Université d'Ottawa · University of Ottawa

PERMISSION DE REPRODUIRE ET DE DISTRIBUER LA THÈSE

PERMISSION TO REPRODUCE AND DISTRIBUTE THE THESIS

NOM DE L'AUTEUR / NAME OF AUTHOR:	ABDEL-AZIZ, Bassem
ADRESSE POSTALE / MAILING ADDRESS:	201 GOLDRIDGE DR. KANATA ON K2T1J6
GRADE / DEGREE:	**ANNÉE D'OBTENTION / YEAR GRANTED**
S.I.T.E (ELECTRICAL ENGINEERING PROGRAM)	2003
TITRE DE LA THÈSE / TITLE OF THESIS: A SOCIOCULTURAL PERSPECTIVE OF PROFESSIONAL IF WATERMARKS COULD TALK: ON TELL TALE WATERMARKS AND THEIR APPLICATIONS FOR DIGITAL CONTENT AUTHENTICATION	

L'auteur permet, par la présente, la consultation et le prêt de cette thèse en conformité avec les règlements établis par le bibliothécaire en chef de l'Université d'Ottawa. L'auteur autorise aussi l'Université d'Ottawa, ses successeurs et cessionnaires, à reproduire cet exemplaire par photographie ou photocopie pour fins de prêt ou de vente au prix coûtant aux bibliothèques ou aux chercheurs qui en feront la demande.

Les droits de publication par tout autre moyen et pour vente au public demeureront la propriété de l'auteur de la thèse sous réserve des règlements de l'Université d'Ottawa en matière de publication de thèses.

The author hereby permits the consultation and the lending of this thesis pursuant to the regulations established by the Chief Librarian of the University of Ottawa. The author also authorizes the University of Ottawa, its successors and assignees, to make reproductions of this copy by photographic means or by photocopying and to lend or sell such reproductions at cost to libraries and to scholars requesting them.

The right to publish the thesis by other means and to sell it to the public is reserved to the author, subject to the regulations of the University of Ottawa governing the publication of theses.

N.B. LE MASCULIN COMPREND ÉGALEMENT LE FÉMININ

March, 26, 2003

DATE

(AUTEUR) SIGNATURE (AUTHOR)

Université d'Ottawa · University of Ottawa

Université d'Ottawa · University of Ottawa

FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

ABDEL-AZIZ, Bassem

AUTEUR DE LA THÈSE - AUTHOR OF THESIS

M.A.Sc. (Electrical Engineering)

GRADE - DEGREE

School of Information Technology and Engineering

FACULTÉ, ÉCOLE, DÉPARTEMENT - FACULTY, SCHOOL, DEPARTMENT

TITRE DE LA THÈSE - TITLE OF THE THESIS

If Watermarks Could Talk:
On Telltale Watermarks and Their Applications
for Digital Content Authentication

Jean-Yves Chouinard

DIRECTEUR DE LA THÈSE - THESIS SUPERVISOR

EXAMINATEURS DE LA THÈSE - THESIS EXAMINERS

J. Zhao

E. Kranakis

J.-M. De Koninck, Ph.D.

LE DOYEN DE LA FACULTÉ DES ÉTUDES SUPÉRIEURES ET POSTDOCTORALES

SIGNATURE

DEAN OF THE FACULTY OF GRADUATE AND POSTDOCTORAL STUDIES

If Watermarks Could Talk:

On Telltale Watermarks and Their Applications for Digital Content Authentication

By:

Bassem Abdel-Aziz, B. Eng.

A thesis submitted to the

School of Graduate Studies and Research

University of Ottawa

In partial fulfillment of the requirements for the degree of

Master of Applied Science

In Electrical Engineering

Ottawa-Carleton Institute for Electrical Engineering

School of Information Technology and Engineering

University of Ottawa

March 2003

Table of Contents

Acknowledgements

I would like to deeply thank my supervisor, Professor Jean-Yves Chouinard, for bringing the problem of digital watermarking to my attention, and for his valuable guidance and feedback during the development of this work. I greatly appreciate his patience and confidence in my research abilities.

I would also like to thank my beloved wife, Dina El-Haddad, for her continuous support while I was completing this work. I would never be able to complete it without her enormous help.

To my father, Mahmoud El-Attar, and to the memory of

my mother, Nagah El-Hanouty.

I wish you were here.

Abstract

The increasing popularity of digital media is a mixed blessing. As opposed to analog media, replication usually produces perfect copies of the original digital content. Digital watermarking has been proposed as a method for embedding a known piece of digital data within another piece of digital data. Many digital watermarking algorithms have been proposed and tested by several parties. This has led to the emerging of new attacks against these algorithms. A good watermark must survive all known attacks to a certain acceptable degree based on the application.

In this work, we first present a brief review of the digital watermarking literature. We look into the class of fragile and semi-fragile digital watermarks used specifically for embedding copyright and authentication codes into digital media. We discuss attacks against semi-fragile watermarks, and propose methods to counter these attacks. We also implement a selected semi-fragile watermarking algorithm in order to analyze its performance. We propose some improvements in order to achieve better performance. The performance of the proposed techniques is verified and evaluated by simulations' results.

We conclude that without well designed watermarks that can withstand various types of attacks, watermarking may be rendered useless as a scheme for proof of ownership or for discouraging illicit copying of copyright material.

List of Tables

List of Figures

List of Symbols

α	Watermark signal scaling factor
β	Degree of pooling
Δ	Watermark scaling factor
a_T	Degree of masking
c_o	Original image
c_w	Watermarked image
C	Cipher-text
$C_{0,0}$	Mean value of all DC coefficients in the image
d	Perceptual distance
$d(\cdot)$	Watermark detection function
D_o	Cut-off frequency
$e(\cdot)$	Watermark embedding function
$e[i,j,k]$	Perceptual errors matrix for the DCT coefficient at location i,j within block k
$E(\cdot)$	Encryption function
f	Original image
$f(G)$	Discrimination function
$f_{o,l}(m,n)$	Haar wavelet coefficient of the image f at decomposition level l
$F(G)$	Flipping function
k_c	Coefficient selection key
k_q	Quantization key
K	Encryption or watermarking key
l_o	Luminance of original image's pixels
lw	Luminance of watermarked image's pixels
L	Wavelet decomposition level
M	Clear-text
O	Original content
O'	Watermarked content
P	Perceptual error estimate
$Q_{\Delta,l}$	Watermark binary mapping function

$s[i,j,k]$	Perceptual slacks matrix for the DCT coefficient at location i,j within block k
t	Sensitivity table
$t[i,j]$	Sensitivity corresponding to the DCT coefficient at location i,j
w_α	Amplified watermark signal in spatial domain
$w(i)$	Watermark bit i
$\hat{w}(i)$	Extracted watermark bit i
w_m	Watermark signal in spatial domain
w_s	Perceptually shaped watermark signal in spatial domain
W	Watermark
W_α	Amplified watermark signal in transform domain
W_m	Watermark signal in transform domain
W_s	Perceptually shaped watermark signal in transform domain

List of Acronyms

2AFC	Two Alternatives, Forced Choice
A/D	Analog-to-Digital
AES	Advanced Encryption Standard
AWGN	Additive White Gaussian Noise
CSF	Contrast Sensitivity Function
D/A	Digital-to-Analog
DC	Direct Current
DCT	Discrete Cosine Transform
DHWT	Discrete Haar Wavelet Transform
DWT	Discrete Wavelet Transform
ECC	Error Correction Coding
GFW	Global Fragile Watermark
HAS	Human Auditory System
HMM	Hidden Markov Model
HPF	High Pass Filter
HVS	Human Visual System
IDHWT	Inverse Discrete Haar Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform
IP	Internet Protocol
IPSec	IP Security Protocol
JND	Just Noticeable Difference
JPEG	Joint Photographic Experts Group
LPE	Local Perceptual Error
LPF	Low Pass Filter
LSB	Least Significant Bit
MPEG	Moving Picture Experts Group

MSB	Most Significant Bit
NIST	National Institute of Standards and Technology
PSNR	Peak Signal to Noise Ratio
RGB	Red Green Blue
RMSE	Root Mean Square Error
ROC	Receiver Operating Characteristic
RST	Rotation Scale Translation
TAF	Tamper Assessment Function
TPE	Total Perceptual Error
VRML	Virtual Reality Modelling Language
wPSNR	weighted Peak Signal to Noise Ratio
XOR	Exclusive-OR

1 Introduction

1.1 Thesis Motivation

1.1.1 Background

Digital watermarking uses various signal processing techniques to address the concerns of tampering and copyright violations of digital media. A watermarking system embeds copyright and authentication information within the digital media content. Compared to cryptography, the watermarking field of research is still in its infancy. A typical new cryptographic algorithm takes no less than 15-20 years before it becomes trustworthy among business and research communities. Similarly, it is not expected that a new watermarking algorithm takes less than that period of time before it is proven to be effective. A good example is the Advanced Encryption Standard (AES) cryptographic algorithm maintained by the National Institute of Standards and Technology (NIST). In January 1997, the AES initiative was announced; and in September 1997, the public was invited to propose suitable block ciphers as candidates for the AES. NIST was looking for a cipher that remains secure well into the next century. It has been about five years since that initiative had started and the winning AES algorithm (Rijndael algorithm [1][2]) is still being under heavy testing and scrutiny to make sure it is strong enough to be widely used for various applications in the 21st century [3].

The aim of this work is to examine the field of digital watermarking in order to help advance that new technology and to verify its applicability to certain real world problems.

1.1.2 History

People started thinking about hiding information many years ago. Most of the time, the goal was to secretly convey messages under the nose of an adversary. Herodotus tells how around 440 BC a trusted slave's head was shaven and tattooed with a message that would be hidden once the hair had re-grown. Herodotus also has told stories about hiding information on innocent looking wax-covered writing tablets. The wax was removed, a message was engraved on the wood, and finally the wood was covered again with wax, hiding the engraved secret message.

Techniques for information hiding ranged from hiding messages in women's earrings, and all the way to using various types of invisible ink to make small dots around an innocent looking letter. These dots were decoded by the recipient of that letter to reveal the secret message [4].

Another interesting category of information hiding techniques uses linguistics as a medium for hiding secret messages. Some writers, especially poets, cleverly wrote their works such that if the first letter of every page or every chapter is compiled into a single sentence, it will reveal some information related to the book or to the author.

Gasper Schott proposed a method to hide messages within music scores by simply mapping letters of the alphabet to the notes as shown in Figure 1.1. This type of music is not, of course, the most pleasant to play [5].

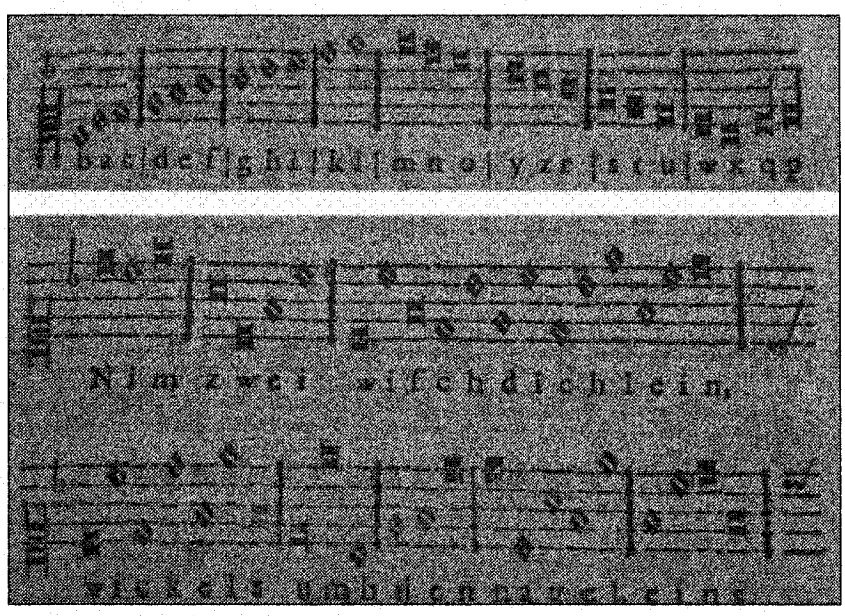


Figure 1.1 Information hiding within music scores.

A historical story shows the roots of watermarks used for transaction tracking. In 1981, some confidential British cabinet documents were being leaked to newspapers. It is rumored that Margaret Thatcher, Britain's Prime Minister at the time, ordered unique copies of each document to be distributed to its ministers. Each minister received a copy that has a small unique change in word spacing. The source of the leaks was identified this way [6].

The oldest watermarked paper that has been found dates back to 1292 AD. It originated in Italy, which has played an important role in the evolution of paper marking industry. Paper watermarks were usually used to indicate the paper brand or the paper mill. Currently, paper watermarks are used to indicate paper format and quality. Historians also use them as basis for dating and authenticating documents [4]. Paper watermarks are also widely used to help prevent forgery of bank notes and stamps.

The use of the term *watermarking* for digital images was an obvious choice for researchers when the first publication on digital image watermarking was published by Tanaka et al. [7] in 1990. Few more papers on digital watermarking were published in the next three years. In 1995, the number of publications increased significantly with the number of publications almost doubling every year.

1.1.3 A Typical Watermarking System

Figure 1.2 presents a simplified block diagram of a typical watermarking system. The original digital media could be a still image, a video clip, a digital audio clip, or even VRML, mesh models or text. Theoretically, a watermark can be embedded within any suitable set of digital bits. A watermarking key is used to guarantee that only the person who embeds the watermark knows where it is embedded and how to detect or extract it. Throughout this thesis, we always assume that the watermarking algorithm is publicly known. We use analogy with cryptography where *security by obscurity* is known to be a recipe for failure. The strength of a watermarking system should rely on the watermarking key as opposed to relying on the fact that the watermarking algorithm is unknown to an adversary [8].

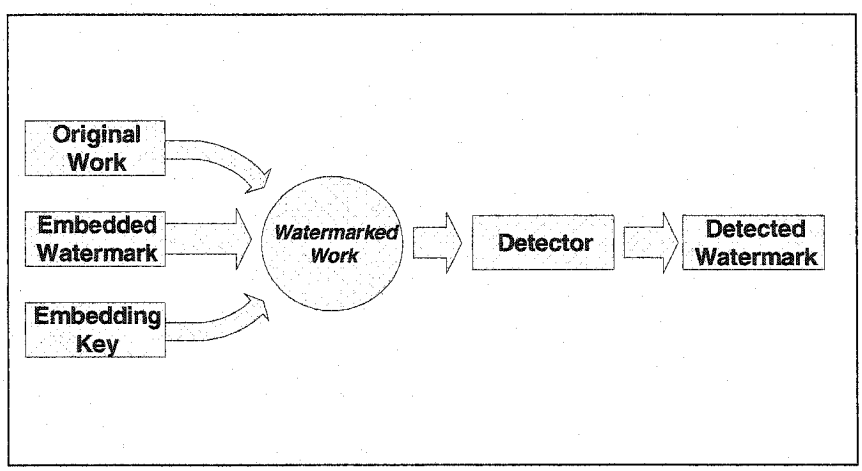


Figure 1.2 A typical watermarking system.

1.2 Contributions

The objectives of this work are to review state-of-the-art technology of digital watermarking for tamper detection and assessment in order to evaluate their potential and applicability. Moreover, we suggest ideas for improved algorithm development. We focus on the wavelet domain as a strong candidate for embedding semi-fragile watermarks and we argue that due to its frequency-scale nature, it is more suitable than other domains.

This thesis presents a background review of fragile and semi-fragile watermarking techniques. We also analyze the performance of a selected telltale semi-fragile watermarking algorithm of digital still images that uses the two-dimensional discrete wavelet domain.

The basic algorithm used, based on the algorithm proposed by Kundur et al. [9], uses the two-dimensional discrete wavelet transform to embed the watermark data by quantizing the wavelet coefficients of the transformed image.

We describe the structure of human perceptual models in general and focus particularly on Watson's model. We also discuss how these models can be used to analyze and improve the performance of watermarking systems.

We analyze the performance of the selected semi-fragile watermarking algorithm, using both traditional quality metrics and perceptual metrics, along with performance of our suggested improvements. A new structure of the algorithm parameters was used. We suggest using the modified algorithm as a robust watermarking one to take advantage of the transform domain nature of the algorithm.

For validation of the watermarking method, we have used a standard set of still images that are widely used in the watermarking literature for all our simulations. Modifications

to the images for simulation purposes were done using Adobe PhotoShop™ and CorelDraw™ image editing software along with proprietary software we developed using the C++ programming language. We have implemented the telltale semi-fragile watermarking algorithm and its suggested variations using the C++ language and MathWorks' Matlab™.

1.3 Thesis Organization

Chapter 2 gives a general introduction to digital watermarking, its applications, and essential features and desired requirements. This chapter also contrasts the role of watermarking from that of cryptography.

Some preliminary concepts that are required to gain more understanding of digital watermarking are presented in Chapter 3. It also presents various techniques used in digital watermarking algorithms along with some attacks on each technique. Benchmarking of watermarking algorithms is also discussed in that chapter.

In Chapter 4, we discuss fragile watermarking techniques and previous work. We also describe the basic telltale semi-fragile watermark algorithm we have implemented. A discussion of advantages, weaknesses, and attacks against this algorithm is also presented in that chapter.

Chapter 5 is devoted to the experimental results of our simulations on digital still images. Various theoretical assumptions are verified against actual results. A brief conclusion follows in Chapter 6, along with suggestions and ideas for future research work.

2 Literature Review

2.1 Introduction

This chapter presents an overview of digital watermarking techniques. We do not give an exhaustive review of the area. Instead, we present several types of digital watermarking techniques and algorithms relevant to our research.

2.2 On Terminology

There has been great interest between researchers with different backgrounds in the various techniques for information hiding as a solution to many modern problems. This has led to variations in the terminology used in various publications. In an attempt to clear confusion, an agreed upon terminology based on [5][10] was used in some literature.

Table 2.1 below describes some common terms.

Table 2.1 Common terminology for information hiding.

Term	Meaning
Embedded <datatype>	Something to be hidden in something else. Examples, embedded text, embedded message.
Stego-<datatype>	The output of the hiding process[1]. Also referred to as *watermarked.*
Cover-<datatype>	The original form of the Stego-message.
Stegokey or Key	Additional secret data that may be needed in the hiding process. In most cases, the same key (or a related one) is needed to detect/extract the embedded message.

[1] The word *steganography* is the modern adaptation of *stegano-graphia* assumed from Greek words that literally means "covered writing" where *stegano* means "covered" and *graphia* means "writing" [4].

In a typical scenario, An *embeddor* embeds an *embedded message* into a *cover-data* trying to hide it from a *steganalyst*. He/she will use a *stegokey* for that purpose. An *extractor* will attempt to use the same *stegokey* (symmetric embedding), or a related key to extract the *embedded message*.

2.3 Types of Information Hiding Systems

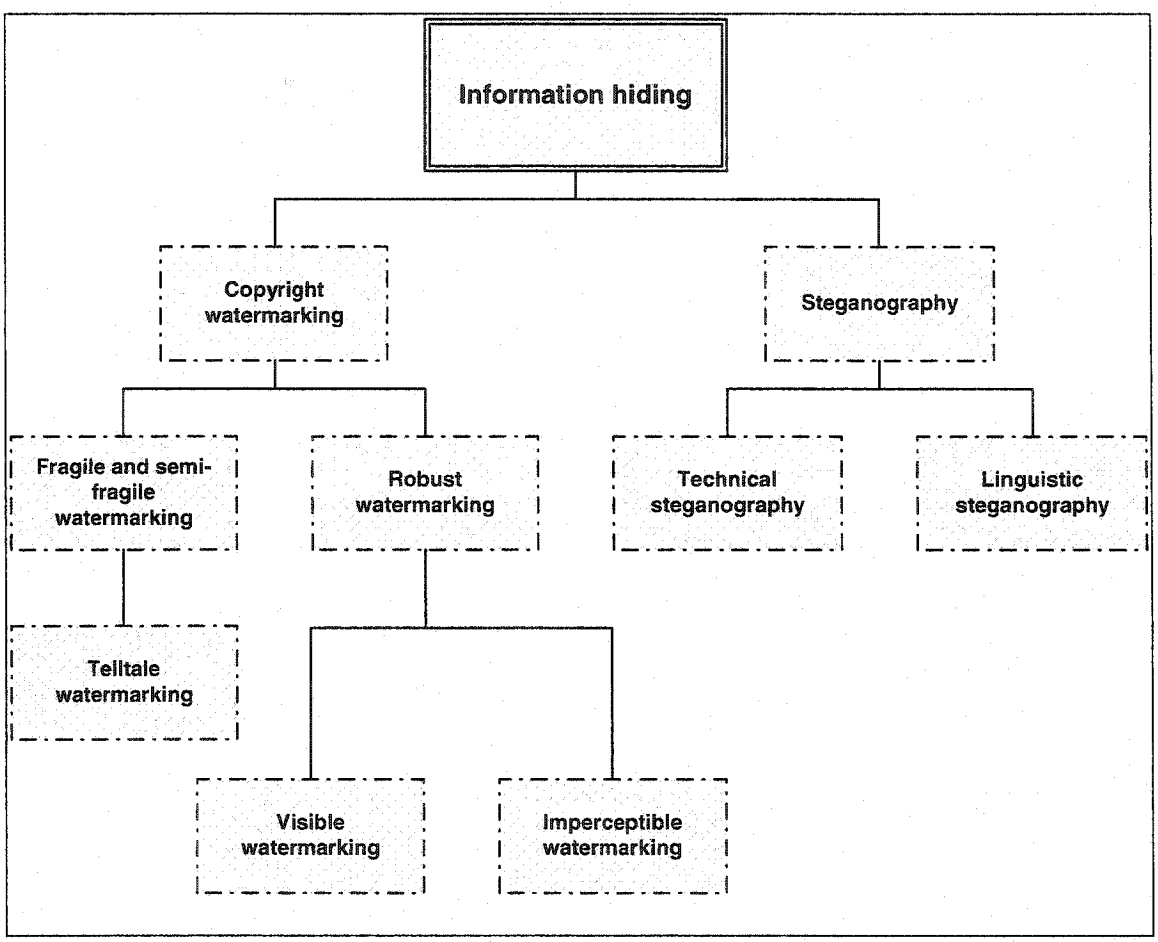


Figure 2.1 A classification of information hiding techniques based on [5].

As Figure 2.1 shows, information hiding techniques are either used to create watermarks for copyright purposes, or for sending secret messages. We are more interested on the

former application. Watermarks can either be fragile or robust based on their application. Robust watermarks can either be visible or invisible. As for fragile watermarks, they can either serve the primitive purpose of indicating that tampering of the original work had occurred, or they can take one step further by trying to give more information about the attack. For example, localization watermarks should give information about which spatial areas of an image were attacked. A more sophisticated telltale watermark might be able to give hints on the nature of attack (low-pass filtering, addition of noise, ..., etc.).

Generally, information hiding systems can be classified based on four different criteria:

1. **Purpose:** Using purpose as the criterion, the systems at hand can be classified into two types: *steganography* and *copyright watermarking*. The purpose of steganography is having a covert communication between two parties whose existence is unknown to a possible adversary. Copyright marking, as opposed to steganography, services the purpose of protecting copyright. By nature, copyright watermarks are required to resist all possible types of attacks. In the literature of digital watermarking, the stego-object is always referred to as the watermarked object [5]. In this work, we use the word watermark to mean a digital watermark, a copyright watermark, and a fragile watermark based on the context.

