account the applications being used. For instance, medical 3D images are very important in different medical fields; this of course necessitates robust authentication process on one hand, and high quality on the other hand.

Another field that is getting more interest is the criminal use of data hiding. Special

work will be performed to analyze different information hiding techniques, not in watermarking field only but in steganography and covert channels, and new ideas for fighting misuse of privacy enhancing technologies will be investigated. Moreover, other criminal activities will be investigated such as obfuscation techniques.

Bibliography

[1] Stefan Katzenbeisser and Fabien Petitcolas. *Information hiding techniques for steganography and digital watermarking*. Artech house, 2000.

[2] Ross Anderson, Roger Needham, and Adi Shamir. The steganographic file system. In *International Workshop on Information Hiding*, pages 73–82. Springer, 1998.

[3] Ross J Anderson and Fabien AP Petitcolas. On the limits of steganography. *IEEE Journal on selected areas in communications*, 16(4):474–481, 1998.

[4] Ching-Yung Lin and Shih-Fu Chang. Issues and solutions for authenticating mpeg video. In *Electronic Imaging'99*, pages 54–65. International Society for Optics and Photonics, 1999.

[5] Herodotus. *The Histories (Everyman's Library)*. J.M. Dent & Sons, 1992.

[6] JD Brongers. Search for effective document security by inventioneering. In *Photonics West'98 Electronic Imaging*, pages 29–38. International Society for Optics and Photonics, 1998.

[7] [7] Y. A. Al-Gaihab. *Detection of Images with Hidden Data Using Side Information*. King Saud university, 2003.

[8] Puneet Kr Sharma. Analysis of image watermarking using least significant bit algorithm. *International Journal of Information Sciences and Techniques (IJIST) Vol*, 2, 2012.

[9] Awad Kh Al-Asmari and Farhan A Al-Enizi. A pyramid-based watermarking technique for digital color images copyright protection. In *Computing, Engineering and Information, 2009. ICC'09. International Conference on*, pages 44–47. IEEE, 2009.

[10] Juan R Hernandez, Martin Amado, and Fernando Perez-Gonzalez. Dct-domain watermarking techniques for still images: Detector performance analysis and a new structure. *IEEE transactions on image processing*, 9(1):55–68, 2000.

[11] Eckhard Koch and Jian Zhao. Towards robust and hidden image copyright labeling. In *IEEE Workshop on Nonlinear Signal and Image Processing*, pages 452–455. Neos Marmaras, Greece, 1995.

[12] Mohammad Aboofazeli, Gabriel Thomas, and Zahra Moussavi. A wavelet transform based digital image watermarking scheme. In *Electrical and Computer Engineering, 2004. Canadian Conference on*, volume 2, pages 823–826. IEEE, 2004.

[13] Deepa Kundur and Dimitrios Hatzinakos. Digital watermarking using multiresolution wavelet decomposition. In *Acoustics, Speech and Signal Processing, 1998. Proceedings of the 1998 IEEE International Conference on*, volume 5, pages 2969–2972. IEEE, 1998.

[14] Wenjun Zeng and Bede Liu. A statistical watermark detection technique without using original images for resolving rightful ownerships of digital images. *IEEE Transactions on Image Processing*, 8(11):1534–1548, 1999.

[15] Hong-Jie He, Jia-Shu Zhang, and Fan Chen. Adjacent-block based statistical detection method for self-embedding watermarking techniques. *Signal Processing*, 89(8):1557–1566, 2009.

[16] Ioannis Pitas. A method for signature casting on digital images. In *Image Processing, 1996. Proceedings., International Conference on*, volume 3, pages 215–218. IEEE, 1996.

[17] Peter Wayner. Mimic functions. *Cryptologia*, 16(3):193–214, 1992.

[18] David A Huffman. A method for the construction of minimum-redundancy codes. *Proceedings of the IRE*, 40(9):1098–1101, 1952.

[19] Peter Wayner. Strong theoretical stegnography. *Cryptologia*, 19(3):285–299, 1995.

[20] Raymond Pickholtz, Donald Schilling, and Laurence Milstein. Theory of spread-spectrum communications–a tutorial. *IEEE transactions on Communications*, 30(5):855–884, 1982.

[21] Darko Kirovski and Henrique S Malvar. Spread-spectrum watermarking of audio signals. *IEEE transactions on signal processing*, 51(4):1020–1033, 2003.

[22] Ingemar J Cox, Joe Kilian, F Thomson Leighton, and Talal Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE transactions on image processing*, 6(12):1673–1687, 1997.

[23] Minoru Kuribayashi. Coded spread spectrum watermarking scheme. In *International Workshop on Digital Watermarking*, pages 169–183. Springer, 2012.