2. **Application:** Copyright watermarking systems can be furthermore classified into two categories. The first category is *robust watermarks,* which are designed such that it is infeasible to remove or render them useless without affecting the quality of the watermarked object to the extent that it has no value to the attacker at the same time. Robust watermarks are

sometimes referred to as *fingerprints* or *labels* indicating that they are used to identify a person, usually a customer or a user. A broadcasting cable company can label pay-per-view movies so that it can track down any customer who illegally taped a movie and made many copies available for sale. This is sometimes referred to as *transaction tracking* [11]. The term *watermark* is sometimes simply used when the hidden message is used to tell us who is the legitimate owner of the content.

The second category is *fragile watermarks* that are destroyed as soon as the watermarked object is modified beyond a specific threshold, or if it undergoes a specific type of transformation or distortion. Some literature refers to fragile watermarks as *signatures* leading to confusion with cryptographic digital signatures. Fragile watermarks are the only hope, to the best of our knowledge, digital content can be used as evidence in court. Imagine a digital camera that embeds a time-stamped fragile watermark within every digital shot it takes. Digital shots taken by that camera can not, hypothetically, be doctored.

3. Visibility: Based on the application at hand, robust watermarks can either be perceptible by humans or not. Examples of visible watermarks are shown in Figure 2.2 below [12,13]. Since visible watermarks are only good for indicating ownership of originals, most of the literature deals with invisible watermarks as they have more applications [5].

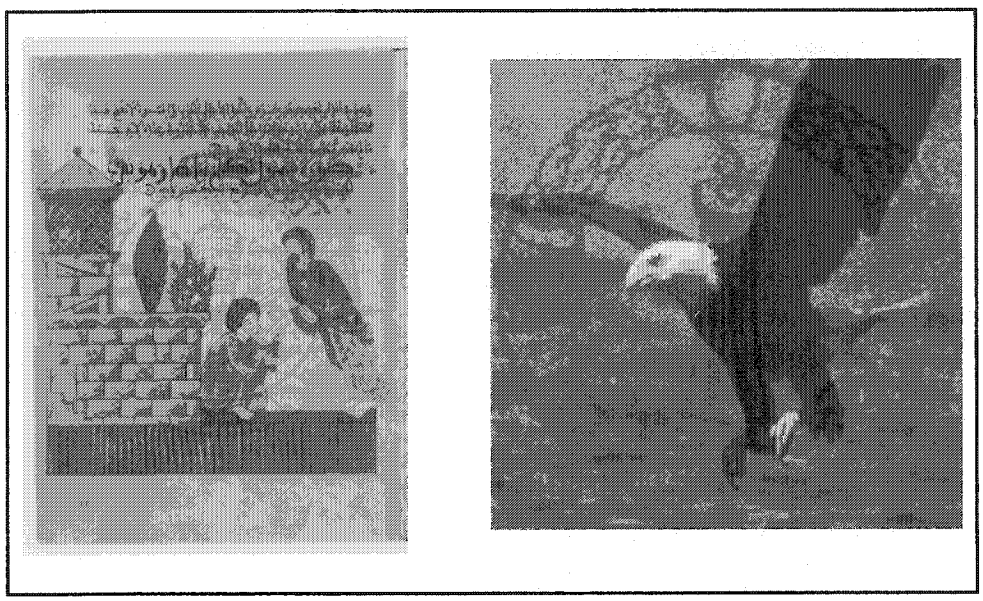


Figure 2.2 Examples of visible watermarks, (a) Arabic Novel, (b) Eagle[2].

4. **Inputs and Outputs:** Watermarking systems can either use a *detector* or an *extractor* to answer the *"is there a watermark embedded within that object?"* question. While a detector outputs either *YES* or *NO* as an answer, an extractor outputs all the bits of the extracted watermark message. Some watermarking systems require the original unwatermarked object to be at the disposal of the detector/extractor. The original object is usually used either for *registration* purposes or to simplify the detection process. As an example on image registration, consider StirMark [14], a well-known attacking tool against various watermarking systems. It can create a random array of subtle distortions to a watermarked image. These minor distortions are imperceptible

[2] http://www.vatican.va/

by humans but they may cause many watermark detectors to generate a false-negative error, i.e., fail to detect the embedded watermark. Some watermarking systems [15,16] counter this attack by using techniques from the pattern recognition literature to align the original image with the attacked, possibly watermarked image. This technique is usually referred to as *registration*. It should be noted that the registration process for images is usually referred to as *synchronization* when used with audio content [11].

Systems that do not require the original content at the detector/extractor side are referred to, in the literature [11], as *blind* or *oblivious* watermarking systems. It is more practical for a watermarking system to be of the blind type, especially for fragile watermarks where users need to authenticate a digital content without access to the original unmarked one.

2.4 The Limitations of Cryptography

When protection of ownership or authentication are mentioned, one automatically thinks about cryptography for a solution. Unfortunately, in many cases cryptography falls short as a solution, as will be explained shortly.

A cryptographic system can be viewed as a method for restricting access to some information to prevent illicit actions. Watermarking, on the other hand, does not prevent unauthorized access. Instead, it provides evidence of a wrongdoing after it has already

been committed. The success of digital watermarks partially relies on the existence of agencies that aggressively prosecute copyright infringements [17].

In symmetric key cryptography, an encryption function $E(\cdot)$ takes a key K, and some clear-text M as parameters, and produces the corresponding cipher-text C:

$$C = E(K, M) \tag{2.1}$$

Decryption is done using a decryption function $D(\cdot)$:

$$M = D(K, C) \tag{2.2}$$

In watermarking, an embedding function $e(\cdot)$ takes an original content O, a key K, and a watermark W as parameters, and outputs a watermarked content O':

$$O' = e(O, K, W) \tag{2.3}$$

Similarly, a detection function $d(\cdot)$ takes a possibly watermarked content O' and a key K as parameters, and outputs a message that could possibly be a watermark:

$$W = d(O', K) \tag{2.4}$$

At the surface, it seems that one can directly apply some well-known cryptographic technique and get guaranteed results without the need to take the risk of using a new watermarking algorithm. The previous statement is not particularly accurate because

what can be called *fuzzy*[3] *cryptography* does not exist, to the best of our knowledge, as will be discussed.

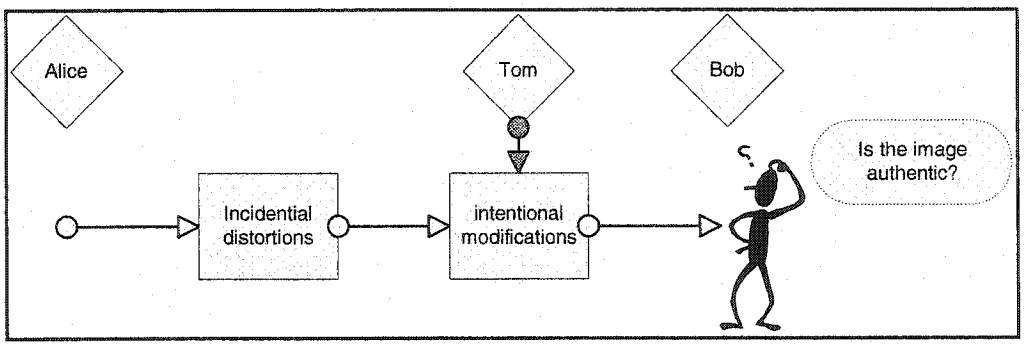


Figure 2.3 Limitation of cryptography for tamper proofing of digital media.

Consider the scenario in Figure 2.3 where Alice sends a still image to Bob through a public network. It is possible that the network is smart enough to detect the image type and apply lossy compression. This compression slightly affects the quality of the image and is deemed acceptable by both Alice and Bob. Moreover, the signal may undergo random bit errors and packet loss that still do not affect the overall signal integrity. Now consider Tom, an adversary who intentionally modifies the image in order to convey false information to Bob. Traditional authentication approaches use a one-way hash function (e.g. SHA-1 and MD5) to generate a signature or a digest of the digital image data. The signature is transmitted to Bob through appropriate mechanisms (e.g. IPSec protocol [18]). Bob, in turn, re-calculates the signature using a key known only to Alice and himself and then compare it with the signature he received from Alice. The problem with

[3] *Fuzzy Logic* is a multi-valued logic that allows intermediate values to be defined between conventional evaluations such as TRUE/FALSE. Notions like *rather warm* or *pretty cold* can be formulated mathematically and processed by fuzzy logic programs. This way, an attempt is made to apply a more human-like way of thinking in the programming of computers. Lotfi A. Zadeh initiated *Fuzzy Logic* in 1965.

that method is that unless the effective bit error rate is zero, Bob will always generate a signature different from that generated by Alice and consequently will reject the image assuming it had been tampered with by some adversary. Therefore, classical cryptography methods are not suitable for multimedia content because they are not able to distinguish between incidental unavoidable distortions and tampering. Digital watermarks promise to give us the ability to characterize the distortions that a signal undergoes on an application-dependent manner. Thus allowing us to decide whether or not a digital content integrity is maintained according to our own rules.

Ideally, a hash function that is suitable for multimedia content authentication should be able to work in a manner similar to fuzzy logic. It should use a threshold to decide whether or not the digital content is modified. This is exactly what digital watermarks can accomplish.

2.5 Applications of Digital Watermarking

2.5.1 Broadcast Monitoring

A broadcast monitoring watermark can be used to prevent radio and television broadcasting companies from overbooking air time leading to advertisers paying millions of dollars for commercials that were never aired. An incident of that type happened in Japan for 20 years before it was detected [19]. It is not easy to implement such watermarking system because it either has to achieve near 100% accuracy or it can not be used at all. Consider a major advertiser being told that 2% of their commercials are not actually aired! A less-than-perfect watermarking system can be acceptable for other applications, but not for broadcast monitoring. Another concern is that the watermark

may degrade the visual or audio quality of the broadcast material [11]. There are currently a few companies that provide watermark-based broadcast monitoring, Confirmedia™ technology for instance[4] [20].

2.5.2 Owner Identification

Consider the dilemma of an honest user who found a song on the Internet but is not sure who is the authentic owner of that song so that he can contact him for further business. Once a watermark signal is embedded into a digital content, it instantly becomes inseparable from other signals within the content. This means that digital watermarks play the technical role that the copyright symbol "©" plays as a legal copyright notice. Moreover, owner information can be encoded into the watermark signal allowing customers to get this information if they can extract the watermark. Digimarc[5] designed an Internet based robust watermarking system for digital still images. The watermark detector is bundled with the popular Adobe PhotoShop™ image editing software. When that software detects a watermark, it contacts a central database and uses the watermark message as a key to find contact information of the image's owner [11]. It should be noted that the effectiveness of owner identification watermarks in court was not tested yet. It is generally expected that it will only serve as a supplement to copyright notices but not a replacement for them.

[4] http://www.confirmedia.com/howitworks/index.cfm

[5] http://www.digimarc.com/

2.5.3 Proof of Ownership

Owner identification can be taken one step further by using watermarks as a proof-of-ownership. This is one of the hardest problems that digital watermarks are trying to solve. Assume that Alice has created an image and embedded a robust watermark as a proof that she is the original owner of that image. Bob can simply get Alice's image and embeds his own watermark claiming that he is the original owner. The resulting image has both Alice's and Bob's watermarks. This attack was introduced in [21] and is known in some literature as the *IBM attack* or the *ambiguity attack*. The solution is to design the watermark such that it is dependent on the contents of the original image. These watermarks are sometimes called *noninvertible watermarks*.

2.5.4 Transaction Tracking

In this application, a different watermark is embedded in every version of the digital content. This way, watermarks work as labels. As mentioned in Section 2.3, a good example is television broadcasting companies that embeds a unique watermark for every pay-per-view movie they send to each of their customers. They can later extract the watermark from a pirated copy of the movie and use the embedded unique watermark as evidence to prosecute the user who made illegal copies of the movie.

2.5.5 Content Authentication

This application is where fragile watermarks come into play. A fragile watermark embedded within an image will be destroyed if the image is heavily modified allowing us to verify authenticity of digital images. Early tries to create a *trustworthy camera* [22] did not use digital watermarks. Instead, cryptographic signature of every image was

generated using a key stored in the camera. The signature is not embedded into the image but it was supposed to be stored in a separate database. If a copy of an image was found to match its signature, one can be sure the image is identical to the original. This method does not work if an image undergoes lossy compression. Moreover, it costs a lot to build a signature database accessible to everyone. When a signature is lost, the associated image can never be authenticated.

A fragile watermark will solve the problems described in the previous paragraph. Once an image is watermarked, the watermark is inseparable. Any attempt to modify the image will destroy the watermark giving evidence that the image is not authentic. Fragile watermarks can also give more information about which areas of the image were doctored, usually referred to as *localization* [11]. If a fragile watermark is designed such that its data is distributed throughout all the image areas, it will be possible to tell which areas of the image were attacked by comparing the spatial representation of the detected watermark with that of the embedded one. Some techniques can detect the change of a single pixel and can locate where the changes occur in images and MPEG video [23].

Localization fragile watermarks can also be designed such that they can detect specific types of attacks and give more information about how the image was doctored. These watermarks are usually called *telltale* watermarks [9].

Semi-fragile watermarks can survive small distortions and minor transformations such as lossy compression, but are destroyed when an image is heavily modified. Since telltale watermarks are not completely destroyed by a typical attack, they are, by definition, semi-fragile watermarks.

2.5.6 Copy Control

Fragile watermarks can be used for copy control by having digital player devices detect a fragile watermark and refuse to play a music file or a video clip if no proper signature watermark is detected, preventing people from making illegal copies of copyrighted material. The main challenge that such systems face is that the whole system will only work if all player devices will contain a watermark detector. Users will always choose a device that can play and record illegal copies (for backup purposes for instance) [11].

2.6 Desired Properties of a Watermark

In this section, some known desired properties that a watermark should possess based on its target application are discussed. For visible watermarks, generally, one would expect a good watermark to be automatically embeddable, readily visible, unobtrusive, and hard to remove or fake. For imperceptible watermarks, different properties are required from both robust and fragile watermarks. Robust watermarks are desired to have the following properties [24]:

- ***Imperceptibility:***

No artifacts introduced by the watermark embedding should by noticeable by an average human eye.

- ***Robustness:***

The watermark should withstand common transformations and image processing techniques. These include digital-to-analog (D/A) and analog-to-digital (A/D) conversions, re-sampling, quantization, lossy compression, and moderate cropping. In

most applications, a robust watermark must also be resilient against processes such as photo-copying, scanning, and faxing which will apply a combination of signal distortions such as rotation, scale, translation, and cropping.

- ***Security:***

The watermark has to resist unauthorized removal, unauthorized embedding, and unauthorized detection. Attackers may add noise or slightly rotate or crop the image in an attempt render the watermark undetectable. This should result in severe degradation in data fidelity before the watermark is lost.

- ***Capacity:***

It is usually desirable to embed specific information as part of the digital watermark instead of just embedding random bits. The data can be encrypted prior to embedding to keep an acceptable, pseudo-random, statistical profile.

- ***Suitable computational complexity:***

Robust watermarks that are used for transaction tracking are usually generated by broadcasting companies right before they are distributed. This means that the embedding algorithm must be fast. Fragile watermarks used for content authentication are usually detected by player devices available to users. The requirement in this case is to have a fast detection algorithm.

Fragile watermarks must be invisible to humans. They also, based on the application, should indicate where alterations have taken place and what type of attack did the image suffer.

There is always a trade-off involved between robustness/security, visibility and capacity as shown in Figure 2.4. To increase robustness, the watermark should be embedded within the more significant parts of an image, the Most Significant Bits (MSB) of pixel intensities for instance. This will drastically change pixel intensities and make the changes involved with the watermark embedding visible. To increase the watermark capacity, more pixels will have to be modified leading again to visible changes. Designing watermarking algorithms involves judiciously trading off between the three conflicting requirements [25].

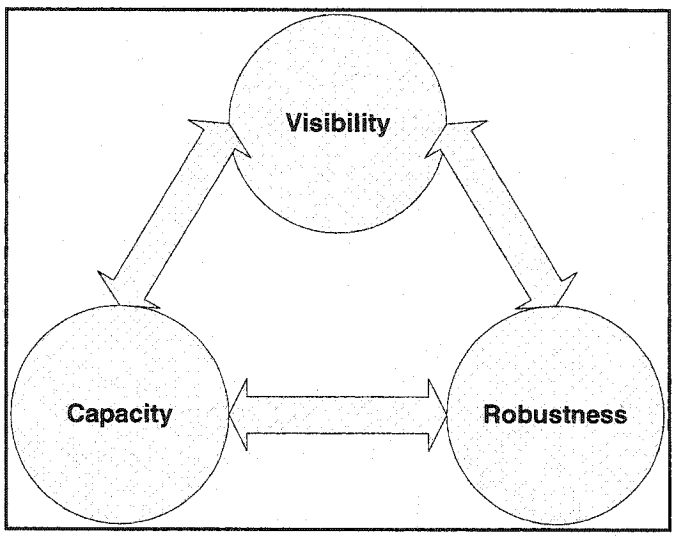


Figure 2.4 Contradictory requirements for watermarking.

In [26], Maître also indicates two other aspects of the watermarking problem. Watermarks have to balance between conflicting interests. One conflict is between efficiency of the watermark for a given application and the universality of the protection. A watermark that is suitable for one application will usually perform poorly when used for another application. We believe this conflict is one that is common with any engineering problem. The second conflict is between the cost of protection and the level of protection. One would prefer a watermark that is computationally fast to embed and/or

detect, but that might also mean it is not as secure as another watermark that requires a dedicated expensive hardware and takes more time to process.

3 Preliminaries

In this chapter we present the basic concepts and techniques used in the watermarking literature. We also discuss various attacks on watermarking algorithms and the available benchmarking tools. Special interest is given to human perceptual models.

We only focus on techniques relevant to content authentication watermarks, the focus of our research, giving some examples of previous algorithms. For more detailed literature reviews, the reader is referred to [5][27]. We also found this annotated bibliography [28] to be useful for interested readers[6].

3.1 Watermarking Benchmarks

Robust watermarks are designed so that unauthorized removal is difficult. They must be *secure*, which means they must resist attacks on the host image. They also must be *robust* by being resilient against common distortions and transformations.

Attacks against various watermarking algorithms are usually tailored for a specific type of watermark. It is not easy to evaluate how successful a watermark is in serving its purpose. This calls for criteria by which a new watermarking algorithm is evaluated. Three different benchmarking tools are well accepted by researchers in the area:

1- **Stirmark** [14][29] was the first generic evaluation tool for digital watermarks. Launched in 1997, the first version of Stirmark was used as a generic tool for simple

[6] Can be found at: http://www.cl.cam.ac.uk/~fapp2/steganography/bibliography/.

robustness testing of image watermarking algorithms. Version 4.0 of Stirmark benchmark [30], has evolved into a fully automated test bench that takes into account the application of the watermarking scheme by proposing various evaluation profiles.

2- **Checkmark** [31] takes the watermark application into account which means that the scores from individual attacks are weighted according to their importance for a given watermark usage. The authors of Checkmark view it as a second-generation benchmarking tool. They re-program most of the attacks that Stirmark supports including cropping, flip, rotation, sharpening, Gaussian filtering, random bending, linear transformations, line removal, and JPEG compression. An added advantage of Checkmark is its ability to measure the quality of the watermarked image based on three different metrics: PSNR, weighted PSNR, and the Watson metric [32], which will be discussed in the next section.

3- **Optimark** [33] is a benchmarking tool for still image watermarking algorithms. In addition to applying various attacks, this tool also generates Receiver Operating Characteristic curves (ROC), which depict plots of the probability of false positives (detection of a watermark in an unwatermarked image) versus the probability of false rejections, or misses (failure to detect a genuine watermark).

To benchmark a semi-fragile watermark, it is basically a matter of correct interpretation of the benchmarking test results. A well behaved semi-fragile watermark should obtain low performance metrics with the attacks that it is supposed to be fragile against and

obtain good performance metrics with the attacks where it is supposed to be robust against.

3.2 Image Quality Metrics

As an invisible watermark should not affect the general quality of an image, a measure by which one can judge how the quality of an image was degraded after embedding a watermark is essential.

The Peak Signal to Noise Ratio (PSNR) metric is widely used to measure the amount of difference between two images based on pixel differences [34]. In our case, the watermarked image versus the original one. For a $N \times M$ pixels image with pixels' luminance values ranging from zero (black) to N_{LUM} (white), PSNR is defined as:

$$PSNR = 10 \log_{10} \left(\frac{N_{LUM}}{RMSE} \right)^2, \qquad (3.1)$$

where RMSE is the root mean square error defined as:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{N} \sum_{j=1}^{M} [l_o(i,j) - l_w(i,j)]^2}{N \times M}}, \qquad (3.2)$$

where l_o and l_w are the respective luminance values of the original and watermarked images.

The reasons for popularity of PSNR are its mathematical tractability and the fact that it is often straightforward to design systems that minimize error using that metric. Raw error

measures work best when the error is due to additive noise. Unfortunately, they do not necessarily correspond to all aspects of the human visual perception of the errors [35].

To provide accurate measurements, image quality metrics should take into account the characteristics of the Human Visual System (HVS). If the same amount of distortion is applied to a textured area of an image, it will be much less noticeable by a human observer than if it was applied to a smooth clear area. Note that the value of PSNR will be the same in both cases.

Another example of the weakness of PSNR as an image quality metric is the human eye blue channel low sensitivity phenomenon. The human eye is less sensitive to variations in colors that lie within the blue wavelength compared to the red and green wavelength ranges. Some watermarking systems take advantage of this characteristic of the human eye by embedding a large proportion of the watermark signal in the blue channel of an RGB image [36]. In [37], a watermark is embedded into images by modifying pixel values in the blue channel using amplitude modulation. PSNR metric does not account for such phenomena of the human eye.

This calls for a quality measure that applies a model of the human visual system. To get a better understanding of such a model, the following parameter needs to be defined:

The Just noticeable difference (JND) is defined in psychophysics as the level of distortion that can be noticed in 50% of experimental observatory trials [11]. A distortion below one JND threshold is considered imperceptible by an average human observer.

In his work [32], Watson defines JNDs as linear multiples of a noise pattern that produces one JND distortion measure. In the following sections, the basic concepts used for

developing an HVS model along with a detailed description of Watson's perceptual model are described.

3.3 Perceptual Models and Watermarking

Several attempts to model the human visual system have been made. Watermarking algorithms could benefit from these models. There is always a trade-off involved in any watermarking algorithm. On the one hand, the watermark energy should be large enough to withstand expected minor distortions, or sometimes major ones depending on the application. On the other hand, a larger watermark energy will affect the visual quality of the watermarked image. Perceptual models can serve as an accurate measure to determine the optimal watermark energy trade-off. This optimal watermark energy will vary depending on the nature of the image.

The notion of JNDs was actually coined long before modern HVS models were developed. In [38], an experiment that was undertaken showed the experimental facet of JNDs. In that experiment, human observers were shown two copies of an image, one of which was of lower quality than the other. For our discussion, we can assume the lower quality image to represent the watermarked image. Observers were asked to decide which image, in their opinion, had higher quality. Assume that an image pair {A, B} was tested by many observers and only 50% of them decided that A has higher quality than B, while the rest decided on B has higher quality. Then, it can be concluded that the two images have the same quality, as the result suggests that observers were unable to identify one image as consistently having better quality than the other. This corresponds to zero JND. If 75% of the observers decided that image A had better quality than image B, then this

corresponds to one JND. This experiment is referred to as the Two Alternatives, Forced Choice (2AFC) experiment.

In most cases, a watermarking algorithm uses a scaling factor to control the amount of energy a watermark has. A perceptual model can be used in two different ways to adjust the value of the scaling factor α such that an optimal trade-off between watermark robustness and imperceptibility is achieved:

1- Perceptually Controlled Embedding

The simplest use for HVS metrics within watermarking algorithms is to scale the watermark energy, by varying the value of the scaling factor α, such that the watermarked image global distortion is lower than a specific threshold, usually one JND. In other words, the embedding strength is adjusted so as to obtain a particular perceptual distance.

In most HVS models, it is possible to measure the per-block visual capacity, i.e., how much change a specific block in the image can absorb without being visually noticed. One can use this feature to control where most of the watermark energy will be applied within the image. Arbitrary block sizes can be used depending on the application. When working with JPEG files, it is best to use 8×8 blocks as that would match the JPEG choice for block size.

It should be noted that restricting the watermark strength directly affects the watermark robustness as less separation is supported between the detection values for watermarked and unwatermarked images [11].

2- Perceptual Shaping

One can use a more sophisticated embedding technique that takes advantage of perceptual models but does not degrade watermark robustness. A stronger watermark is embedded in image areas where it is well hidden (e.g., heavily textured areas) and an attenuated watermark signal is embedded in perceptually-sensitive image areas (based on the perceptual model used).

A perceptual slack is assigned to each term of the original image, depending on the domain. For Watson's model, a slack is assigned to each DCT coefficient of an image in the block (8×8) DCT domain. Block DCT is popular as it is the transform used in JPEG still image compression standard [39]. A slack, in Watson's perceptual model is defined as the amount by which individual coefficients of the block DCT may be modified before resulting in one JND.

After the slacks are obtained using the perceptual model, the watermark signal is transformed into the perceptual model's domain. Each term of the watermark signal, which is represented by a DCT coefficient if Watson's model is used, is scaled by its slack value. This effectively *perceptually shapes* the watermark signal so that optimal perceptual embedding is achieved. If any term of the scaled watermark signal is amplified any further, it might result in visual distortion. If a term value is attenuated, that means that there was visual room for the watermark signal that was not utilized.

Figure 3.1 and Figure 3.2, from [11], illustrates the structure of the embeddor and the detector of a system that uses perceptual shaping. It should be noted from the diagram that the detector must have access to the original image. This is essential as the array of

slacks is required at the detector so that it could scale back the watermark signal to remove the effect of using the perceptual slacks. This limits the use of perceptual shaping to non-blind (non-oblivious) watermarking algorithms only. To get around this limitation, one approach [40] is for the detector to generate the slacks array using the watermarked image. Although this would generate a slacks array different from that generated by the embeddor, it would make an acceptable approximation assuming that applying the watermark to the original image was perceptually transparent.

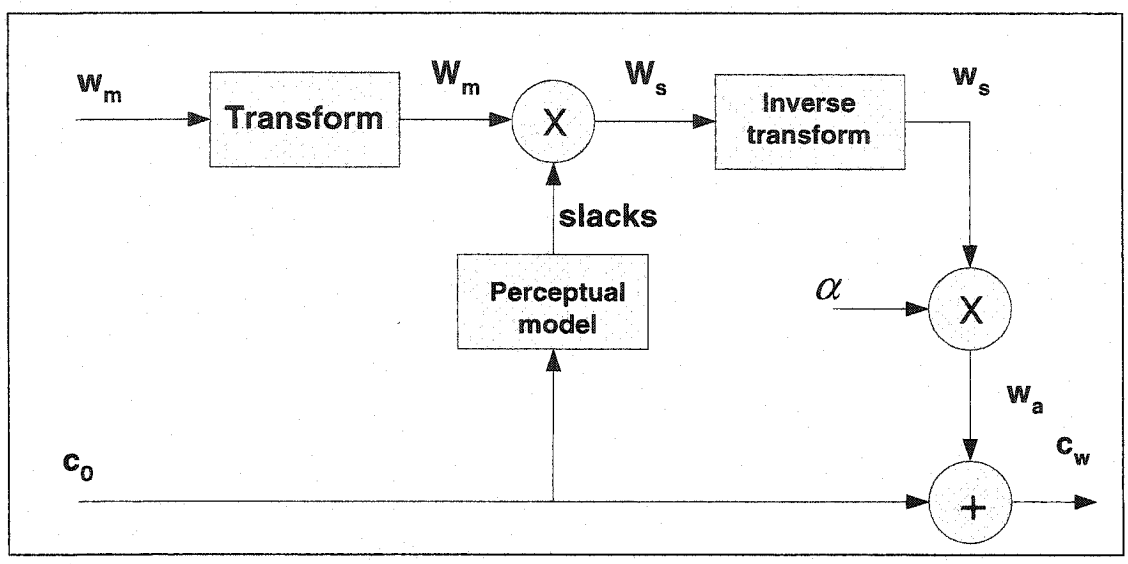


Figure 3.1 A basic embeddor that uses perceptual shaping.

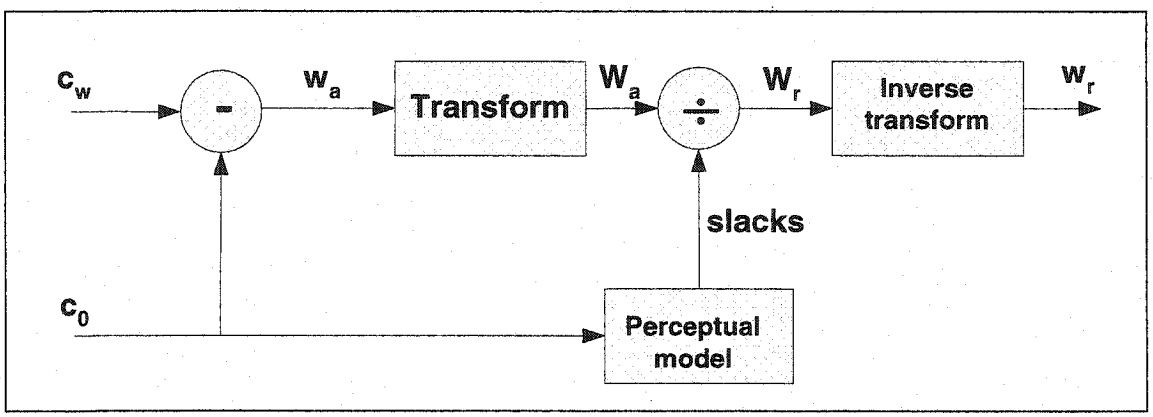


Figure 3.2 A basic detector that uses perceptual shaping.

In robust watermarking algorithms, the watermark detector could disregard the embeddor's use of perceptual shaping altogether. In that particular case, perceptual shaping could be dealt with as noise. If the algorithm used is robust against noise, it would still be able to detect the watermark without having to undo the perceptual scaling. On the contrary, watermarking algorithms that use quantization are generally more sensitive to any scaling of the watermark signal. Using perceptual shaping with this category of watermarking algorithms might lead to a higher rate of detection errors.

In [41], Watson et al. developed a similar perceptual model for the wavelet domain quantization noise. This model could also be used to generate slack values suitable for image formats that uses the wavelet transform, such as JPEG-2000 for instance. In the following section, we describe the basic features of Watson's perceptual model for block DCT domain [32].

3.4 Watson's Block DCT Perceptual Model

Human perception to visuals and audio is not uniform. Human ear responds differently depending on loudness and frequency of input audio. Terhardt's model of the Human Auditory System (HAS) [42] indicates that the human ear is most sensitive to frequencies between 2-4 kHz. It also shows that that sensitivity substantially declines at very low and very high frequencies.

The HVS shows the same pattern of variable sensitivity based on properties of its input. These properties could include spatial frequency [11], luminance contrast, and color (spectral frequency). The following are some examples of the variable sensitivity of the HVS:

- Variations in spatial frequencies are perceived by the human eye as variations in perceived patterns and textures. An image showing large smooth areas throughout has low spatial frequency. Another image representing a heavily textured area has high spatial frequency (for example, the "Baboon" test image used in our simulations). Human eyes were found to have higher sensitivity to mid-range spatial frequencies. Sensitivity drops significantly at lower spatial frequencies and it also drops to about 40% of its peak value at higher spatial frequencies [43]. It has also been shown that the human eye sensitivity varies with orientation of the two-dimensional spatial frequency patterns. In particular, the human eye is most sensitive to horizontal and vertical edges of an image and is least sensitive to lines and edges with an orientation near 45-degrees.

- Considering the Red-Green-Blue (RGB) color channel model, it is found that the human eye sensitivity to variations in the Blue-channel is significantly lower than that for other channels [34]. Some early watermarking systems took advantage of that property of the human eye and used the Blue-channel to embed a large proportion of the watermark signal.

- It was also shown that the human eye is less sensitive to brighter signals. This means that it is less probable that a human observer can notice the difference between two images if all modifications were done to brighter areas of the original image. The human eye sensitivity to variations in brightness is nonlinear and can be roughly approximated by a cube root relationship [11].

It can be noticed from the discussion above that several factors affect the human eye sensitivity. This raises two important factors that are key to the success or failure of any HVS model:

1- Context masking

In real-world images, various spatial and spectral frequencies are usually inter-woven together in such a way that it is not easy to isolate them so that the sensitivity models described above can be applied. For example, a human observer can easily notice a textured area that is in the middle of a larger smooth area. But the same observer might not be able to notice the same textured area if it was in the middle of another textured area. So, it is very important for a perceptual model to take into account the context as it might affect perception.

Context masking is defined as a measure of a human observer's response to one visual stimulus when a second masking stimulus is also present. Examples of context masking in HVS include frequency masking and brightness masking. Consider embedding a watermark that is represented by a vector of pseudo-random bits into an image. Based on context masking, it is expected that in textured areas of the image distortions caused by the watermark embedding will be less noticeable than in smooth less-textured areas.

2- Pooling

The second important issue is how a perceptual model can merge individual sensitivity models into one global model that can simulate a human perceptual system. The model must also be able to combine the sensitivity and context masking information for various

frequencies. Accounting for cases where multiple frequencies are changed rather than just one, is also required. This is known as *pooling*.

A standard feature of current visual models is the so-called *β-norm* or *Minkowski summation*. It is a method to combine the perceptibilities of separate errors to give a single estimate for the overall visual distortion of an image. Let $d[i]$ be an estimate of the probability that a human observer will notice the difference between an original image c_o and a watermarked one c_w in an individual perceptual parameter. *Minkowski summation* $D(c_o, c_w)$ representing the perceptual distance between c_o and c_w is defined as:

$$D(c_o, c_w) = \left(\sum_i |d[i]|^\beta \right)^{\frac{1}{\beta}}, \qquad (3.3)$$

where the exponent β represent the degree of pooling. When $\beta \to \infty$, the pooling rule works in such a way that only the largest error matters while all other errors are ignored. When $\beta = 1$, the pooling becomes a linear summation of absolute error values. $\beta = 2$ allows individual perceptual errors to be combined in a standard deviation type measure [32].

In the simplest case, β is given as a scalar value. In that particular case, *Minkowski summation* will give an estimate of the global perceptual difference between the two images. This is referred to as the Total Perceptual Error (TPE). To take perceptual models one step further, β can be made a matrix β_{jk} so that every DCT frequency is assigned an element of that matrix. β_{jk} becomes a measure of the visibility of artifacts within each of the frequency bands defined by the DCT basis functions. Watson [32] refers to β_{jk} as the

perceptual error matrix. A perceptual model can be applied to blocks of the image in which case $D(c_o, c_w)$ is a matrix of just noticeable differences (JNDs) where each individual element of that matrix represents the perceptual error between a single block of the original image and its corresponding block in the watermarked image.

Although *Minkowski summation* is widely accepted and used in several perceptual models, some researchers suggest that it is not a good enough measure and suggest modeling image degradation as structural distortions instead of errors [44].

Parameters of Watson's Perceptual Model

Watson's model is based on block DCT as its author intended for it to be used in JPEG image compression. JPEG compression [39] works by quantizing the DCT coefficients resulting from DCT transformation of the image pixel values. Watson suggested using a quantization function that has frequency-dependent step size. Watson's model is used to estimate perceptibility of the resulting quantization noise. The quantization step sizes are adapted accordingly so that minimum perceptual impact is introduced by the JPEG quantization process.

Watson applied most of the basic concepts described above into his model. The model consists of one sensitivity function, two masking components (for luminance and contrast masking), and a final pooling function to combine these components. Basically, an 8×8 frequency sensitivity table t, based on [45], is generated which measures the amount of change in every DCT coefficient that produces one JND. This table does not take into account any masking effects. Luminance masking is accounted for in the frequency

sensitivity table for each 8×8 block k of the image by scaling its entries using the block's DC coefficient. The resulting luminance masked sensitivity table is given as:

$$t_L[i,j,k] = t[i,j]\left(C_o[0,0,k]/C_{0,0}\right)^{a_T}, \qquad (3.4)$$

where $t[i,j]$ is the sensitivity table entry corresponding to the DCT coefficient at location i,j, where $1 \le i,j \le 8$. $C_o[0,0,k]$ is the DC coefficient of DCT block k. $C_{0,0}$ is the mean value of all DC coefficients in the image.

The value of a_T used in Watson's model, as suggested by Ahumada et al. in [45], is 0.649. a_T represents the degree of masking. Setting $a_T = 0$ suppresses luminance masking.

Equation (3.4) suggests that the perceptual capacity of a DCT block is directly proportional to the value of its DC coefficient. This matches our knowledge of the human visual system where brighter areas of the image can withstand more changes without being visually noticed.

Watson's model also accounts for contrast masking, defined as the reduction in visibility of a change in one frequency due to the energy present in that frequency [11]. The result of applying contrast masking is the *slacks* matrix s:

$$s[i,j,k] = \max\left\{t_L[i,j,k],\ |C_o[i,j,k]|^{w[i,j]}\, t_L[i,j,k]^{1-w[i,j]}\right\}, \qquad (3.5)$$

where $w[i,j]$ is a matrix of constants with values between zero and one. Watson uses the same value of $w[i,j] = 0.7$ for all i,j. Having the slacks matrix calculated, we have an estimate of how much change every coefficient in every DCT block can absorb before showing a one JND distortion.

To obtain an estimate of the visual difference between the original image and the modified (watermarked) one, the differences between corresponding DCT coefficients resulting in the error matrix are computed first:

$$e[i,j,k]=|C_w[i,j,k]-C_o[i,j,k]| \tag{3.6}$$

The error matrix is then scaled according to the slacks matrix. Each error is divided by its corresponding slack resulting in the perceptual distance matrix between the two images:

$$d[i,j,k]=\frac{e[i,j,k]}{s[i,j,k]} \tag{3.7}$$

The above perceptual distance matrix d measures the visual errors in the i,j^{th} frequency of block k in JNDs.

The final component of Watson's perceptual model is pooling where the resulting per-frequency perceptual distances are combined into a single global per-image or per-block perceptual distance. Watson's model uses two pooling steps: the first step is the *spatial error pooling* step where perceptual distances for a particular frequency $\{i,j\}$ over all blocks are combined as:

$$P_{i,j}=\left(\sum_k |d[i,j,k]|^4\right)^{\frac{1}{4}} \tag{3.8}$$

The second pooling step combines perceptual distances over all coefficients within a block. This is referred to as *frequency error pooling* corresponding to combining per-frequency error distances generated by the first pooling step into a single global perceptual error estimate P defined as:

$$P = \left(\sum_{i,j} P_{i,j}^{\ 4} \right)^{\frac{1}{4}} \qquad (3.9)$$

In our simulations, we refer to P as the Total Perceptual Error (TPE) for clarity.

The perceptual distance matrix d can be used to generate a more useful error estimate. Since each entry of d represents an estimate of the perceptual error corresponding to a specific DCT coefficient, one can calculate the mean value of all perceptual distance values within one block and use the result as an estimate of the perceptual error for that particular block. In our simulation, this per-block perceptual error estimate is referred to as the Local Perceptual Error (LPE). Each entry in the LPE matrix corresponds to the perceptual error estimate of one DCT block of the image. It should be noted that any arbitrary block size can be used depending on the application. For an image of 512×512 pixels, and using a block size of 16×16, the resulting LPE matrix will have a size of 32×32.

3.5 Summary

In this chapter, we presented various tools that can be used to benchmark a watermarking algorithm. We compared their features and suitability for various types of watermarking algorithms.

We also discussed the weaknesses of traditional image quality metrics such as PSNR. We described how human perceptual models can be used to improve accuracy of such image quality metrics. In particular, we discussed in detail how Watson's model was designed and why it is highly applicable to watermarking systems.

4 Telltale Watermarking

4.1 Problem Statement

A semi-fragile watermark protects the integrity of the content of the image rather than protecting its exact representation. Each watermark should resist specific allowed signal processing operations while being fragile against malicious attacks based on the application [46]. In the next sections, we discuss and contrast the two different types of modifications to a watermarked image. The first type comprises normal mild distortions that a watermark should be able to withstand without being lost from the watermarked image. The second type of modifications would be intentional attempts to tamper with the watermarked image.

4.1.1 Allowed Signal Processing Operations

4.1.1.1 Mild JPEG Compression

JPEG is a lossy compression based on the two-dimensional Discrete Cosine Transform (DCT). The basic idea is to quantize DCT coefficients based on pre-calculated quantization tables that are experimentally optimized for better perceptibility. DCT coefficients that are less visually significant in the structure of the image are quantized and sometimes removed (replaced by zeros). Figure 4.1 below shows a block diagram of the DCT based encoder from [39].

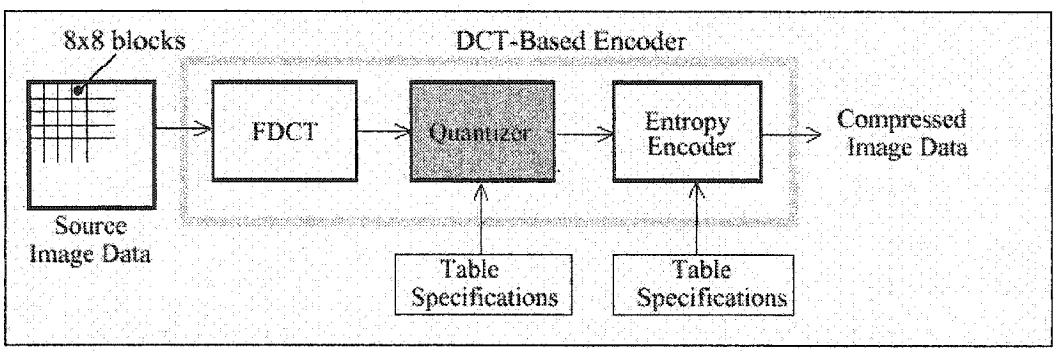


Figure 4.1 Block diagram of JPEG compression algorithm [39].

JPEG-2000 is the new emerging standard for digital image compression. It uses the 2D Discrete Wavelet Transform rather than DCT transform [47].

In Figure 4.2, an original uncompressed image, a 70% JPEG compressed version, and a 90% JPEG compressed one are shown. The 70% version is deemed visually acceptable for most applications, except for medical applications for instance. A semi-fragile watermark should withstand up to that amount of JPEG compression.

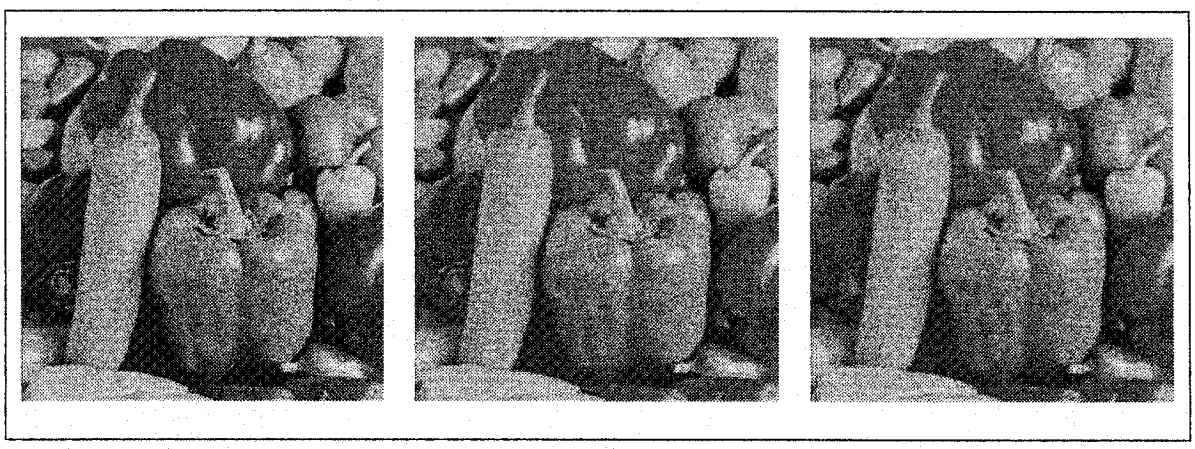


Figure 4.2 (a) Original image, (b) 70% JPEG compression, (c) 90% JPEG compression.

4.1.1.2 Histogram Equalization

Histogram equalization is a technique used to improve the visual appearance of an image. Peaks in the image histogram are widened, while the valleys are compressed. This improves the dynamic range of the pixel values and hence usually improves visual

appearance of an image [34]. Generally, histogram equalization allows one to modify the dynamic range of pixel intensity values by altering these values such that it follows the desired shape of a predefined histogram. A side effect of excessive histogram equalization is ending up with an image that has washed-off areas.

Figure 4.3 shows a rather dark image before and after histogram equalization. The following figures, Figure 4.4 and Figure 4.5, depict the original and the improved (equalized) histograms of the image respectively. The histogram of the original image shows that most of the pixels are located toward the dark end of the gray scale. Equalization was applied so that the resulting histogram is such that a roughly equal number of pixels is mapped into one of 64 gray color levels. The histogram equalized image has a better dynamic range of intensities and a better appearance as well.

"Smile" before histogram equalization

"Smile" after histogram equalization

Figure 4.3 The visual effect of histogram equalization.

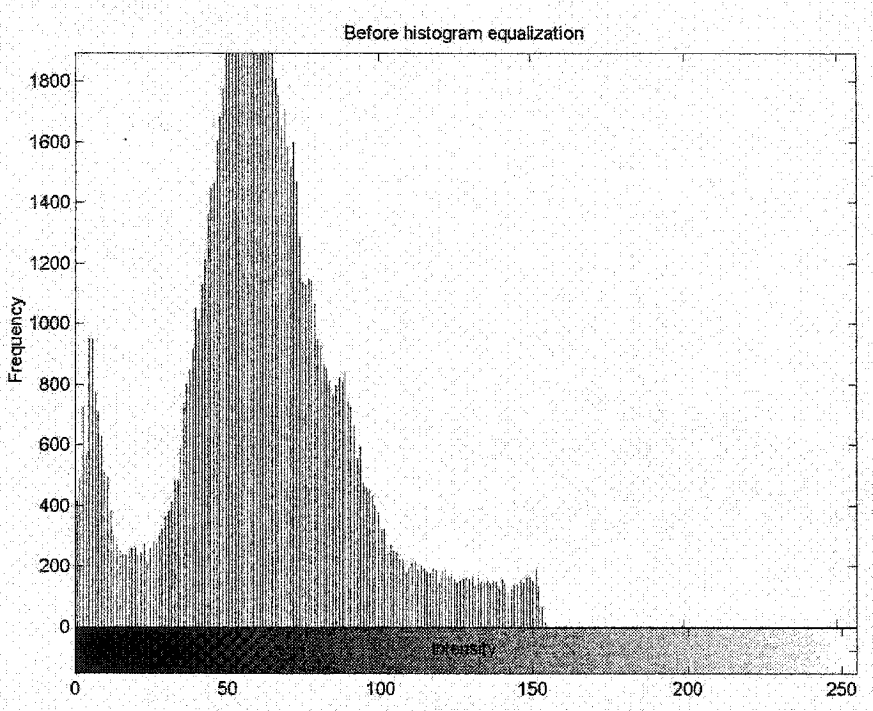


Figure 4.4 Histogram of the original, dark "Smile" image.