[24] CP Sumathi, T Santanam, and G Umamaheswari. A study of various steganographic techniques used for information hiding. *arXiv preprint arXiv:1401.5561*, 2014.

[25] Nicholas F Maxemchuk. Electronic document distribution. *Bell Labs Technical Journal*, 73(5):73–80, 1994.

[26] II Sandford, Jonathan N Bradley, and Theodore G Handel. The data embedding method. SPIE, 1995.

[27] Digital watermarking — watermarkingworld.com. `http://www.watermarkingworld.com/digital_watermarking`. (Accessed on 04/15/2017).

[28] Martin Kutter and Fabien AP Petitcolas. Fair benchmark for image watermarking systems. In *Electronic Imaging'99*, pages 226–239. International Society for Optics and Photonics, 1999.

[29] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing*, 13(4):600–612, 2004.

[30] Guillaume Lavoué. A multiscale metric for 3d mesh visual quality assessment. In *Computer Graphics Forum*, volume 30, pages 1427–1437. Wiley Online Library, 2011.

[31] Guillaume Lavoué, Elisa Drelie Gelasca, Florent Dupont, Atilla Baskurt, and Touradj Ebrahimi. Perceptually driven 3d distance metrics with application to watermarking. In *SPIE Optics+ Photonics*, pages 63120L–63120L. International Society for Optics and Photonics, 2006.

[32] Yun Q Shi and Huifang Sun. *Image and video compression for multimedia engineering: Fundamentals, algorithms, and standards*. CRC press, 1999.

[33] Sviatolsav Voloshynovskiy, Shelby Pereira, Thierry Pun, Joachim J Eggers, and Jonathan K Su. Attacks on digital watermarks: classification, estimation based attacks, and benchmarks. *IEEE communications Magazine*, 39(8):118–126, 2001.

[34] José-Emilio Vila-Forcén, Sviatoslav Voloshynovskiy, Oleksiy Koval, Fernando Pérez-González, and Thierry Pun. Practical data-hiding: Additive attacks performance analysis. In *International Workshop on Digital Watermarking*, pages 244–259. Springer, 2005.

[35] Frank Y Shih. *Digital watermarking and steganography: fundamentals and techniques*. CRC Press, 2017.

[36] Vinicius Licks and Ramiro Jordan. Geometric attacks on image watermarking systems. *IEEE multimedia*, 12(3):68–78, 2005.

[37] Xuejuan Zhang, Xiaochun Cao, and Jingjie Li. Geometric attack resistant image watermarking based on mser. *Frontiers of Computer Science*, 7(1):145–156, 2013.

[38] Tong Ming, Yan Tao, and Hongbing Ji. Watermarking technique resisting to strong cropping. In *Intelligent Information Technology Application, 2008. IITA'08. Second International Symposium on*, volume 3, pages 357–361. IEEE, 2008.

[39] Matthew Holliman and Nasir Memon. Counterfeiting attacks on oblivious block-wise independent invisible watermarking schemes. *IEEE Transactions on image processing*, 9(3):432–441, 2000.

[40] HJ He, JS Zhang, and HX Wang. Synchronous counterfeiting attacks on self-embedding watermarking schemes. *International Journal of Computer Science and Network Security*, 6(1B):251–257, 2006.

[41] Gerhard C Langelaar, Iwan Setyawan, and Reginald L Lagendijk. Watermarking digital image and video data. a state-of-the-art overview. *IEEE Signal Processing Magazine, 17 (5)*, 2000.

[42] Gary J Sullivan, Jill M Boyce, Ying Chen, Jens-Rainer Ohm, C Andrew Segall, and Anthony Vetro. Standardized extensions of high efficiency video coding (hevc). *Selected Topics in Signal Processing, IEEE Journal of*, 7(6):1001–1016, 2013.

[43] Gary J Sullivan, J-R Ohm, Woo-Jin Han, and Thomas Wiegand. Overview of the high efficiency video coding (hevc) standard. *Circuits and Systems for Video Technology, IEEE Transactions on*, 22(12):1649–1668, 2012.

[44] Jantana Panyavaraporn. Multiple video watermarking algorithm based on wavelet transform. In *Communications and Information Technologies (ISCIT), 2013 13th International Symposium on*, pages 397–401. IEEE, 2013.

[45] Peter Meerwald and Andreas Uhl. Survey of wavelet-domain watermarking algorithms. In *Photonics West 2001-Electronic Imaging*, pages 505–516. International Society for Optics and Photonics, 2001.

[46] Mong-Shu Lee. Image compression and watermarking by wavelet localization. *International journal of computer mathematics*, 80(4):401–412, 2003.