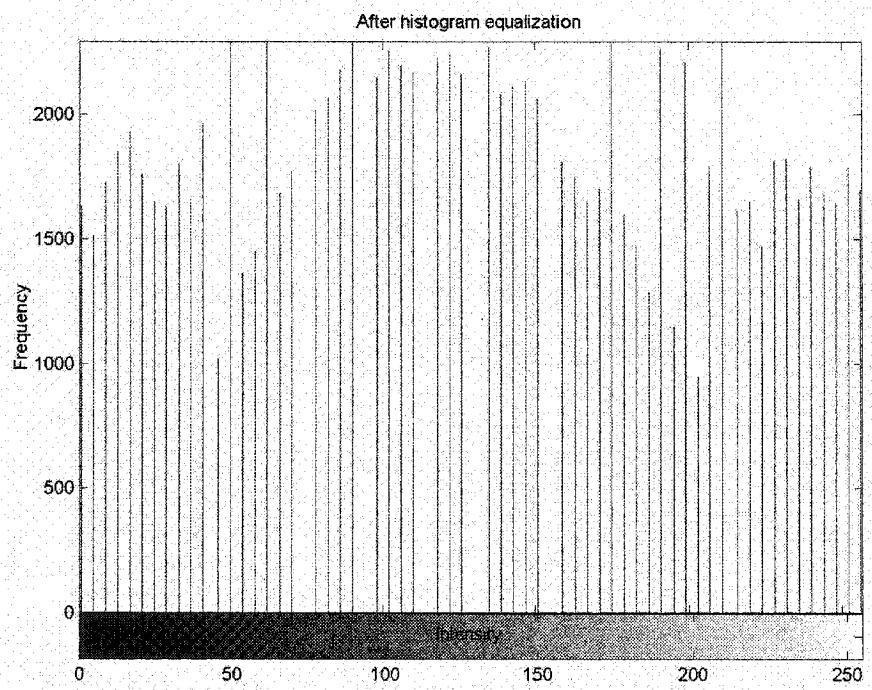


Figure 4.5 Equalized histogram of "Smile" image.

4.1.1.3 Spatial Filtering

Spatial filtering is an operation applied in spatial domain using spatial masks to enhance images. The two most widely used spatial filters are the Low Pass Filter (LPF) and the High Pass Filter (HPF).

LPFs generally have a smoothing effect on images. Therefore, they are used for purposes such as blurring and noise reduction. HPFs are often referred to as *sharpening filters* as their effect is to enhance image details that have been blurred. A linear spatial filter can be represented with a mask that is convoluted with the image pixels.

Figure 4.6 and Figure 4.7 show two ideal 3×3 two-dimensional spatial filters along with cross sections of their impulse responses, where D_o is the *cut-off frequency* of the filter measured from the origin of the frequency plane. Effectively, for the LPF shown, each center pixel value is averaged using the values of its eight neighboring pixels. For the HPF, the values of the eight neighboring pixels are subtracted from an amplified value of the affected pixel. This has the effect of contrasting that pixel's intensity from neighboring pixels' intensities.

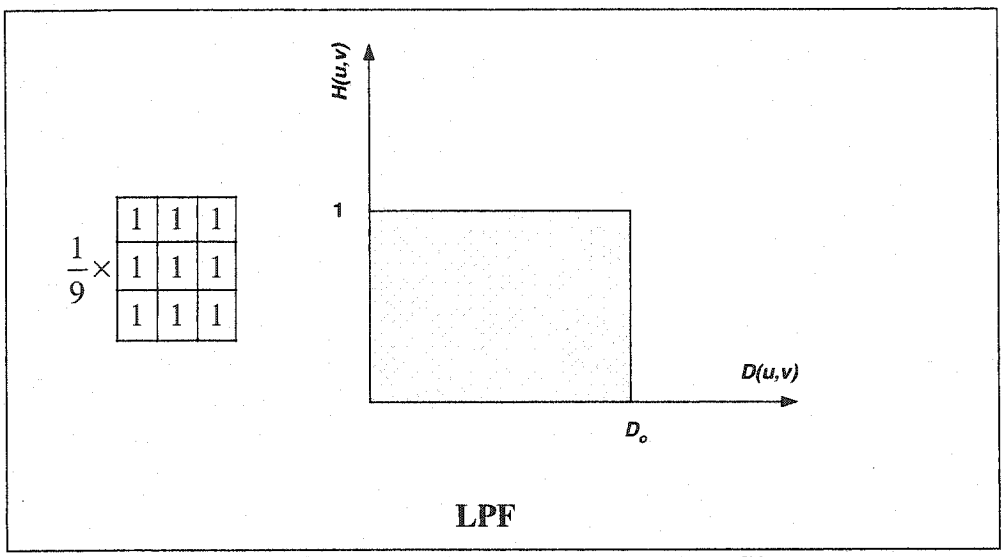


Figure 4.6 An ideal low-pass (averaging) filter.

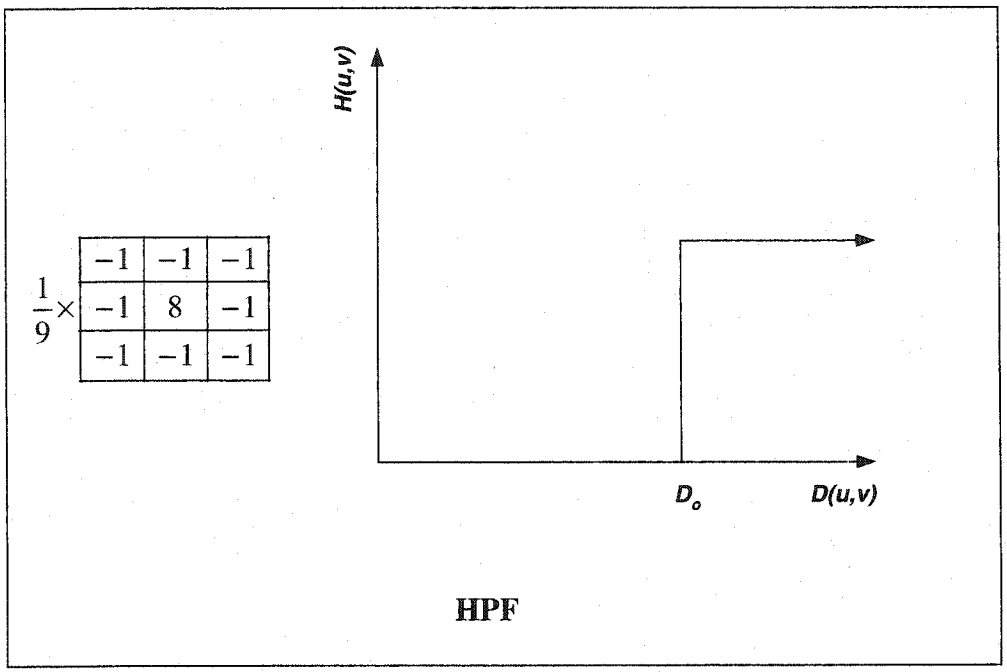


Figure 4.7 An ideal high-pass (sharpening) filter.

4.1.1.4 Gaussian Noise

Applying JPEG compression to an image with a compression ratio of 75% will generate PSNR value in the range of nearly 35 dB for an average image. A watermarked image with PSNR ranging between 40-50 dB is indistinguishable from the original [48]. Using

the same analogy, one can state that a semi-fragile watermark should withstand mild Gaussian noise as long as the PSNR of the affected image is in the range 40-50 dB. The same criteria can also be applied on random bit errors.

4.1.2 Attacks

A semi-fragile watermark should be able to detect some of the following intentional modifications:

4.1.2.1 Removal, Substitution, and Insertion Attacks

These attacks will, for instance, try to replace a face of a suspect caught by a surveillance camera by another face in an attempt to confuse prosecution authorities. These attacks might only affect specific areas of the image. So, a desired feature of the watermark is to detect which areas were modified, a feature referred to as *localization*. Some semi-fragile watermarks use *self-embedding*, a technique where the watermark data represent a thumbnail representation of the original image. If watermark bits are interleaved properly throughout the image, either spatially or spectrally, the watermark extractor would be able to restore the attacked areas, albeit with a lower resolution.

4.1.2.2 Attempts to Modify Image Geometry.

This category of attacks comprises Rotation-Scale-Translation (RST) operations, and other operations such as cropping. These attacks are easy to detect because they will cause de-synchronization between the embedded watermark and the extracted one.

A semi-fragile watermark is deemed usable if it has a good pattern of *false alarm rate* versus *detection failure rate*. A false alarm is the event when a watermarked image gives an indication of tampering in the absence of malicious attacks. A detection failure happens when an attack on the watermarked image is not detected. It should be noted that the desired watermark behavior is dependent on the application. For instance, a semi-fragile watermark that is generated by a trustworthy camera is required to demonstrate an extremely low rate of false alarms, while this might not be necessary for other applications.

4.2 Fragile and Semi-Fragile Watermarks

4.2.1 Fragile Watermarks and Content Authentication

One main problem that darkens the future of watermarking as a content authentication tool is that as soon as a new watermark is used, an attack is readily developed to render the new watermark useless.

Applications of fragile watermarks are usually different in nature. For example, equipping digital cameras with a watermark embeddor to make sure that all digital shots taken by that camera will not be tampered with. If the embedding system can be trusted, then digital images can be used as evidence in court. If this does not seem to be a serious application for fragile watermarks, then one can consider cases where an inadvertent change to an X-ray image results in misdiagnosis. This will also prevent someone from replacing an X-ray with a fake, or from altering an original X-ray one.

A fragile watermark, as its name implies, will disappear (i.e., become undetectable) if the content (usually called *host*) is changed in any way. This will raise a very important issue: how can the word "change" be defined in that context. In the digital camera example, one might need to apply JPEG compression to an image and will not expect the watermark to vanish because of that. On the other hand, a watermark designed for medical purposes should not withstand any change (usually for legal reasons, e.g., in malpractice lawsuits).

An ideal fragile watermark will be able to give us information about:

1- Whether the host content was altered and how significant was the alteration.

2- What parts of the host content have been altered.

In some cases, the watermark can also help restore the altered parts of the content to its original state by deploying the self-embedding techniques mentioned in Section 4.3.3. In the following section, we describe and classify fragile watermark algorithms.

4.2.2 Global Fragile Watermarks

A global fragile watermark is only able to tell us that "some change" took place in the host image. Even if a single pixel was changed, such a watermark will become undetectable.

One of the simplest GFW algorithms for grayscale images will involve modifying the Least Significant Bits (LSB) of pixel values according to an embedding key. Since any change to an image will change many LSBs, detection of the embedded sequence of bits implies that the image is authentic.

The authors of [49] argued, using statistical analysis, that LSBs of pixel values are not completely random, but bare a rather loose correlation with each other. Therefore, embedding a watermark sequence in them will lead to a change in their statistical properties. This is more of a problem to secure or robust watermarks than to fragile ones. But, still, the attacker now knows for a certain degree that an authentication watermark is embedded, and he will be more cautious in carrying over his attack.

Another potential weakness of simple LSB watermarks is that the embedded sequence is usually not dependent on the host. An attacker can simply copy LSBs from an authenticated host and embed it into a tampered host creating a fake authentic host.

4.2.2.1 Fragile Watermarks as Signatures

Fragile watermarks are suitable for embedding authentication signatures. Authentication signatures are cryptographic digests of the host data. Without watermarks, one has to either store these signatures as meta-data in the digital file, or store them separated from the host increasing the risk of losing them. Using watermarks reduce that risk. For example, the authentication signature of a digital image is generated and then embedded in the image in the form of a fragile watermark.

There is only one problem with the above technique. It will not work! Embedding the watermark will change the image causing the authentication test to fail. To solve this problem, the data used to calculate the signature should not be changed by the watermark sequence. For instance, one can estimate the signature over the odd image lines and embed the watermark in the LSBs of the even lines. More complex algorithms to partition the image can be used to increase security.

4.2.2.2 Invertible Fragile Watermarks

In some applications it is required that once the authenticity of host content has been verified, it should be possible to erase the fragile watermark so as to remove any distortions introduced by the watermark. This feature is particularly crucial for medical lawsuits and military applications.

Designing a fragile watermark that is invertible is not easy because it means that it is impossible to embed the watermark in 100% of digital content [11]. It is possible that an embeddor may fail to embed the watermark because there is no room for the watermark data in the host content, or because of restrictions enforced by the HVS model used for embedding. The term *watermarking effectiveness* can be defined as the ability of a detector to detect a watermark right after it is embedded to the host. It is always desired to have a 100% effectiveness, but unfortunately it is impossible to achieve this with invertible watermarks.

Another challenge for invertible watermarks is the *truncation* problem. If every grayscale image pixel can have a value $[0,255]$ and a non-adaptive algorithm is used, then pixels holding 255 (highest intensity represented by white color) may be truncated to 0 (lowest intensity, black) if a watermark bit is added to their intensity values. Similarly, a black pixel will be inverted to a white one if a watermark bit was subtracted from its intensity value. This will create a phenomenon referred to as *salt and pepper noise* which is usually quantified by the percentage of pixels which are corrupted. Figure 4.8 shows an image with 2% salt and pepper noise.

Figure 4.8 Salt and pepper noise affecting 2% of the "Roses" image pixels.

Assume that an image has N pixels with each pixel represented by 8 bits; then there are $2^{8\times N}$ possible unwatermarked images. Invertibility requires one-to-one mapping between unwatermarked and watermarked images. But since the image space is limited to $2^{8\times N}$, one needs extra $2^{8\times N}$ images to represent the watermarked image space. This is not possible as only eight bits are used to represent a pixel. So, to achieve 100% effectiveness, one will end up with a detector with 100% false positives because *any* image in the $2^{8\times N}$ space will be considered to be watermarked by the detector. It should be noted that the one-to-one relation is not required for regular watermarks. For instance, it is completely correct that two unwatermarked images map to a single watermarked image. This happens all the time with non-adaptive embedding algorithms where the embedded data is independent on the image (consider two images that only differ in LSBs and an embeddor that uses LSBs to embed the watermark bit sequence). This conclusion was experimentally verified in [11].

In [46], an invertible watermark is proposed. The authors used lossless compression to solve the invertibility problem. Lossless compression is applied to one bit-plane starting

from the LSB-plane. If the chosen bit-plane is losslessly compressible, the space created by the compression process is used to embed the watermark bits. If the chosen plane is not compressible or if compression did not create enough space for the watermark bits, then the process is repeated with the next bit-plane. It was found that for most high-quality images a signature watermark can fit into one of the lowest three bit-planes.

The algorithm described above was further improved in [50] where an image is partitioned into groups of consecutive pixels G with arbitrary group size n. A discrimination function $f(G)$ is defined to estimate the degree of "regularity" of every group in the image. An example discrimination function is one that measures the variation of the group:

$$f(G) = f(x_1, x_2, \cdots, x_n) = \sum_{i=1}^{n-1} |x_{i+1} - x_i| \qquad (4.1)$$

Finally, an invertible operation over a pixel group called "flipping" F is defined. This flipping function permutes the intensity values of some or all pixels in the group such that $F(F(G)) = G$. An example flipping function will just flip the LSB of each pixel as follows: $0 \leftrightarrow 1$, $1 \leftrightarrow 2$, $2 \leftrightarrow 3$, ... , $254 \leftrightarrow 255$. Note that F is fully invertible. Groups are classified according to the output of the discrimination function $f(\cdot)$ as shown in Table 4.1.

Table 4.1 Group classification according to *f()*

G is Regular R if:	*f(F(G)) > f(G)*
G is Singular S if:	*f(F(G)) < f(G)*
G is Unusable U if:	*f(F(G)) = f(G)*

In typical images, most of the groups are usually of type R (hence the name). Groups of type U can not be used to embed watermark bits. Doing so will confuse the detector because $f(\cdot)$ will be the same for the group before and after embedding watermark. It is obvious that if G is regular, $F(G)$ will be singular and vice versa. A zero is assigned to R groups and a one is assigned to S groups. The embeddor will apply the flipping function to each S or R group that does not match its corresponding watermark bit, otherwise the group pixels are unchanged. As mentioned earlier, invertible watermarks can not use all of the image pixels for embedding. So before embedding the watermark, a RS-vector that stores the types of image groups is generated and losslessly compressed. The watermark bits are appended to that vector and the result bit-stream is embedded using the flipping function F.

The detector will extract the embedded bit-stream (RS-vector + watermark W). Once W is verified, the detector will decompress RS-vector and use it to restore all groups to their original state. This way, an exact copy of the original image is obtained.

4.2.3 Selective Fragile Watermarks

In many applications, the perceptual similarity between images is enough to prove authenticity. However, a compressed JPEG image will fail authentication by all fragile watermarks described in previous section even though it is visually identical to the uncompressed original.

Selective watermarks will allow us to decide which distortions are significant enough to cause authentication to fail. These watermarks can withstand some minor - legitimate -

distortions, so they are sometimes called *semi-fragile* watermarks [11]. Deciding whether some type distortion is legitimate or not is application dependent.

An intuitive way to look at semi-fragile watermarks is to think of them as dual-marks. A semi-fragile mark is robust against some legitimate transformations and fragile against illegitimate ones. A technique similar to *Cocktail Watermarking* [51] readily lends itself to that application. In other words, instead of embedding one watermark, one can embed many watermarks using various embedding algorithms such that every watermark is robust against specific type of attack. This is analogous to cryptographic algorithms that use several iterations - cycles - of a simple substitution operation in order to generate encrypted output that is statistically pseudo-random.

An alternative way to look at semi-fragile watermarks is to think of them as robust watermarks that are tuned so that they are destroyed when the host undergoes transformations beyond a predetermined level. Many systems were designed using that perspective, e.g., [52].

A more advanced fragile watermark can also give information about the nature of distortions the host has undergone. Possibly, this watermark can also localize the distortions either spatially or spectrally.

Some systems that attempt to identify features of the host content that are invariant to legitimate distortions but are not so to illegitimate ones were presented, e.g., [53]. Other systems base their signature watermarks on perceptually-significant features of host content, e.g., [54]. There is room for improvement of these algorithms especially with the newly revised visual models developed in the last three years.

4.3 Telltale Watermarks, Wavelets, and Other Techniques

Telltale watermarks exploit the fact that the watermark undergoes the same transformations that the host content does. By analyzing a telltale watermark, one can get information on "how" the host was attacked rather than on "whether" it was attacked or not. A good use for telltale watermarks is when the line between legitimate and illegitimate distortions can not be easily drawn.

An example of telltale fragile watermark embedded in the DCT domain is proposed in [55]. This scheme can detect any tampering in the watermarked image and can also locate where the tampering has occurred.

4.3.1 The Discrete Wavelet Transform

In this section, we discuss why wavelets are a good vehicle to implement telltale watermarks. As shown in Figure 4.9, transforming an image into the Discrete Wavelet Domain (DWT) will give us information about the scale which can be used to do frequency domain analysis.

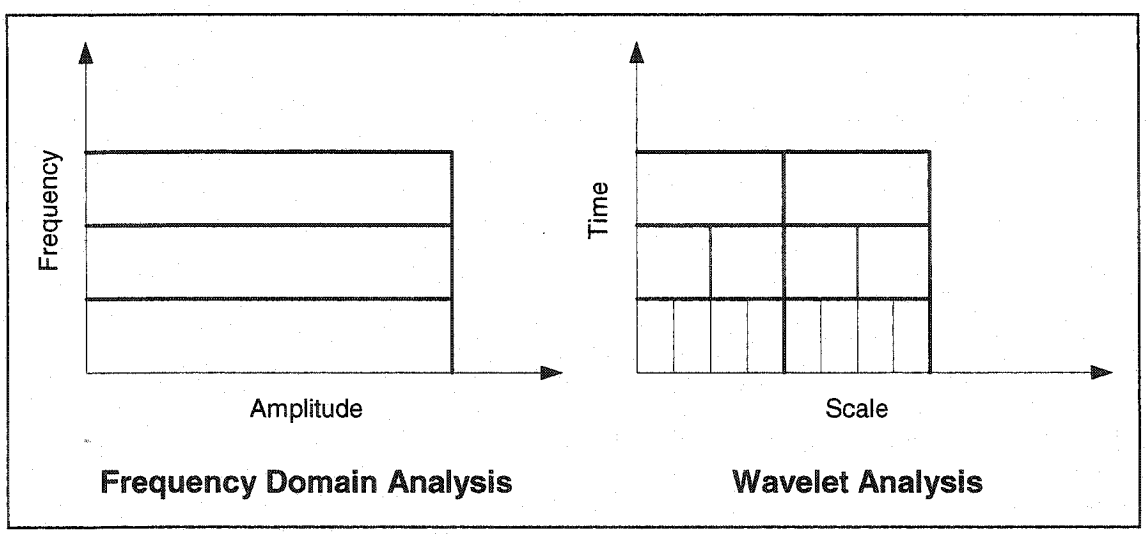


Figure 4.9 Wavelet analysis versus Fourier analysis.

It is already known that Fourier analysis can be used for that purpose. The advantage of wavelet analysis over Fourier analysis is that in the wavelet domain, spatial information about the image is not lost. More information about the DWT and the roots of wavelet analysis can be found in [56] and [57].

Wavelet decomposition of an image will generate a group of coefficient subsets. Each subset corresponds to a scale or a frequency sub-band. By inspecting how the coefficients of each sub-band are changed from their values in the un-attacked image, a good measure on how severely each sub-band was attacked can be obtained.

For example, by inspecting coefficients of high-frequency sub-bands one can decide whether or not the image had undergone a low-pass filtering attack. Taking advantage of the spatial information in the DWT domain, cases when all frequencies in a narrow spatial region have been equally distorted can also be detected. This implies to a high degree of certainty that this region was edited [11].

Consider cases where attackers maliciously change the license plate number in a digital photo of a car that was involved in a fatal crash.

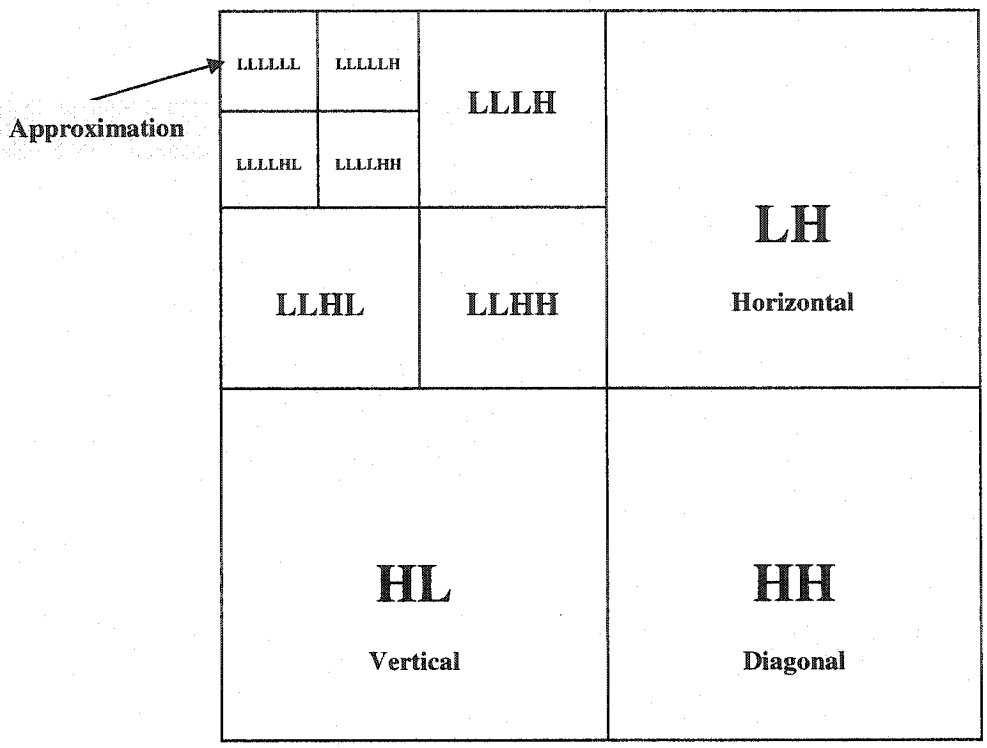


Figure 4.10 Three levels of wavelet decomposition.

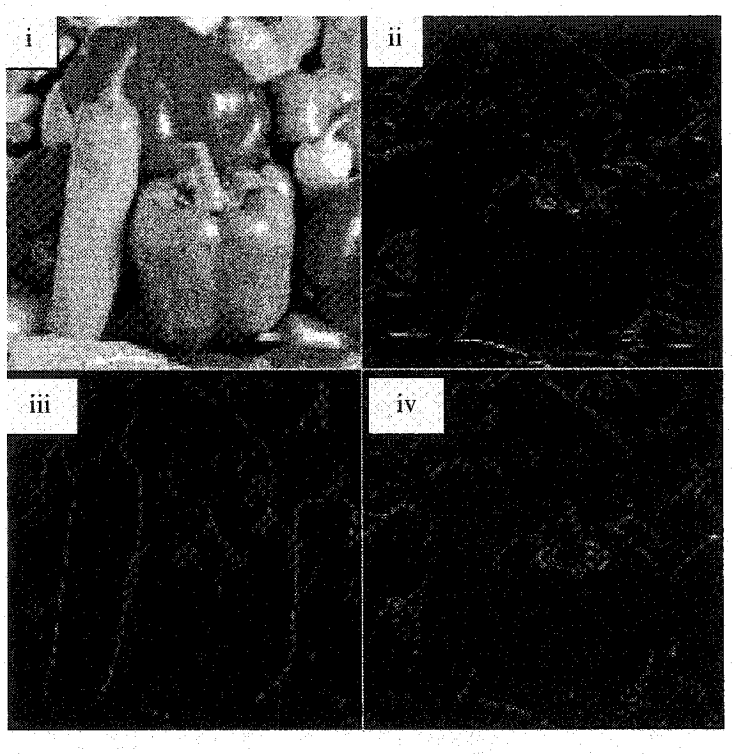


Figure 4.11 i- Approximation, ii- Horizontal, iii- Vertical, and iv- Diagonal.

Telltale watermarks can generally tell us about many other things. An exhaustive list of all known practical and hypothetical attacks can be built, then a telltale watermark that can distinguish between a subset of these attacks according to the application at hand can be designed. A special case of telltale watermarks is a *localization watermark* that can tell us which times (for audio content) and regions (for video and images) that were distorted.

4.3.2 Error Diffusion

A very important technique that could be used with most watermarks to alleviate the visual distortions introduced by the watermark bits was proposed in [58]. As each pixel value is altered, a fraction of the error introduced to that pixel is distributed to its neighboring pixels that have not yet been modified. The effect of that operation is usually a watermarked image with less visual artifacts.

One major problem with localization watermarks is that they are usually implemented as separate block-wise independent watermarks. Block-wise watermarks are vulnerable to many attacks [11]. There are two ways to avoid these attacks: (1) as proposed in [59], one can make the signature watermark embedded in each block dependent on some surrounding data (this will make attacks on such watermark much more complex), or (2) one can use variable block sizes. The first n blocks have a fixed size defined by the embedding algorithm. When the watermark bits in these blocks are extracted, they will tell the detector about the sizes and locations of the remaining blocks. We may experiment with this technique in future research.

4.3.3 Self-Embedding

Self-embedding is a technique to embed a highly compressed version of the image as a watermark. The embeddor makes sure the compressed version of each image region is embedded in a different region of the image [60]. Many ideas can be implemented to improve that watermark and solve some of its problems: (1) One can use spatial spread-spectrum techniques to implement this algorithm so as to increase its robustness. For a review of spread-spectrum watermarking techniques and possible attacks, refer to [61]. A good spread-spectrum watermarking system was presented by Cox et al. in [62] and [63]. (2) One can merge the techniques in [60] and [64] to get a telltale watermark that has a self-restoration capability. (3) Error Correction Coding (ECC) can be used to improve the performance of the algorithm. (4) A serious problem with this watermark is that there is no automatic technique that would distinguish whether an attacker had manipulated the LSBs of the compressed block or replaced the six most significant bits of the block itself. The only current solution to this problem is through visual inspection by a human operator.

4.3.4 Domain-Visible Watermarks

In [65], a DWT-based watermark is proposed. The new idea in that system is that when the detected watermark is transformed into the wavelet domain and decrypted, it can be displayed as a grayscale image. This idea can be used in a courtroom to give more evidence that someone is the owner of a digital image.

4.4 The Basic Telltale Watermarking Algorithm

The basic telltale watermark that was implemented and enhanced as part of this work, based on [17] and [9], is described in this section. We also discuss its advantages and weaknesses and the reasons we believe it is a promising algorithm.

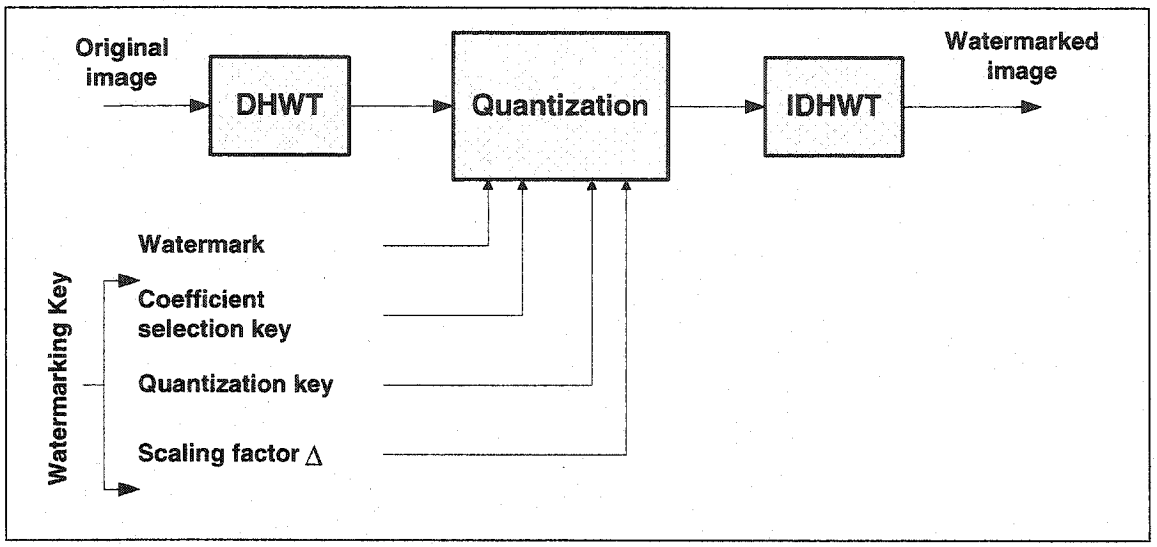


Figure 4.12 Embeddor Block Diagram [9].

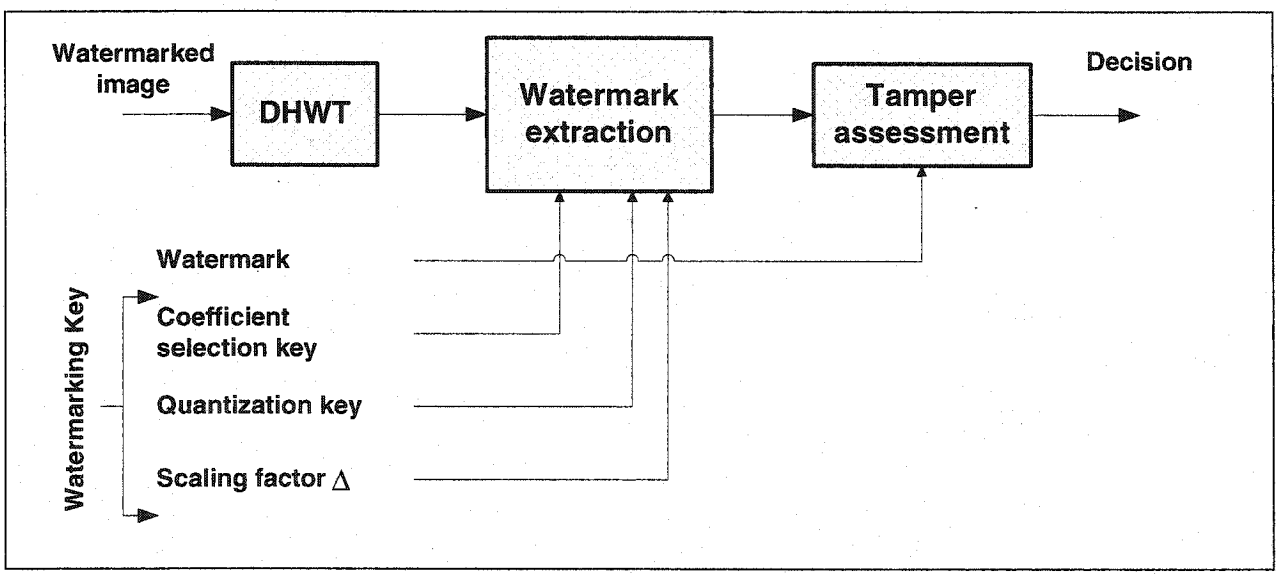


Figure 4.13 Extractor Block Diagram [9].

The watermark is a random sequence of bits that is embedded in the coefficients of the image's Discrete Wavelet Transform (DWT). The DWT domain is arguably preferred to spatial, DCT, or Fourier domains to embed the watermark data. The reason is that, by nature, the wavelet analysis provides both spatial and spectral information about the signal [56]. The embedded watermark will be spread over the complete spatial range of the image signal allowing localized tampering detection. Moreover, the spectral localization of the wavelet transform will allow frequency spread of the watermark signal. This should make the watermark sensitive to large-scale signal distortions.

By designing a watermark that holds information about both spatial and frequency domains, it is now up to us to interpret changes that the watermark undergoes and draw conclusions about the type and strength of tampering that it suffered.

4.4.1 Haar Wavelet Decomposition and Reconstruction

The simplest of all the wavelets is Haar wavelet which is shown in Figure 4.14 below. This wavelet resembles a step function and is a special case of the Daubechies family of wavelets (db-1) [57]. Although Haar wavelet is not popular in other applications, it was chosen because all calculations use fixed-point arithmetic. This means that computational complexity is lower, and it also means that there is a lower chance of rounding errors. This is an important advantage because it directly reduces the algorithm's false alarm rate.

The image signal is decomposed to an arbitrary number of wavelet decomposition levels, L which is one of the algorithm's parameters.

Using a large L makes embedding and detection slower but increases the watermark robustness against global image attacks (such as applying a filter on the whole image).

Using a small L has the disadvantage that the watermark is only embedded in the high-resolution wavelet coefficients which means that it will be more sensitive to any minor modifications of the image pixels.

An interesting observation is that choosing to start with $L > 1$ (by skipping one or more of the higher resolution levels) increases the watermark robustness because all the watermark bits are now embedded in the lower frequency sub-bands of the image signal. These sub-bands hold most of the image signal's energy. A disadvantage is an increased chance of creating visual artifacts in the image, specially when the watermark energy is high compared to that of the original image. These artifacts will appear as subtle low-frequency brightness changes which are more visible at smoother areas of the image. To illustrate this, see Figure 4.15 where a high energy watermark is embedded by quantizing coefficients of the fifth decomposition level.

$$w(t) = \begin{cases} 1 & for\ 0 \le t < 0.5 \\ -1 & for\ 0.5 \le t < 1 \\ 0 & otherwise. \end{cases}$$

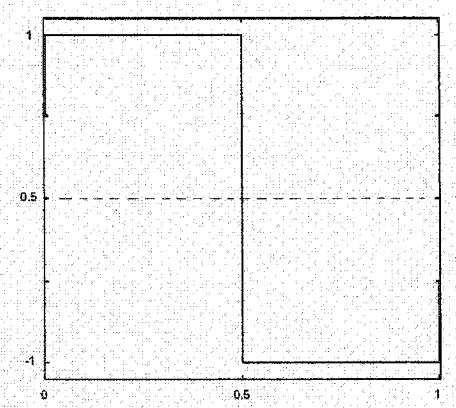


Figure 4.14 Haar wavelet.

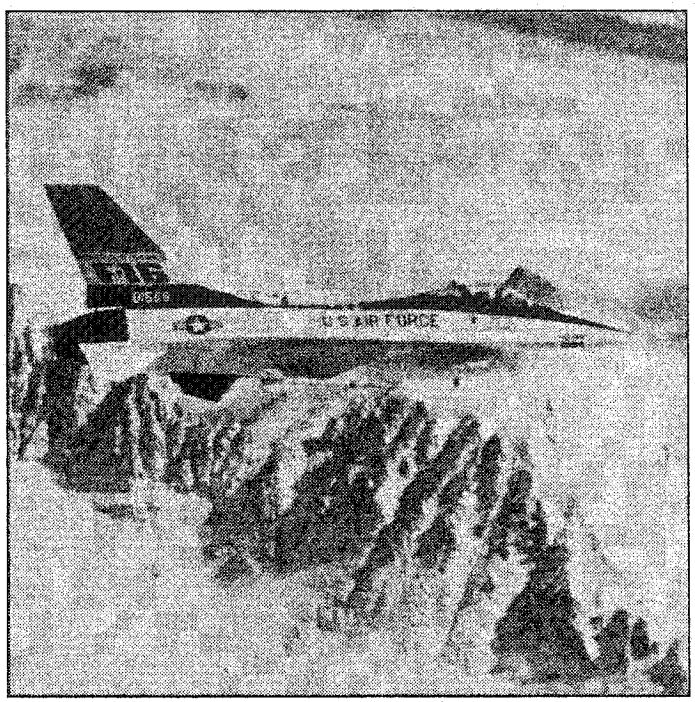


Figure 4.15 Visual artifacts resulting from changes to coefficients of lower-frequency fifth wavelet decomposition level.

Al-Mohimeed [66] modified the algorithm so that it can be used for robust watermarking by choosing to embed the watermark into specific low-frequency wavelet decomposition levels.

4.4.2 Algorithm Description

Kundur et al. [9] proposed a semi-fragile watermarking algorithm that can be used for content authentication and tamper detection of still images. The algorithm embeds the watermark bits into the discrete wavelet domain. This allows for the detection of image changes in both localized spatial and frequency domains.

The watermark detector compares the detected watermark bits with the corresponding embedded watermark bits. By analyzing the difference, the detector can make a decision about whether or not any modification had occurred in specific regions of the image. The

detector can also analyze the percentage of watermark bits that were not detected correctly for each wavelet frequency band. This analysis can lead to decisions regarding the type of distortion that a watermarked image had undergone, and whether or not there was an attempt to tamper with the image contents.

In the following sections, we will describe the algorithm in detail. Discrete Haar wavelets are used as the domain into which the watermark bits are embedded. Haar wavelets have low complexities as they use fixed-point arithmetic. We have experimented with other wavelets types (Daubechies wavelet db2) in our simulations and preliminary results show no significant improvement over the Haar wavelets. It should be noted that Haar wavelet are the simplest form of wavelets in the Daubechies wavelet family where they are referred to as db1.

The telltale watermarking algorithm can be broken down into a number of building blocks. Both watermark embeddor and extractor accomplish their task using some of these building blocks.

Step 1 - Discrete Haar Wavelet Transform

The L^{th}-level discrete Haar wavelet transform (DHWT) is computed for the original image f. This process generates one approximation of the image at the coarsest resolution level along with *3L* detail images corresponding to the horizontal, diagonal, and vertical details at each of the L resolution levels. The value of L is arbitrary. In our simulations, we experimentally choose a suitable range of values for L. Let $f_{o,l}(m,n)$ denotes a Haar wavelet coefficient of the image f at decomposition level l where

$l \in \{1,2,\ldots,L\}$ and $o \in \{1,2,3\}$ corresponding to the horizontal, diagonal, and vertical image details respectively.

Step 2 - Embed Watermark Bits

The watermark bits are embedded into the wavelet coefficients by quantization of these coefficients according to the following binary mapping function:

$$Q_{\Delta,l} = \begin{cases} 0 & if \left\lfloor \frac{f}{2^l \Delta} \right\rfloor is\ even \\ 1 & if \left\lfloor \frac{f}{2^l \Delta} \right\rfloor is\ odd \end{cases}, \qquad (4.1)$$

where Δ is a positive real number representing the watermarking scaling factor that is used to amplify or attenuate the watermark energy. $\lfloor \cdot \rfloor$ is the floor function. The mapping function simply maps real values to binary numbers.

Equation (4.1) suggests that the higher the decomposition level, and the lower the frequency band, the coarser the quantization. This supports the argument that to increase watermark robustness, more energy is needed to be embedded into the lower frequency range.

For an $2^N \times 2^N$ image that is decomposed using L wavelet decomposition levels, the total number of wavelet coefficients is also $2^N \times 2^N$. Out of these coefficients, the number of coefficients used for image approximation is $2^{N-L} \times 2^{N-L}$. The watermark bits are only embedded into the details coefficients, so the approximation coefficients are not used for watermark embedding. The reason for this is that most of the image's energy is carried by the approximation coefficients. Any slight change to these coefficients will

potentially have a significant impact on the image's visual quality. This is analogous to the DC coefficients of the DCT used with JPEG image compression.

The number of coefficients that can be used to embed the watermark bits is thus $2^N \times 2^N$ minus $2^{N-L} \times 2^{N-L}$ giving $4^N - 4^{N-L}$. To reduce the number of watermark bits required for tamper detection, one does not have to watermark all details coefficients. Instead, we only use one third of the detail coefficients for watermark bits embedding. To increase the security of the watermark and to guarantee that the watermark is spread evenly throughout the image, both spatially and spectrally, a randomly generated coefficient selection key k_c is used to choose one detail coefficient for every particular resolution and spatial location. A typical coefficient selection key k_c will have the following values {3,2,2,1,3,1,3,2,3,2,1,2,3,2,2,3,2,1,etc., ...}.

As mentioned above, $o \in \{1,2,3\}$ corresponding to the horizontal, diagonal, and vertical image details respectively. The watermark bit, $w(i)$, is embedded by quantizing the coefficient specified by $k_c(i)$. This means that the length of k_c must be equal to the number of watermark bits.

It is also important to note that if synchronization of k_c with $w(i)$ at the detector is lost, the watermark will not be extracted correctly. This implies that the algorithm may not be immune against RST attacks. To improve the watermark resistance to such attacks, more watermark energy must be embedded into the coarser decomposition levels. This requirement is more important for robust watermarking algorithms.

The total number of detail coefficients used to embed the watermark is thus $\frac{4^N - 4^{N-L}}{3}$.

For relatively large values of L, this approximates to one third of the number of image pixels.

Each watermark bit is embedded into its assigned coefficient according to the following rule:

If $Q_{\Delta,l}[f_{o,l}(m,n)] = w(i) \oplus k_q[f_{o^*,l}(m,n) \times 2^l]$, no change in the coefficient is required, *(4.2)*

otherwise, set $f_{o,l}(m,n) = f_{o,l}(m,n) - 2^l \times \Delta \times \mathrm{sgn}[f_{o,l}(m,n)]$,

where $o^* = (o+1) \bmod 3 + 1$, $\oplus$ is the XOR operator, and

$$\mathrm{sgn}(f) = \begin{cases} 1 & \text{if } f \geq 0 \\ -1 & \text{otherwise} \end{cases}$$

The quantization key, k_q, is implemented as an arbitrary secret function that maps every possible amplitude of the wavelet coefficients to a binary number. This image-dependent binary number is then XORed with the watermark bit to generate a bit value that is based both on the watermark bit and on the image wavelet coefficients. The final bit value is unknown to the public and thus can be used to prevent forgery or impersonation attacks. k_q guarantees, with a probability close to one, that the watermark can not be extracted from one watermarked image and duplicated into another image to make a forged watermarked image.

It is worth noting that in equation (4.2) above the coefficient at index $o^* = (o+1) \bmod 3 + 1$ was used as an input to the quantization key rather than the coefficient at index o. The reason is that the latter will be, potentially, quantized as a result of the watermark bit embedding. So, at the extractor, when this coefficient is fed into the quantization key secret function, it will have a different value and consequently it will cause $k_q(\cdot)$ to generate a different value. This will cause the watermark extraction to fail even for a watermarked image that had never undergone any modifications. As described earlier, at every coefficient index, only one detail coefficient out of the three potential coefficients (horizontal, diagonal, and vertical) is chosen to be modified. This way, one is guaranteed, as long as the image is not attacked, that $o^* = (o+1) \bmod 3 + 1$ will have the same value at the extractor side and thus $k_q(\cdot)$ will generate an output identical to that was generated at the embeddor, leading to successful watermark detection.

While using $k_q(\cdot)$ increases the algorithm security, it can also make the watermark very sensitive to even the minor changes to the values of the coefficients. To avoid this weakness, one can design the function $k_q(\cdot)$ such that it is dependent on features of the image, but not directly dependent on the exact value of the coefficients. Two examples are:

- Choose $k_q(\cdot)$ such that it is directly dependent on the mean value of each 8×8 block of the image. If, for any of the watermark bits, the corresponding mean value is changed, the extracted watermark bit will not be successfully detected with a high probability.

- $k_q(\cdot)$ can be designed to be dependent on the relative value of the coefficient to its neighboring coefficients. Then, if the whole image suffered from some global minor distortion, the coefficients will still maintain their relative values and thus the watermark bits will be successfully extracted.

Figure 4.16 illustrates the watermark binary mapping function $Q_{\Delta,l}$ described in equation (4.2). The binary mapping function maps each possible range of the wavelet coefficients f into a binary number. Increasing the value of Δ means that the width of each range also increases and thus, in case the image was attacked, there is a higher probability that a modified coefficient will still lie within the same range. This is equivalent to increasing the resistance of the watermark to attacks. It also means that the watermark sensitivity decreases. Tuning the value of Δ is crucial to the performance of telltale algorithm as will be shown in our simulations.