[47] Víctor V Hernández Guzmán, Mariko Nakano Miyatake, and Héctor M Pérez Meana. Analysis of a wavelet-based watermarking algorithm. In *Electronics, Communications and Computers, 2004. CONIELECOMP 2004. 14th International Conference on*, pages 283–287. IEEE, 2004.

[48] Shih-Hao Wang and Yuan-Pei Lin. Wavelet tree quantization for copyright protection watermarking. *Image Processing, IEEE Transactions on*, 13(2):154–165, 2004.

[49] L Agilandeeswari and K Ganesan. A robust color video watermarking scheme based on hybrid embedding techniques. *Multimedia Tools and Applications*, 75(14):8745–8780, 2016.

[50] Yan Liu and Jiying Zhao. A new video watermarking algorithm based on 1d dft and radon transform. *Signal Processing*, 90(2):626–639, 2010.

[51] Sourav Bhattacharya, T Chattopadhyay, and Arpan Pal. A survey on different video watermarking techniques and comparative analysis with reference to h. 264/avc. In *Consumer Electronics, 2006. ISCE'06. 2006 IEEE Tenth International Symposium on*, pages 1–6. IEEE, 2006.

[52] Raul Martinez, Rogelio Reyes, Clara Cruz, Mariko Nakano, and Hector Perez. A dwt-based video watermarking scheme resilient to mpeg-2 compression and collusion attacks. In *Information Theory and Its Applications, 2008. ISITA 2008. International Symposium on*, pages 1–5. IEEE, 2008.

[53] Awad Kh Al-Asmari and Farhan A Al-Enizi. *A Pyramid-Based Watermarking Technique for Digital Images Copyright Protection Using Discrete Wavelet Transforms Techniques*. INTECH Open Access Publisher, 2013.

[54] Peter Burt and Edward Adelson. The laplacian pyramid as a compact image code. *IEEE Transactions on communications*, 31(4):532–540, 1983.

[55] Emad E Abdallah, A Ben Hamza, and Prabir Bhattacharya. Video watermarking using wavelet transform and tensor algebra. *Signal, Image and Video Processing*, 4(2):233–245, 2010.

[56] Awad Kh Al-Asmari and Farhan A Al-Enizi. *A Pyramid-Based Watermarking Technique for Digital Images Copyright Protection Using Discrete Wavelet Transforms Techniques*. INTECH Open Access Publisher, 2013.

[57] Qingtang Su, Yugang Niu, Hailin Zou, and Xianxi Liu. A blind dual color images watermarking based on singular value decomposition. *Applied Mathematics and Computation*, 219(16):8455–8466, 2013.

[58] Xiaotian Wu and Wei Sun. Robust copyright protection scheme for digital images using overlapping dct and svd. *Applied Soft Computing*, 13(2):1170–1182, 2013.

[59] Awad Kh Al-Asmari. Optimum bit rate pyramid coding with low computational and memory requirements. *IEEE transactions on circuits and systems for video technology*, 5(3):182–192, 1995.

[60] Soo-Chang Pei and Sy-Been Jaw. A class of efficient recursive quadrature mirror filters for subband coding. *IEEE Transactions on signal processing*, 39(12):2721–2725, 1991.

[61] Martin Vetterli. J. kova cevi c, wavelets and subband coding, 1995.

[62] Pragya Agarwal, Arvind Kumar, and Ankur Choudhary. A secure and reliable video watermarking technique. In *Computer and Computational Sciences (IC-CCS), 2015 International Conference on*, pages 151–156. IEEE, 2015.

[63] Gaurav Bhatnagar and Balasubramanian Raman. A new robust reference watermarking scheme based on dwt-svd. *Computer Standards & Interfaces*, 31(5):1002–1013, 2009.

[64] Jantana Panyavaraporn. Wavelet based video watermarking scheme for h. 264/avc. In *Intelligent Signal Processing and Communications Systems (ISPACS), 2011 International Symposium on*, pages 1–5. IEEE, 2011.

[65] Dawen Xu, Rangding Wang, and Jicheng Wang. Video watermarking based on spatio-temporal jnd profile. In *International Workshop on Digital Watermarking*, pages 327–341. Springer, 2008.

[66] Gwenaël Doërr and Jean-Luc Dugelay. Collusion issue in video watermarking. In *Electronic Imaging 2005*, pages 685–696. International Society for Optics and Photonics, 2005.

[67] P Vinod and PK Bora. A new inter-frame collusion attack and a countermeasure. In *International Workshop on Digital Watermarking*, pages 147–157. Springer, 2005.

[68] Awad Kh Al-Asmari. Low bit rate video compression algorithm using 3-d discrete wavelet decomposition. `http://www.intechopen.com/books/discrete-wavelet-transforms-algorithms-and-applications/low-bit-rate-video-compression-algorithm-using-3-d-discrete-wavelet-decomposition`, 2011. [Online; accessed 5-Jan-2017].