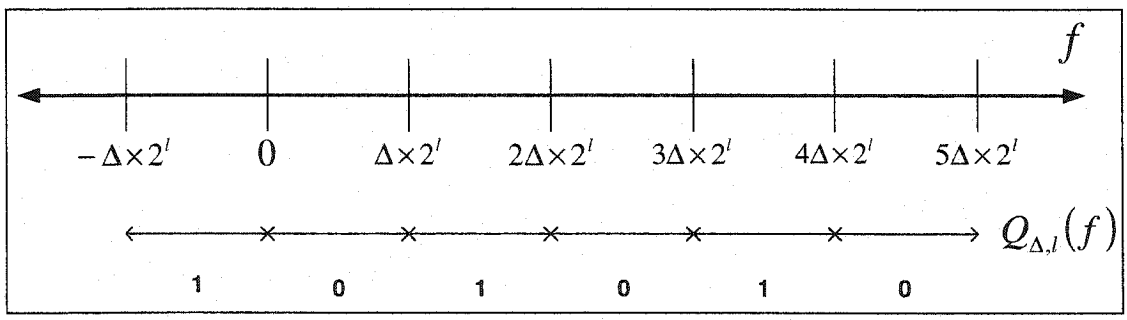


Figure 4.16 Telltale binary mapping function of detail wavelet coefficients.

Figure 4.16 illustrates the watermark embedding function in equation (4.2), $f_{o,l}(m,n) = f_{o,l}(m,n) - 2^l \times \Delta \times \text{sgn}[f_{o,l}(m,n)]$. Effectively, this function simply moves a coefficient value from one range to an adjacent range. The choice of whether to move it to the right neighbor range or to the left neighbor range is arbitrary as both ranges will generate the same value of $Q_{\Delta,l}$. The authors of the algorithm opted to shift the

coefficient value by the amount of $-2^l \times \Delta \times \text{sgn}[f_{o,l}(m,n)]$ which indicates that a positive coefficient will have its value reduced by that amount while a negative coefficient will have its value increased by the same amount. The algorithm authors used that approach as they experimentally have found that it generates the least visual degradation for a given magnitude of Δ.

Step 3 – Extract Watermark Bits

The watermark extractor has no access to the original unwatermarked image. It only have access to the watermarking key along with the embedded watermark itself. This key is composed of the following parameters: Δ (watermark energy), k_c (coefficient selection key), and k_q (quantization key). Representation of these keys is implementation dependent. Fewer watermark bits can be embedded by only watermarking every second detail coefficient so that only one sixth of the detail coefficients are modified.

To extract a watermark bit $\hat{w}(i)$, the following equation is used:

$$\hat{w}(i) = Q_{\Delta,l}[f_{o,l}(m,n)] \oplus k_q\left(f_{o^*,l}(m,n) \times 2^l\right) \qquad (4.3)$$

The watermark extractor should compare the embedded watermark w to the extracted one $\hat{w}$ in order to assess any tampering to the watermarked image. The simplest Tamper Assessment Function (TAF) will have the form [9]:

$$TAF(w,\hat{w}) = \frac{1}{N_w}\sum_{i=1}^{N_w} w(i) \oplus \hat{w}(i), \qquad (4.4)$$

where N_w is the length of the watermark.

$TAF(w,\hat{w})$ can take any value between 0 and 1 with 0 indicating no tampering and 1 indicating that the extracted watermark is orthogonal to the embedded one.

More detailed tamper assessment can be performed in two ways:

1. To detect filtering and JPEG compression, one can measure the percentage of watermark bits that were extracted in error for each decomposition level. If higher error rates are detected in higher frequency decomposition levels, this indicates that the watermarked image has undergone either JPEG compression or low-pass filtering. Higher extraction error rates in lower frequency decomposition levels indicate that sharpening filtering was applied to the image. Generally, if some spectral effect can be characterized by its effects on various frequency ranges, then this effect can be detected by the telltale algorithm.

2. An attacker may attempt to modify a specific spatial area of the image. This can also be detected by the telltale watermark as each wavelet coefficient covers a specific spatial area of the image. At the highest frequency decomposition level, each coefficient corresponds to an area of size 2×2 pixels. Each coefficient in the next decomposition level covers a specific 4×4 pixels area.

Combining both spectral and spatial analyses of the detected watermark, one can come to a conclusion whether or not the watermarked image has been tampered with. Moreover, one can also detect the nature of distortions that the image has undergone due to various transmission errors, or due to lossy compression and other incidental distortions.

4.4.3 Algorithm Performance

Figure 4.17 shows an original 256×256 pixels image, a watermarked image with $\Delta=1$, and a watermarked image with $\Delta=3$. In both watermarked images, we used 3 levels of decompositions to embed the watermark bits.

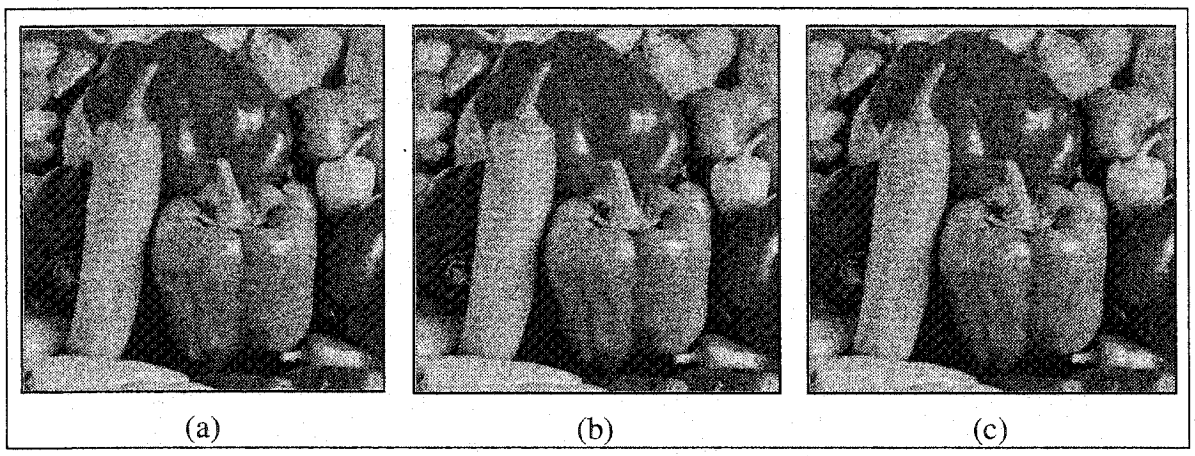


Figure 4.17 "Pepper" image, (a) Original, (b) $\Delta=1$, and (c) $\Delta=3$.

Figure 4.18 shows a doctored image where one of the peppers was removed. All other pixels of the image were not modified. It also shows the difference bitmaps for the three decomposition levels. We generate these difference bitmaps in order to facilitate the visual inspection of the watermark extractor output.

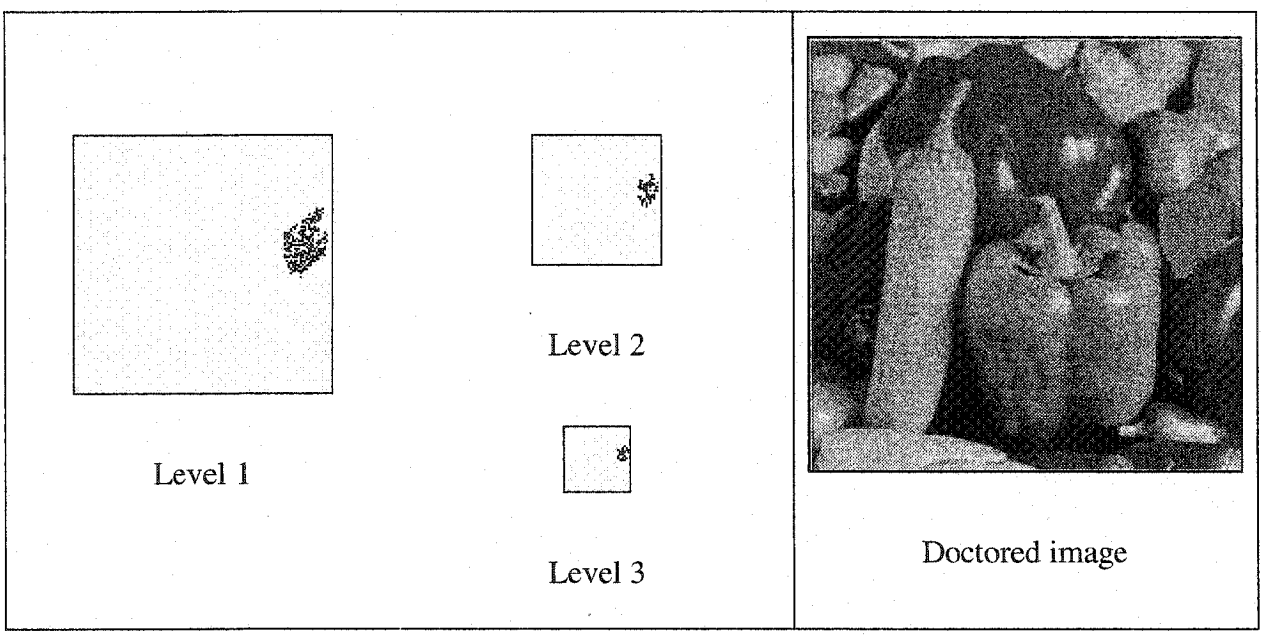


Figure 4.18 Tampering of the image being detected by the watermark detector.

A difference bitmap for each level is generated using the following algorithm shown as pseudo-code in Table 4.2:

Table 4.2 An algorithm to generate difference bitmaps.

```
START
;WE    = Embedded watermark
;WX    = Extracted watermark
;DIFF  = An array equal in size to the watermark
;W(i)  = Bit i of the watermark W
For all watermark bits ( I=1 to N)
        If WE equals WX
                DIFF(i) = 255 ; WHITE PIXEL
        Else
                DIFF(i) = 0 ; BLACK PIXEL
End For
Convert DIFF from a vector to a 2D array
END
```

The reason why some of the attacked pixels in the difference bitmap shown above have white color, indicating no error, is due to two factors:

1. some of the pixels in the attacked regions are not changed at all, and

2. because the algorithm uses quantization of wavelet coefficients to embed and detect watermark bits, a modified pixel can just happen to have the right value so that it will not cause an extraction error.

It is still difficult for the attacker to adjust pixel values such that the doctored image can fool the watermark extractor. The reason for that is because the attacker has no knowledge of where the watermark bits are in the pixels, and also because detection happens in the wavelet domain, so it will be very difficult for the attacker to adjust the wavelet coefficient values accurately.

4.5 Summary

In this chapter, we introduced the problem of telltale tamper proofing. We discussed various types of modifications and attacks that semi-fragile watermark designers should account for. We reviewed various techniques used for tamper detection along with various types of fragile and semi-fragile watermarks.

We also describe a selected telltale watermarking algorithm that used the DHWT. The features and potential of the algorithm were highlighted, and its weaknesses and possible improvements were discussed.

5 Experimental Results

5.1 Introduction

In this chapter, we evaluate the performance of telltale watermarking through several simulation sets. The first simulation set involves embedding watermarks into ten test images using different embedding parameters, followed by measuring the effect of the watermark signal on the host image's visual quality. We use three different quality metrics, PSNR, wPSNR, and the Watson metric. In that simulation, we also attempt to evaluate the limitations and advantages of each of these metrics. In the second simulation set we attempt to evaluate the effect of tuning the watermark embedding parameters on the extractor performance and on the visual quality of the watermarked image. This simulation is carried out using parameters that obey the allowed visual quality limits resulting from the first simulation. Another simulation set is dedicated to analysis of watermark resistance against JPEG compression as a common incidental distortion. The last two simulation sets measure watermark resistance against mild rotation and additive Gaussian noise respectively.

We also introduce a modified version of the embeddor that takes advantage of our perceptual metrics simulations to improve the watermark robustness to incidental distortions while keeping visual artifacts within acceptable levels.

5.2 Test Setup

Our simulations use C++ language to analyze the test images as well as Matlab™ language to embed, extract, and analyze telltale watermarks. We also partially use the Checkmark benchmark tool to verify our results.

We use ten grayscale 512×512 pixel images with *256* grayscale levels per pixel. We choose to only use grayscale images without loss of generality as the algorithm can be used with color images. The algorithm modifies the pixel intensities, which is directly applicable to grayscale images without prior preparation. In the case of color images, we would have to convert the image from the RGB color model, where the luminance has no separate channel, into a suitable model such as the YIQ one, where Y represents luminance of the image pixels [34]. We can then apply the algorithm to the Y component.

Figure 5.1 shows the ten test images we use. We select the images such that they have various features. For instance, while "Bear" image has heavy textures and a dark area near the lower center, we can see that "Watch" image has sharp edges and many smooth areas.

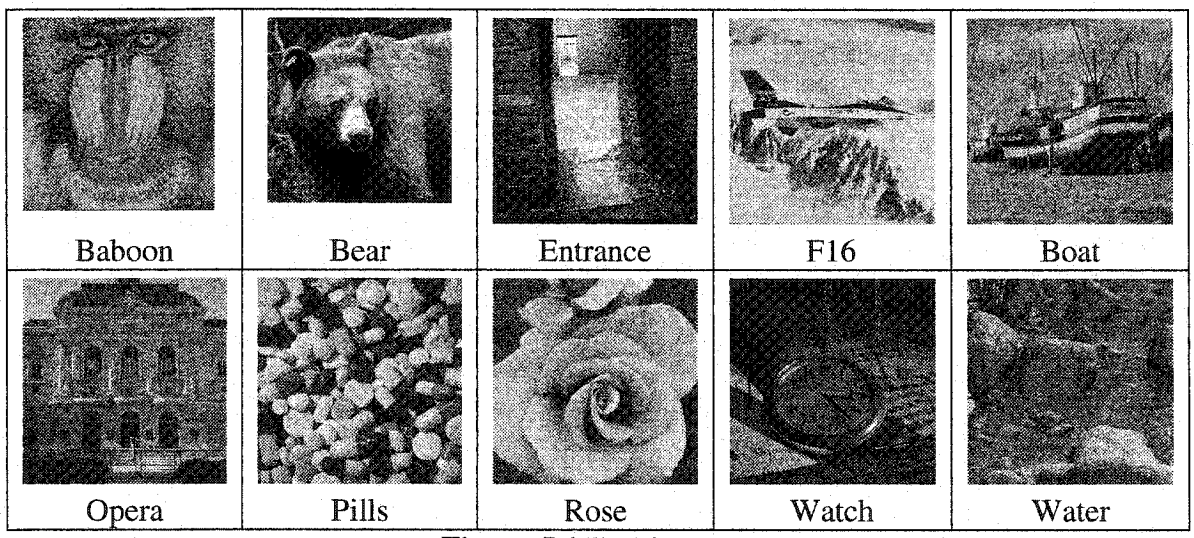


Figure 5.1 Test images.

5.3 Algorithm Parameters

Our implementation of the telltale watermarking algorithm allows for tuning of some or all of the following parameters:

1- Quantization level Δ:

We varied this parameter to all values from 1 to 20 to measure the effect of watermark embedding on the visual quality of images. The higher this value, the higher the distortion of the image. In our simulation set, we try to adjust this parameter so that the trade-off between visual quality and robustness is quantified.

2- Starting wavelet decomposition level L_{start}:

The starting wavelet detail decomposition level can vary from one (generating 256×256 wavelet coefficients), and up to nine (generating one wavelet coefficient).

3- Ending wavelet decomposition level L_{end}:

The ending wavelet detail decomposition level has to be equal to or greater than the starting one. In one of our simulations, we also use a fixed number of levels ($L_{end} - L_{start} = C$), where we choose $C = 3$ meaning that four decomposition levels coefficients are modified to embed the watermark data.

5.4 Simulation Results

5.4.1 Simulation Set 1:

In this test set, we embed a watermark in each of the ten test images. We repeat the process using the same starting decomposition level (level one) while incrementing the ending decomposition level (from one to eight). The more levels used to embed the watermark bits, the more total distortion of the original image. However, using more decomposition levels to embed the watermark bits means that the total watermark energy will be higher. It also means we can get more information about any attacks. We use $\Delta = 1$ for all tests in simulation set 1.

5.4.1.1 Peak Signal to Noise Ratio (PSNR):

Table 5.1 shows the effect of embedding watermark bits into more decomposition levels on the PSNR of the original image compared to the watermarked one.

Table 5.1 PSNR (dB) values for watermarked images using different decomposition levels.

	L1-L1	L1-L2	L1-L3	L1-L4	L1-L5	L1-L6	L1-L7	L1-L8
Baboon	51.10	48.17	46.39	45.16	44.26	43.47	42.77	42.08
Bear	51.23	48.28	46.51	45.30	44.26	43.53	42.95	42.46
Entrance	51.13	48.17	46.47	45.18	44.24	43.47	42.72	42.19
F16	51.06	48.14	46.39	45.18	44.23	43.36	42.60	42.19
Boat	51.05	48.17	46.45	45.21	44.15	43.44	43.01	42.13
Opera	51.09	48.16	46.39	45.12	44.03	43.43	42.68	41.70
Pills	51.10	48.15	46.40	45.15	44.12	43.44	42.53	42.36
Rose	51.07	48.15	46.37	45.17	44.15	43.27	43.06	42.46
Watch	51.13	48.15	46.41	45.14	44.16	43.60	42.67	41.88
Water	51.06	48.16	46.38	45.18	44.22	43.52	42.88	42.44

Figure 5.2 indicates how the quality of the image degrades when more decomposition levels are used for embedding. It is observed that the rate of quality degradation is a decreasing one. This is mainly due to the fact that higher decomposition levels (corresponding to lower frequency ranges) have less coefficients which means that less watermark bits can be embedded into them. Table 5.2 depicts the number of coefficients for each one of the wavelet decomposition levels for the 512×512 pixels images used. Note that we only use the details levels (horizontal, vertical, and diagonal) to embed the watermark bits. Approximation coefficients are not used as the image quality is very sensitive to the slightest change in them. They correspond to the DC components of the DCT of an image.

Table 5.2 Wavelet decomposition levels and their sizes.

Level	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**Image**
Size	256x256	128x128	64x64	32x32	16x16	8x8	4x4	2x2	512x512

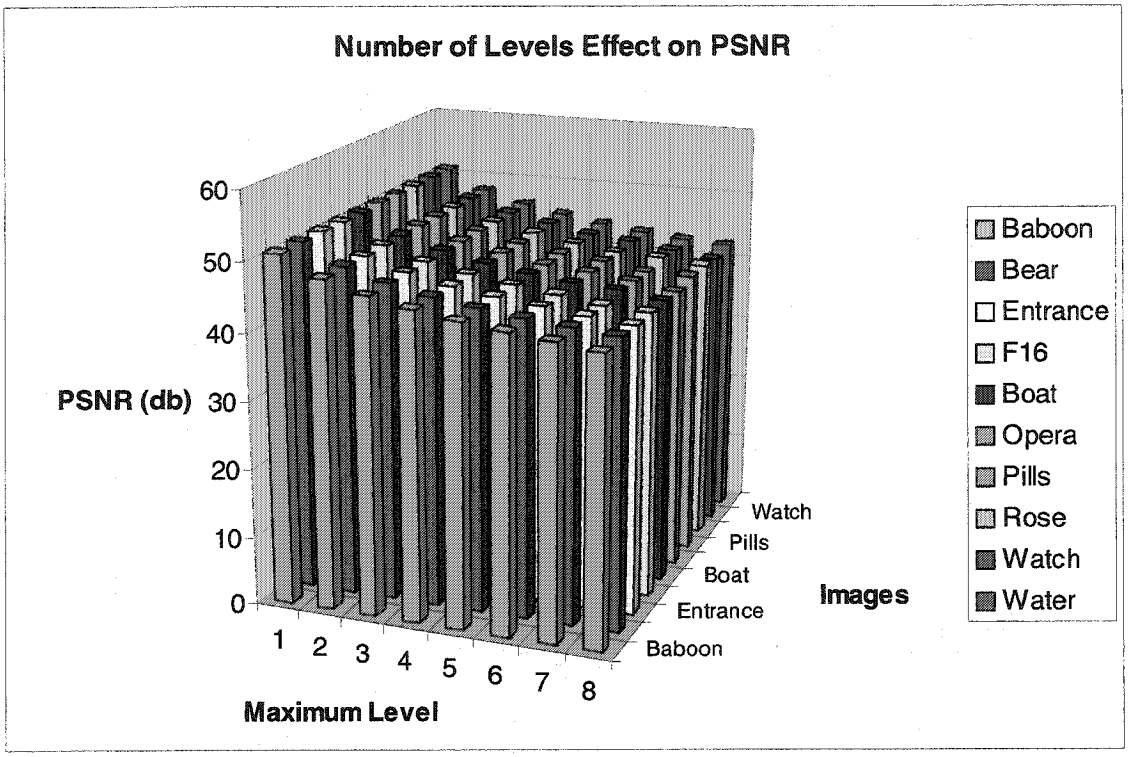


Figure 5.2 The effect of watermark embedding into more decomposition levels on PSNR (dB) values of watermarked images.

5.4.1.2 Weighted Peak Signal to Noise Ratio (wPSNR):

The weighted PSNR (wPSNR) is a different quality metric that was suggested in [67]. The wPSNR metric uses a new parameter called the Noise Visibility Function (NVF) which is a texture masking function. NVF uses a Gaussian model to estimate how much texture exists in any area of an image. The value of NVF ranges from close to zero, for extremely heavily textured areas, and up to one, for clear smooth areas of an image. The equation to calculate wPSNR is a modified version of the regular PSNR equation, but it uses the value of VNF as a penalization factor as we see in the following equation.

$$wPSNR = 10 \log_{10}\left(\frac{255}{RMSE \times NVF}\right)^2 \qquad (5.1)$$

For flat, smooth areas, NVF is equal to one, which represents the maximum penalization. In that case wPSNR has the same value as PSNR. For any value of NVF less than one, the value of wPSNR will be slightly higher than that of PSNR to reflect the fact that human eyes will have less sensitivity to modifications in textured areas than to changes in smooth areas. Imagine a picture of a flat white wall with one small gray spot. The spot will be immediately noticed by any human observer.

Table 5.3 depicts the effect of embedding watermark bits into more decomposition levels on the wPSNR of the original image compared to the watermarked one.

Table 5.3 wPSNR (dB) values for watermarked images using different decomposition levels.

	L1-L1	L1-L2	L1-L3	L1-L4	L1-L5	L1-L6	L1-L7	L1-L8
Baboon	55.94	53.01	51.24	50.06	49.05	48.35	47.58	46.98
Bear	52.75	49.79	48.01	46.79	45.75	45.00	44.42	43.91
Entrance	53.51	50.52	48.83	47.55	46.59	45.79	45.03	44.48
F16	52.26	49.35	47.59	46.36	45.41	44.56	43.76	43.46
Boat	53.22	50.36	48.64	47.40	46.32	45.61	45.12	44.29
Opera	53.40	50.46	48.69	47.45	46.34	45.76	45.00	43.89
Pills	52.33	49.41	47.65	46.39	45.35	44.65	43.72	43.53
Rose	51.80	48.88	47.09	45.89	44.85	43.99	43.79	43.15
Watch	52.56	49.56	47.81	46.53	45.55	45.01	44.20	43.29
Water	53.24	50.34	48.57	47.37	46.41	45.73	45.10	44.63

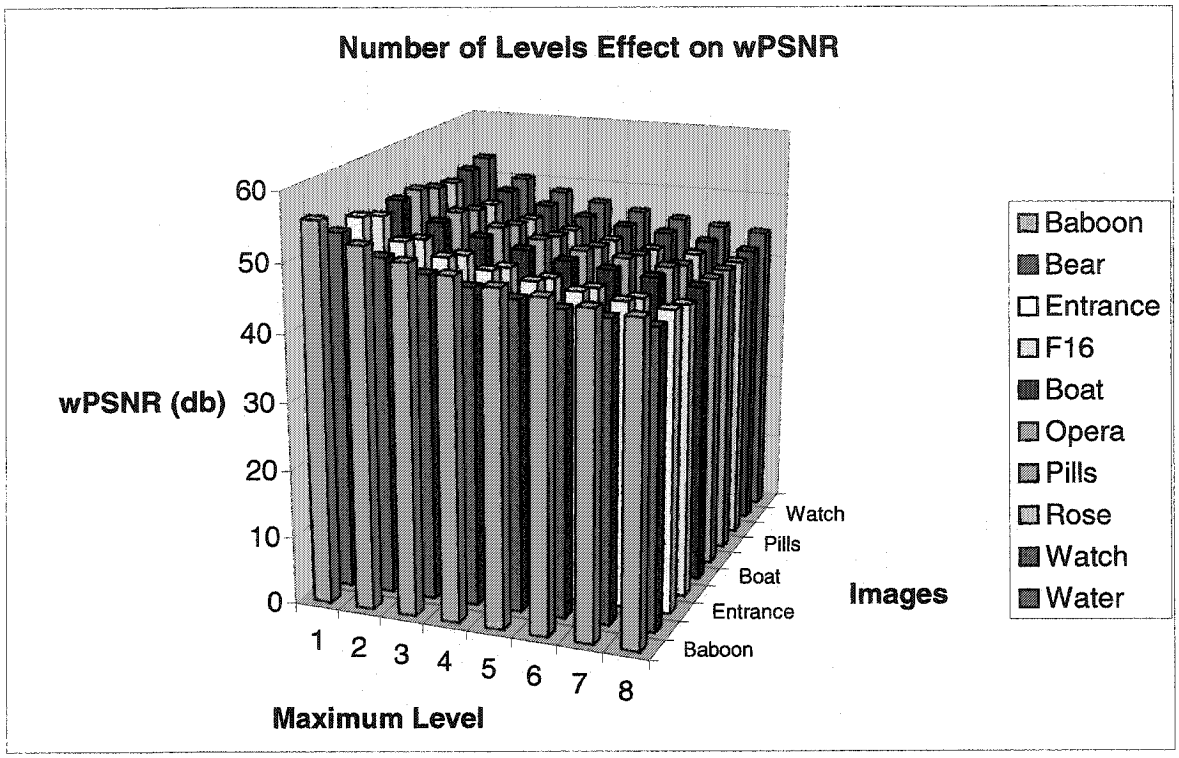


Figure 5.3 Effect of watermark embedding into more decomposition levels on wPSNR (dB) values of watermarked images.

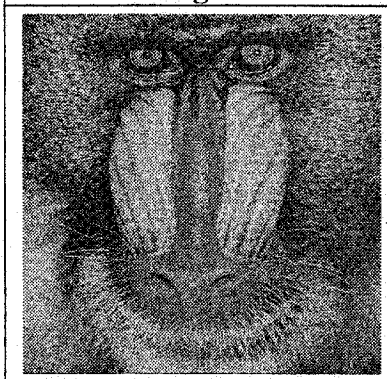
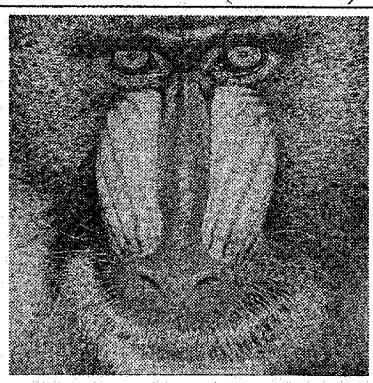
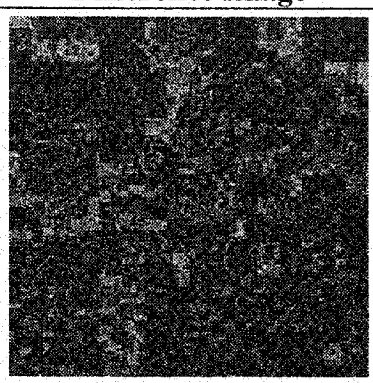
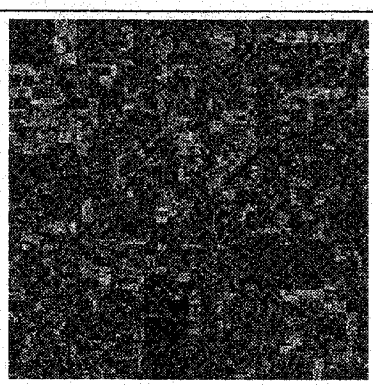
We observe from Table 5.3 and Figure 5.3 that wPSNR for "Bear" image is slightly lower than other images, even though that image has texture in many areas. The reason for that is the large smooth dark area representing the bear's chest. Because this area is black and smooth, any changes to that area will be highly penalized by the NVF, thus causing the quality metric to drop. For "Water" image, we observe that the wPSNR value is significantly higher than the PSNR one. This reflects the heavy details and textures of that image and its capacity to absorb more modifications without showing visual artifacts.

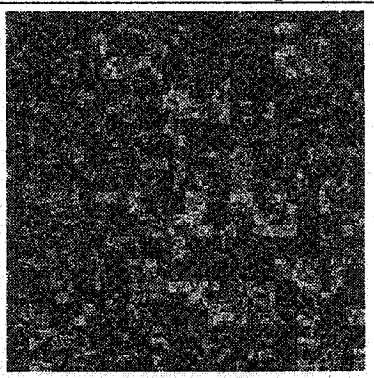
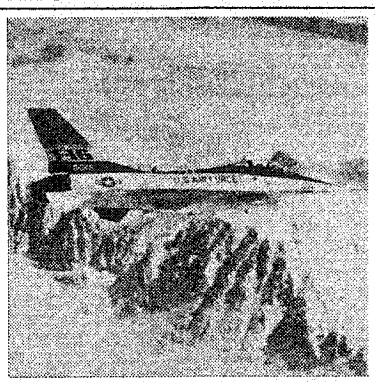
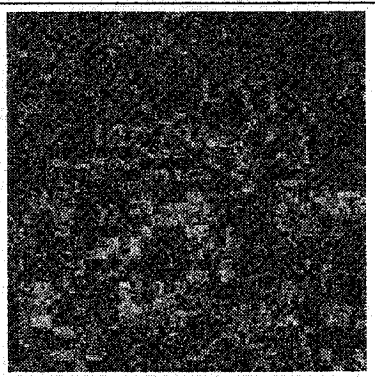
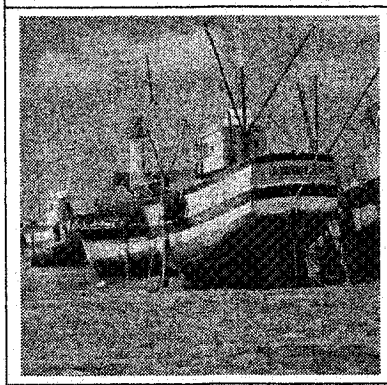
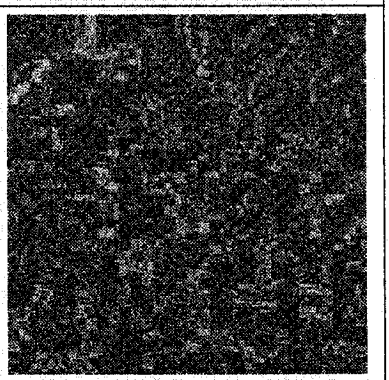
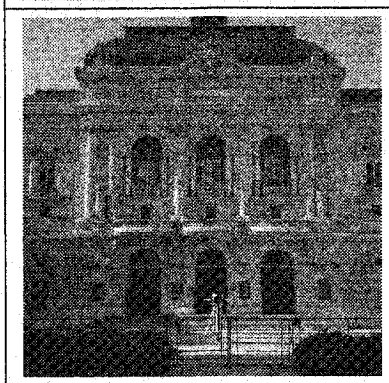
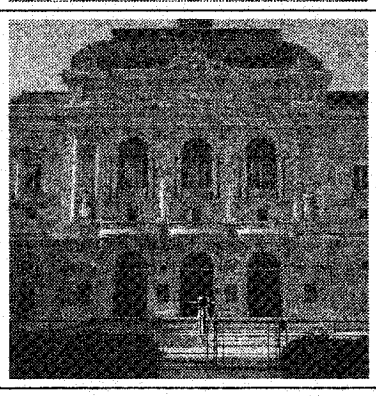
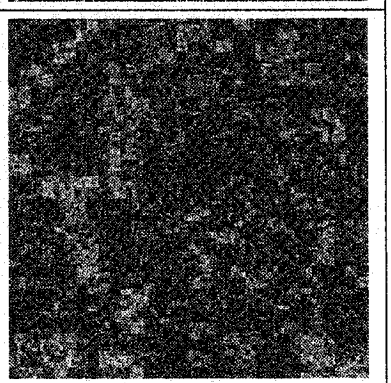
We also observe that even when all decomposition levels are used, the overall quality of the images does not heavily degrade. This indicates that one can use any set of levels to embed the watermark bits. As mentioned before, this is mainly due to the fact that higher decomposition levels have less number of coefficients.

The following set of figures, Figure 5.4, show original images and their corresponding watermarked images with highest energy (from level one to level eight). They also show difference images. These images are the visual representation of the absolute difference between the original images and the corresponding watermarked ones. We used the following equation to generate the pixels of these images:

$$DiffPixel = \min\{[\text{abs}(Im_1\ pixel - Im_2\ pixel) \times 16], 255\} \qquad (5.2)$$

We scale the difference by a scale factor (arbitrarily *16*) to make the difference more visible. We also make sure the scaling factor will not cause any rounding errors by limiting the result pixel value to *255* (maximum allowable luminance). Brighter areas indicate higher watermark energy, while darker areas indicate lower watermark energy.

Original	Watermarked (level 1-8)	Difference Image
		
		

Original	Watermarked (level 1-8)	Difference Image
		
		
		
		

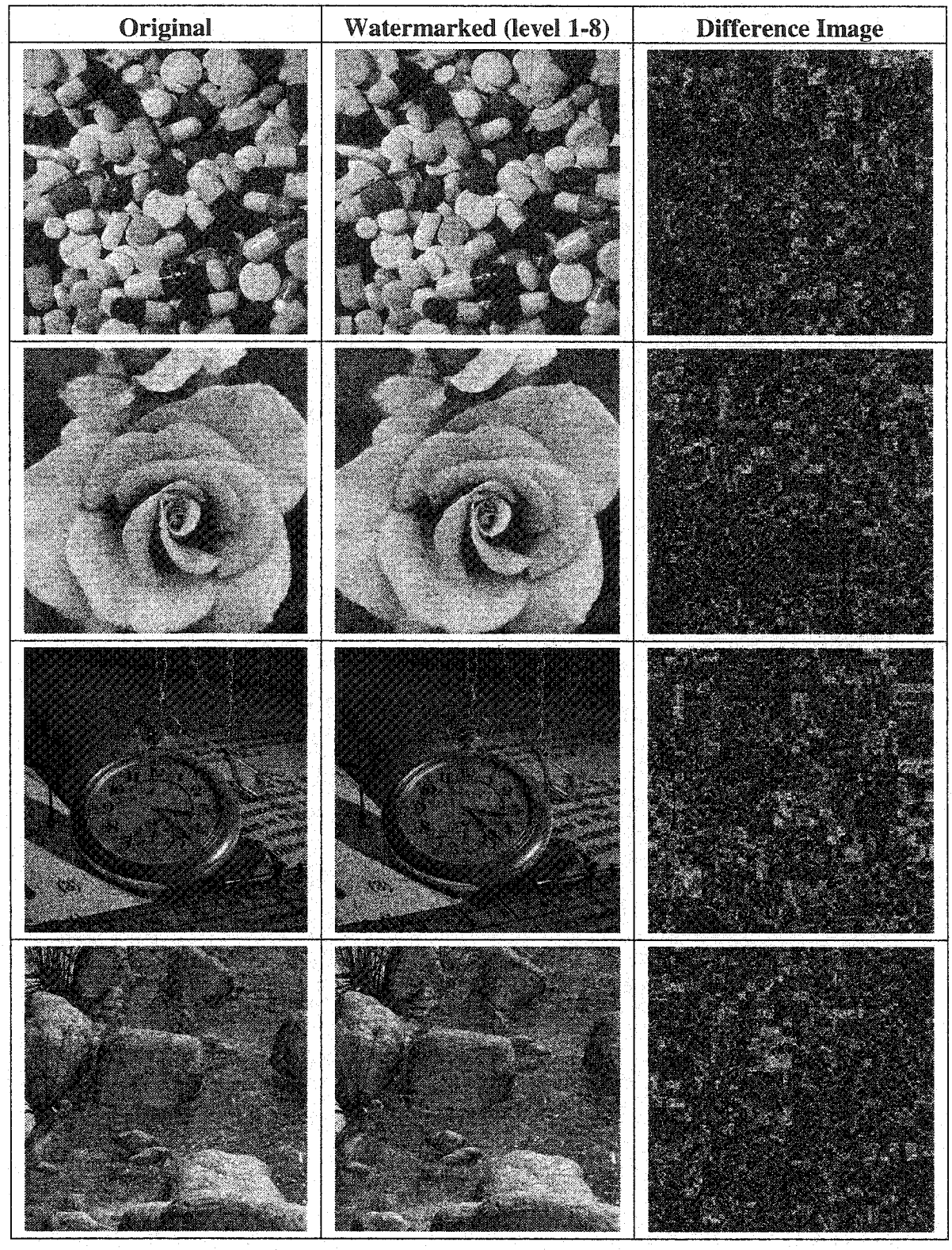


Figure 5.4 Original images, watermarked images with watermark bits embedded in all decomposition levels, and scaled difference image.

5.4.1.3 The Watson Metric:

The Watson metric [32] is discussed in detail in Chapter 3. In this simulation set, the Watson metric is used to generate a perceptually lossless quantization matrix of the DCT transform of each of our test images. The entries of this matrix represent the amount of quantization that each coefficient can withstand without affecting the visual quality of the image. This matrix, referred to as the "visibility threshold matrix", uses three visual aspects in order to model the human visual system: contrast sensitivity, contrast masking, and luminance masking. We choose to build that matrix for blocks of 16×16 pixels each. This decision was guided by the fact that a human face scaled below a 16×16 block is of such a low resolution that recognition becomes impossible [68]. We then compare blocks of the watermarked image with corresponding blocks of the original image using the visibility threshold matrix as a reference in order to judge which blocks have been modified to the extent that the modifications can be visible to humans. We also calculate the average of all block errors and use it as an estimate of the TPE of the watermarked image. As the simulation will show, the TPE is not accurate and it is much better to use the per-block error matrix, LPE, values instead.

We refer to the following parameters of the Watson metric as they are essential for interpreting the results:

1- Local Perceptual Error (LPE):

LPE is a two-dimensional array of values in units of Just Noticeable Differences (JNDs). Since in our simulation we use block sizes of 16×16, the size of the LPE array will be $512\div16=32\times32$ entries. We can scale the values of that array to make suitable for viewing as a bitmap using the following equation, where S is a suitable scaling integer value. The resulting bitmap will have 32×32 blocks where darker blocks represent low visual error and lighter blocks represent high visual errors. By looking at that bitmap, which we can call the Block Visual Error (BVE) bitmap, we can determine which blocks took more visual damage due to the watermark embedding. In our simulation, we use LPE to get an estimate of the visual damage of the watermarked images and, hence, to decide optimal parameters to use with the telltale watermark. A further step would be to use Watson model to embed the watermark bits adaptively such that for blocks that can withstand more quantization, we can use a coarser quantization step to embed the watermark.

$$Pixel = \min(LPE_{entry} \times S, 255) \qquad (5.3)$$

2- Total Perceptual Error (TPE):

The value of TPE gives a rough estimate of the overall visual distortion of the image due to the embedding of the watermark. It is calculated as the mean value of the entries of the LPE array. It should be noted that the value of TPE can be misleading as heavy distortion of few blocks can render the total image visually unusable, while the TPE will still report a total error value that is not sensible. This is because, for instance, blocks of the image that did not suffer any distortion will dilate the total error through averaging. In other

cases, images with smooth areas could report an overall high TPE while in fact only smooth areas have visible artifacts. To give a practical example that supports our argument, we show "Bear" image watermarked with a relatively high quantization step value ($\Delta = 5$). TPE for the watermarked image is *2.852* JNDs, which give a strong indication about the visual damage in the image.

As shown in Figure 5.5, the watermarked image has only heavy visual artifacts in the dark smooth areas of the image, while the textured areas of the image, despite having been modified, are showing much less distortions. Using the BVE bitmap, calculated from LPE matrix, we get a more sensible and accurate estimate about the amount of visual error in every 16×16 pixels block of the image.

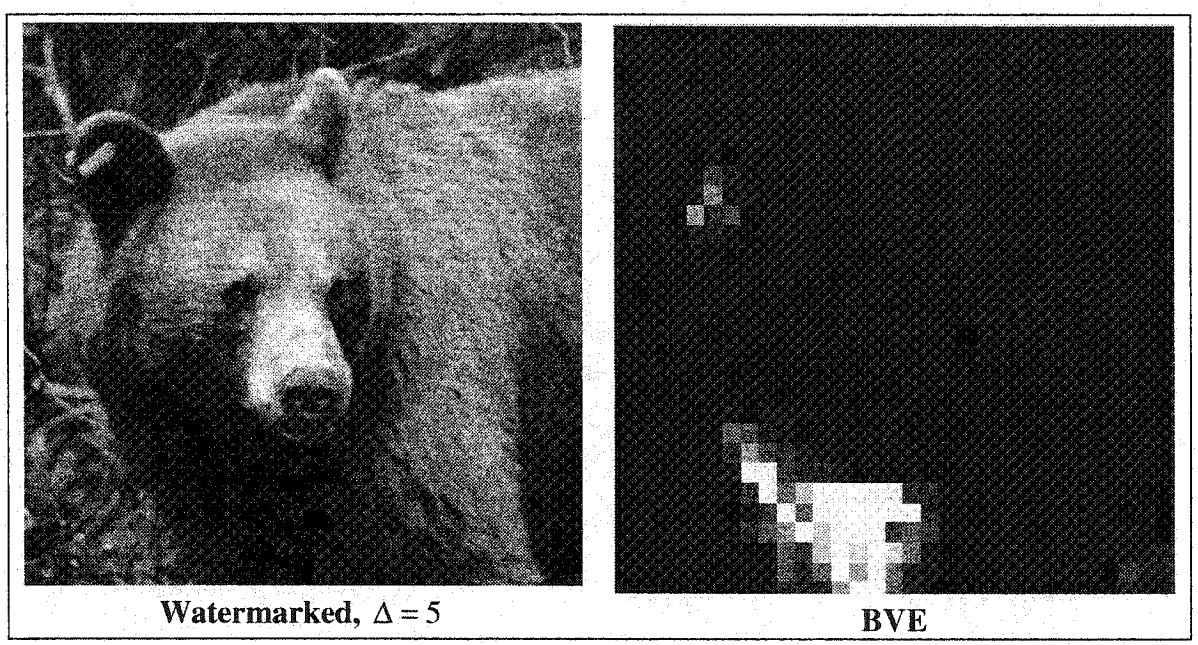


Figure 5.5 Watermarked "Bear" image and its Block Visual Error bitmap.

The total perceptual error values for simulation 1 are shown in Table 5.4. All values are shown to be less than one JND, indicating that the watermarked images have no global distortions and are generally usable. These results do not guarantee that specific areas of

one watermarked image will not have local visual artifacts. To get such a guarantee, one should inspect the LPE matrices.

Table 5.4 TPE in units of JNDs of simulation set 1.

	L1-L1	L1-L2	L1-L3	L1-L4	L1-L5	L1-L6	L1-L7	L1-L8
Baboon	0.11	0.19	0.22	0.23	0.24	0.25	0.26	0.26
Bear	0.30	0.50	0.57	0.60	0.60	0.64	0.62	0.64
Entrance	0.12	0.21	0.23	0.25	0.26	0.27	0.28	0.28
F16	0.10	0.17	0.20	0.21	0.22	0.23	0.23	0.24
Boat	0.12	0.20	0.23	0.24	0.25	0.26	0.27	0.27
Opera	0.12	0.21	0.24	0.26	0.27	0.28	0.29	0.29
Pills	0.11	0.19	0.21	0.23	0.24	0.25	0.26	0.26
Rose	0.11	0.20	0.22	0.24	0.25	0.26	0.26	0.26
Watch	0.13	0.22	0.25	0.27	0.28	0.28	0.29	0.30
Water	0.13	0.22	0.25	0.27	0.28	0.29	0.29	0.30

Figure 5.6 depicts BVE bitmap for watermarked "Entrance" using all decomposition levels. It is obvious from the BVE bitmap that textured and high-contrast areas show less visual errors than darker smoother areas. We also observe that visual errors in the brighter area near the center of the image are the lowest compared to other image areas. This observation matches Watson's use of luminance masking, where his perceptual model accounts for the fact that brighter regions of an image can be changed by a larger amount before being noticed [32].

We also observe that textured areas of the image tend to have less perceptual errors. This confirms Mannos's observations [69] about the human eye's Contrast Sensitivity Function (CSF), which implies that human eyes are most sensitive to luminance differences at mid-range frequencies. The eye's sensitivity decreases significantly at lower and higher frequencies.

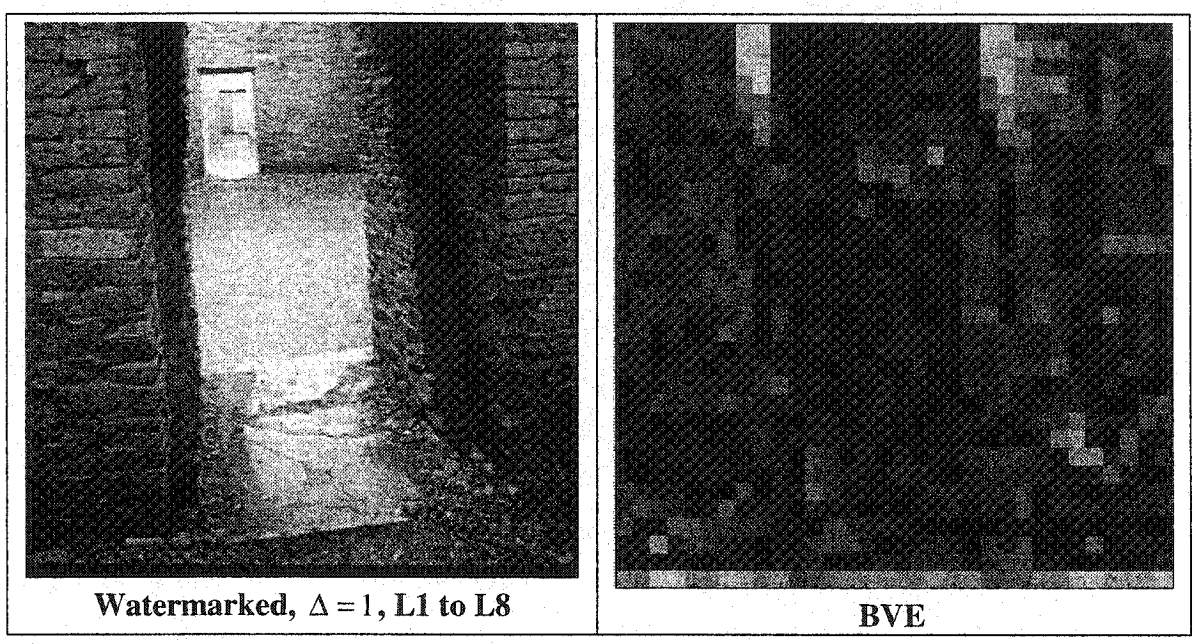


Figure 5.6 Watermarked "Entrance" image along with its block visual error bitmap.