[69] Min Li, Ting Liang, and Yu-jie He. Arnold transform based image scrambling method. In *3rd International Conference on Multimedia Technology*, 2013.

[70] B Chandra Mohan and S Srinivas Kumar. A robust image watermarking scheme using singular value decomposition. *Journal of Multimedia*, 3(1):7–15, 2008.

[71] Kai Wang, Guillaume Lavoué, Florence Denis, Atilla Baskurt, and Xiyan He. A benchmark for 3d mesh watermarking. In *Shape Modeling International Conference (SMI), 2010*, pages 231–235. IEEE, 2010.

[72] Alan Frieze, Ravi Kannan, and Santosh Vempala. Fast monte-carlo algorithms for finding low-rank approximations. *Journal of the ACM (JACM)*, 51(6):1025–1041, 2004.

[73] Kai Wang, Guillaume Lavoué, Florence Denis, and Atilla Baskurt. A comprehensive survey on three-dimensional mesh watermarking. *IEEE Transactions on Multimedia*, 10(8):1513–1527, 2008.

[74] Xavier Rolland-Neviere, Gwenaël Doërr, and Pierre Alliez. Triangle surface mesh watermarking based on a constrained optimization framework. *IEEE Transactions on Information Forensics and Security*, 9(9):1491–1501, 2014.

[75] Rolland-Nevière Xavier, Gwenaël Doërr, and Pierre Alliez. Spread transform and roughness-based shaping to improve 3d watermarking based on quadratic programming. In *International Conference on Image Processing*. IEEE, 2014.

[76] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proceedings of the 26th annual conference on Computer graphics and interactive techniques*, pages 49–56. ACM Press/Addison-Wesley Publishing Co., 1999.

[77] Adrian G Bors. Watermarking mesh-based representations of 3-d objects using local moments. *IEEE Transactions on Image processing*, 15(3):687–701, 2006.

[78] Jinrong Wang, Jieqing Feng, and Yongwei Miao. A robust confirmable watermarking algorithm for 3d mesh based on manifold harmonics analysis. *The Visual Computer*, 28(11):1049–1062, 2012.

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

[80] Guillaume Lavoué, Florence Denis, Florent Dupont, and Atilla Baskurt. A watermarking framework for subdivision surfaces. In *International Workshop on Multimedia Content Representation, Classification and Security*, pages 223–231. Springer, 2006.

[81] Nassima Medimegh, Samir Belaid, Mohamed Atri, and Naoufel Werghi. Statistical robust watermarking for 3d mesh models based on salient points. In *Advanced Technologies for Signal and Image Processing (ATSIP), 2016 2nd International Conference on*, pages 52–56. IEEE, 2016.

[82] Adrian G Bors and Ming Luo. Optimized 3d watermarking for minimal surface distortion. *IEEE Transactions on Image Processing*, 22(5):1822–1835, 2013.

[83] Jae-Won Cho, Rmy Prost, and Ho-Youl Jung. An oblivious watermarking for 3-d polygonal meshes using distribution of vertex norms. *IEEE Transactions on Signal Processing*, 55(1):142–155, 2007.

[84] Ming Luo and Adrian G Bors. Minimal distortion 3-d watermarking using statistics of geodesic distances. In *Signal Processing Conference (EUSIPCO), 2012 Proceedings of the 20th European*, pages 1683–1687. IEEE, 2012.

[85] Hongyan Li, Zhengxing Sun, Miao He, and Wei Ma. A mesh watermarking method based on local roughness analysis. In *Software Engineering and Service Science (ICSESS), 2015 6th IEEE International Conference on*, pages 379–383. IEEE, 2015.

[86] Roland Hu, Patrice Rondao-Alface, and Benoit Macq. Constrained optimisation of 3d polygonal mesh watermarking by quadratic programming. In *Acoustics, Speech and Signal Processing, 2009. ICASSP 2009. IEEE International Conference on*, pages 1501–1504. IEEE, 2009.

[87] AV Tuzikov, SA Sheynin, and Pavel V Vasiliev. Computation of volume and surface body moments. *Pattern Recognition*, 36(11):2521–2529, 2003.

[88] Kai Wang, Fakhri Torkhani, and Annick Montanvert. A fast roughness-based approach to the assessment of 3d mesh visual quality. *Computers & Graphics*, 36(7):808–818, 2012.

[89] Guillaume Lavoué, Martial Tola, and Florent Dupont. Mepp-3d mesh processing platform. In *GRAPP/IVAPP*, pages 206–210, 2012.

[90] Roy De Maesschalck, Delphine Jouan-Rimbaud, and Désiré L Massart. The mahalanobis distance. *Chemometrics and intelligent laboratory systems*, 50(1):1–18, 2000.