From results of simulation set 1, we observe the following:

1- wPSNR gives a better estimate of distortion than PSNR.

2- The best visual metric among the three metrics we used, based on our simulations, was the Watson metric. The reason is its ability to give localized visual error estimates. BVE bitmaps that were generated using Watson's TPE gives the most realistic and detailed estimate on visual distortion of the host image due to watermark embedding.

3- In our simulation set, we embed watermarks using $\Delta = 1$. We embed eight different watermarks for each image, always starting at the first decomposition level ($L_{start} = 1$) and changing the maximum level where we embed the watermark bits from one to eight. From the simulation results, we can see that using all decomposition levels does not badly affect the visual quality compared to using just one or two levels. This

means we have the freedom of embedding the watermark bits at any decomposition level as long as this improves the algorithm's performance. In the next simulations, we will study this further. We will also study the effect of varying the quantization step Δ.

5.4.2 Simulation Set 2:

In this second simulation set, we watermark the test images using different values of the quantization step Δ and decomposition levels L1-L5. We study the visual effect of embedding telltale watermarks by comparing PSNR, wPSNR, and the Watson metric results. Based on theses results, we come to a conclusion on a practically suitable range for the quantization step Δ.

5.4.2.1 PSNR:

The following table shows the effect of embedding watermark bits using various values of the quantization step Δ on the PSNR (dB) of the original image compared to the watermarked one. We used the decomposition levels from one to five to embed the watermark bits.

Table 5.5 PSNR (dB) values for watermarked images using various quantization steps.

	Δ=1	2	5	10	20
Baboon	44.14	38.16	30.14	24.30	18.16
Bear	44.25	38.23	30.46	24.43	18.64
Entrance	44.13	38.18	30.22	24.35	18.53
F16	44.17	38.17	30.16	24.17	18.37
Boat	44.11	38.14	30.17	24.27	18.34
Opera	44.28	38.11	30.20	24.14	18.23
Pills	44.04	38.16	30.20	24.21	18.21
Rose	44.18	38.16	30.20	24.17	18.22
Watch	44.24	38.02	30.24	24.37	18.48
Water	44.30	38.12	30.08	24.08	18.11

The slight difference in PSNR values in that table and those in the fifth column (L1-L5) of Table 5.1 is due to the fact that every time we embed the watermark, a new random key is used to select the exact locations of the coefficients to quantize in order to embed the watermark bits. The difference of the watermark bits distribution could lead to that slight change of the PSNR values.

Figure 5.7 indicates how the quality of the image unacceptably degrades when we use any values of Δ higher than two. This restricts our algorithm to only use scale factor values of either one or two in order to be practically applicable.

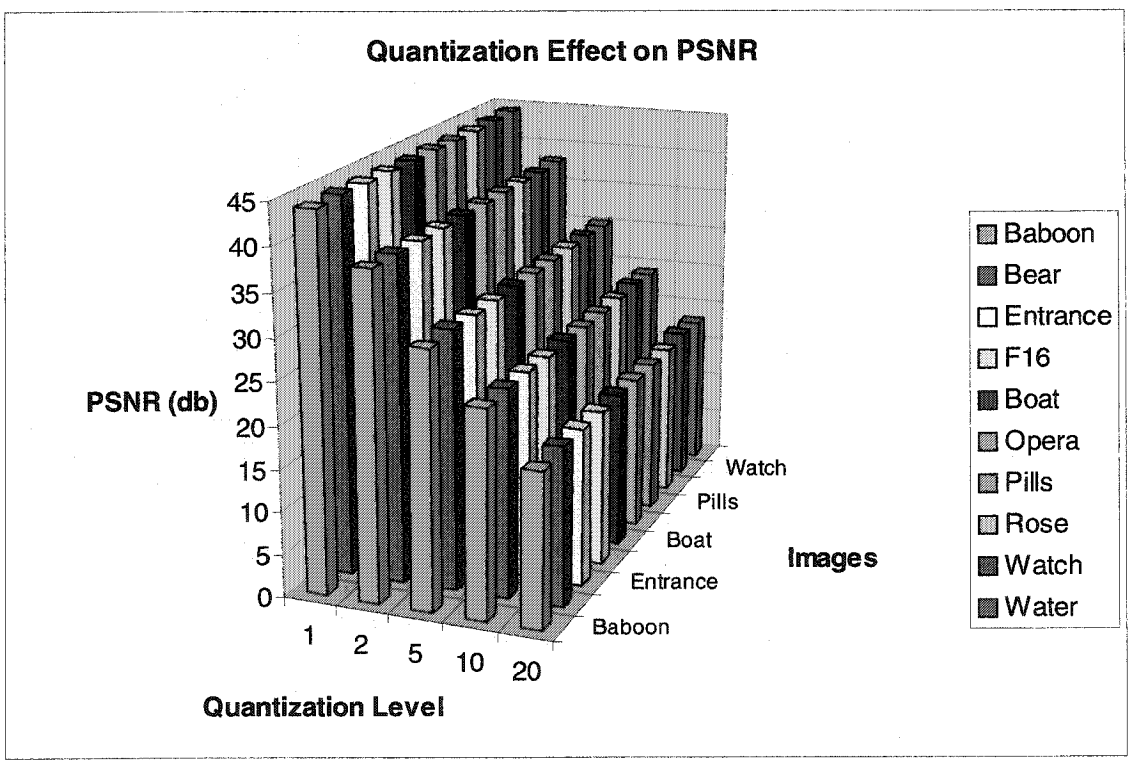


Figure 5.7 Effect of watermark embedding using various quantization steps on PSNR (dB) values of watermarked images.

5.4.2.2 wPSNR:

Table 5.6 and Figure 5.8 depict the results of the same simulation applied in order to measure effect of watermark embedding on wPSNR (dB).

Table 5.6 wPSNR (dB) values for watermarked images using various quantization steps.

	Δ=1	2	5	10	20
Baboon	48.94	43.05	34.98	29.19	23.01
Bear	45.74	39.73	31.97	25.98	20.19
Entrance	46.49	40.50	32.59	26.75	21.00
F16	45.36	39.33	31.32	25.33	19.66
Boat	46.29	40.32	32.35	26.49	20.57
Opera	46.57	40.43	32.48	26.46	20.54
Pills	45.26	39.38	31.43	25.45	19.45
Rose	44.90	38.88	30.92	24.89	18.95
Watch	45.67	39.42	31.63	25.78	19.94
Water	46.49	40.32	32.27	26.24	20.33

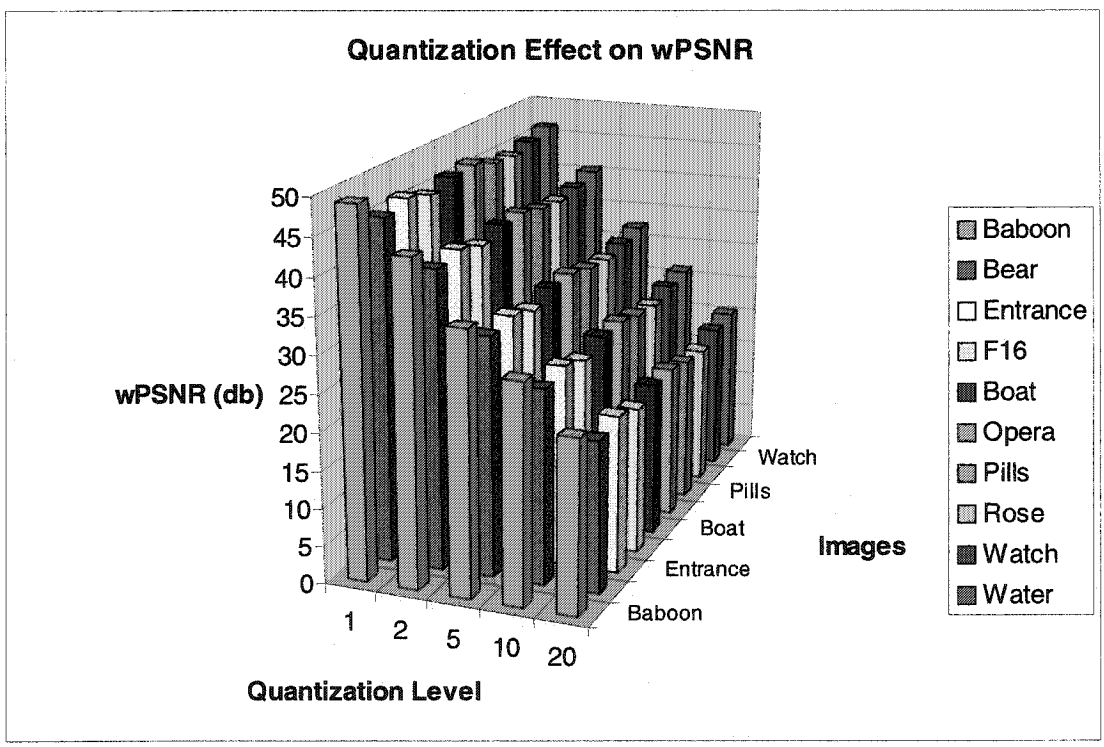
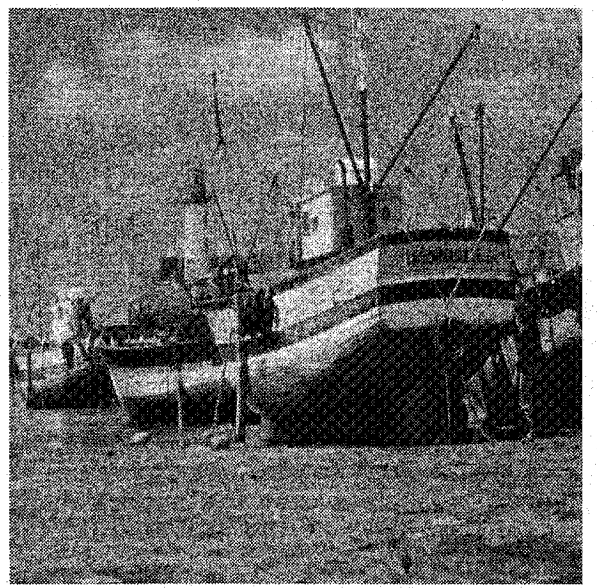


Figure 5.8 Effect of watermark embedding using various quantization steps on wPSNR (dB) values of watermarked images.

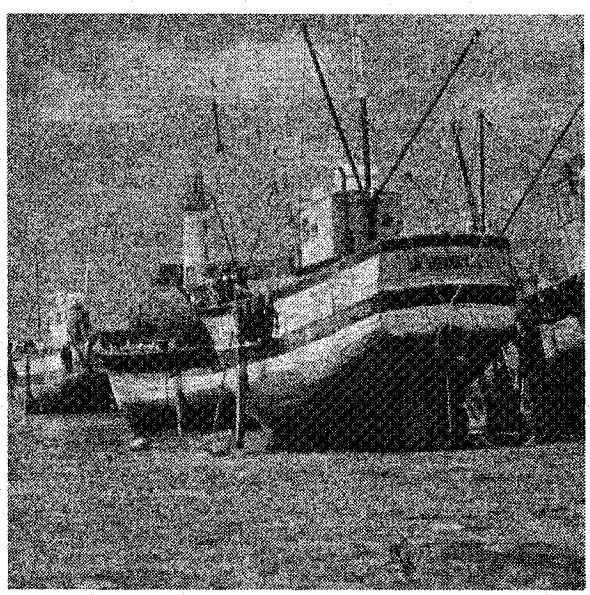
Figure 5.9 shows one of the sample images ("Boat") watermarked using various values of Δ. It is visually clear that $\Delta = 2$ would be the highest quantization level we can use without significantly degrading the host image's visual quality.



Original

$\Delta = 1$

$\Delta = 2$



$\Delta = 5$

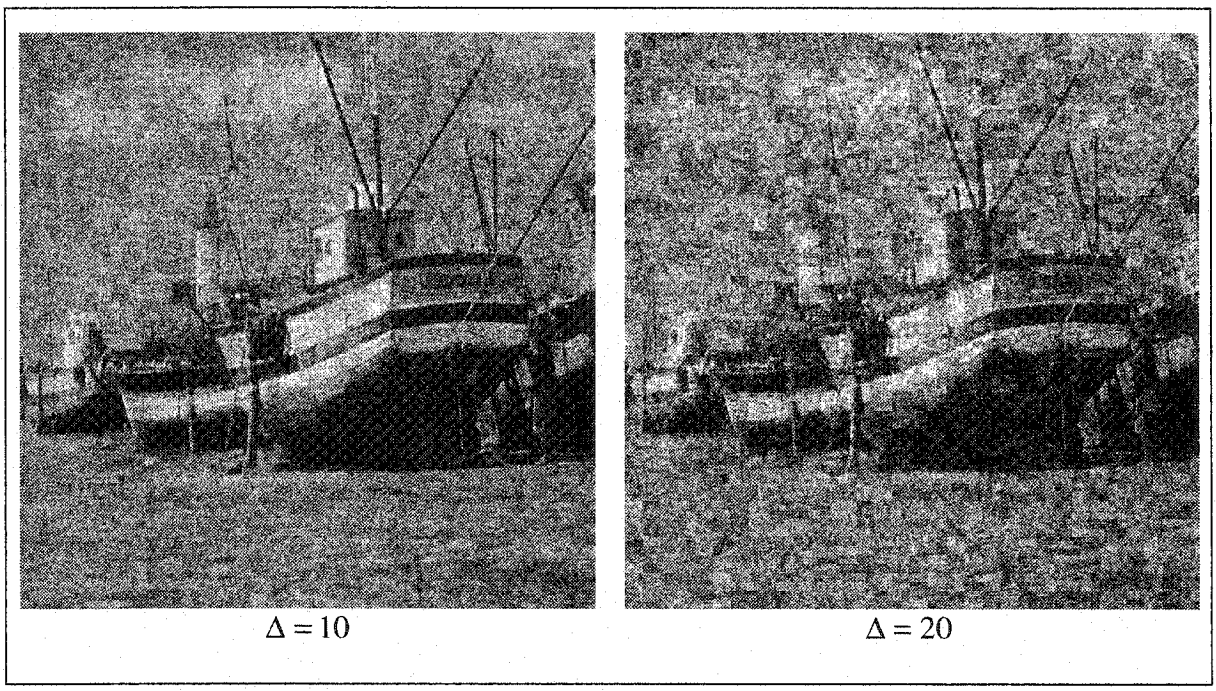


Figure 5.9 "Boat" watermarked with various quantization strengths.

In the next section, we present the Watson metric results for simulation set 2.

5.4.2.3 The Watson Metric:

Table 5.7 depicts the resulting TPE, in units of JNDs, of simulation set 2.

Table 5.7 TPE in units of JNDs of simulation set 2.

	Δ=1	2	5	10	20
Baboon	0.25	0.49	1.22	2.44	4.90
Bear	0.60	1.22	2.85	5.77	10.97
Entrance	0.26	0.53	1.33	2.59	5.02
F16	0.22	0.44	1.11	2.21	4.33
Boat	0.26	0.51	1.28	2.53	5.00
Opera	0.27	0.55	1.35	2.71	5.38
Pills	0.24	0.48	1.21	2.42	4.74
Rose	0.25	0.49	1.24	2.48	4.93
Watch	0.28	0.56	1.39	2.74	5.34
Water	0.28	0.56	1.41	2.83	5.63

Figure 5.10 below, based on the data in Table 5.7, indicates that except for "Bear" image, all test images can be watermarked using Δ value of one or two without damaging the visual quality of the images.

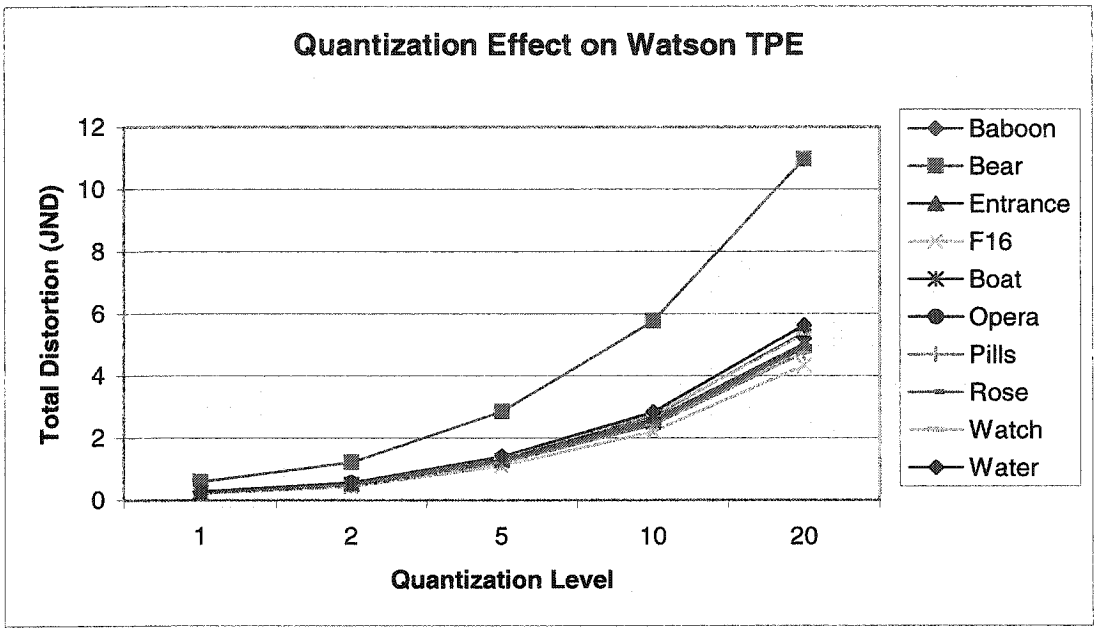


Figure 5.10 Effect of watermark quantization strength on TPE.

As mentioned in Section 5.4.1.3, the TPE error values for "Bear" image are generally higher than those of other test images due to the fact that the watermarking algorithm embeds watermark bits into the totally dark (almost black) areas of that image. This leads to highly visual distortions in these areas. Since the TPE is the mean value taken over all of the image blocks, the highly distorted dark areas account for the higher TPE for "Bear" image.

To make our argument more clear, we use Figure 5.11 that depicts mesh representation of the LPE matrix for "Bear" image that was watermarked using $\Delta = 20$. The peak areas can be shown, by looking at the "Bear" image itself, to correspond to the darker regions of the

image. All other region of the image generally have LPE less than one JND, meaning that their visual quality was not damaged.

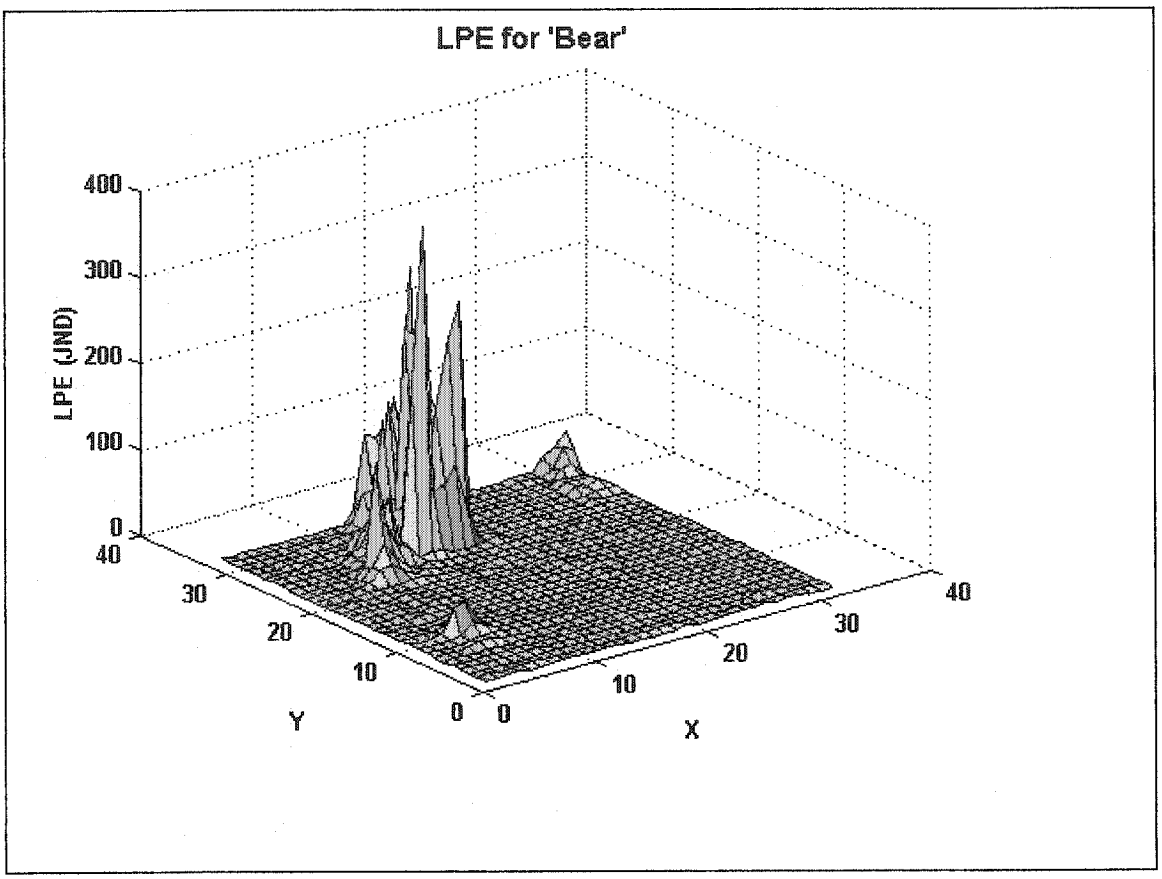


Figure 5.11 Local Perceptual Error matrix for "Bear" watermarked using $\Delta = 20$.

In Figure 5.11, we can see three different peaks: The first one which is closer to the (0,0) point corresponds to the dark area in the bear's ear. The second peak at the opposite corner of the chart corresponds to the dark area at the lower-right corner of the image. The third peak, and the largest of the three peaks, corresponds to the dark chest area of the bear. We can see in Figure 5.11 that other than these three areas that have extremely high perceptual distortion levels, all other areas of the image have distortion levels much less than the TPE=10.97 (JND) reported in Table 5.7.

Figure 5.11 gives a hint about one weakness of non-adaptive watermarking algorithms. If a watermarking algorithm can use the Watson metric, or a similar perceptual metric, to avoid embedded high watermark energy in sensitive image areas, it would be much easier to use such an algorithm to watermark a broader set of images.

To show that this weakness is not only associated with images that have large contiguous dark areas, we show a similar graph of LPE matrix for another image that does only have some small dark areas. It is obvious that the LPE values for these rather small areas are still much higher than the mean TPE value. In Figure 5.12 below, we use the watermarked "Boat" image with $\Delta = 2$. For this value of Δ, Table 5.7 shows a TPE value of 0.51 (JND). The graph shows that most of the image block are only distorted by a fraction of a JND. It also shows that a rather small dark area (at the right edge of the image) has a peak LPE of about 1.2 (JND), which is approximately three times the mean perceptual error value reported by TPE.

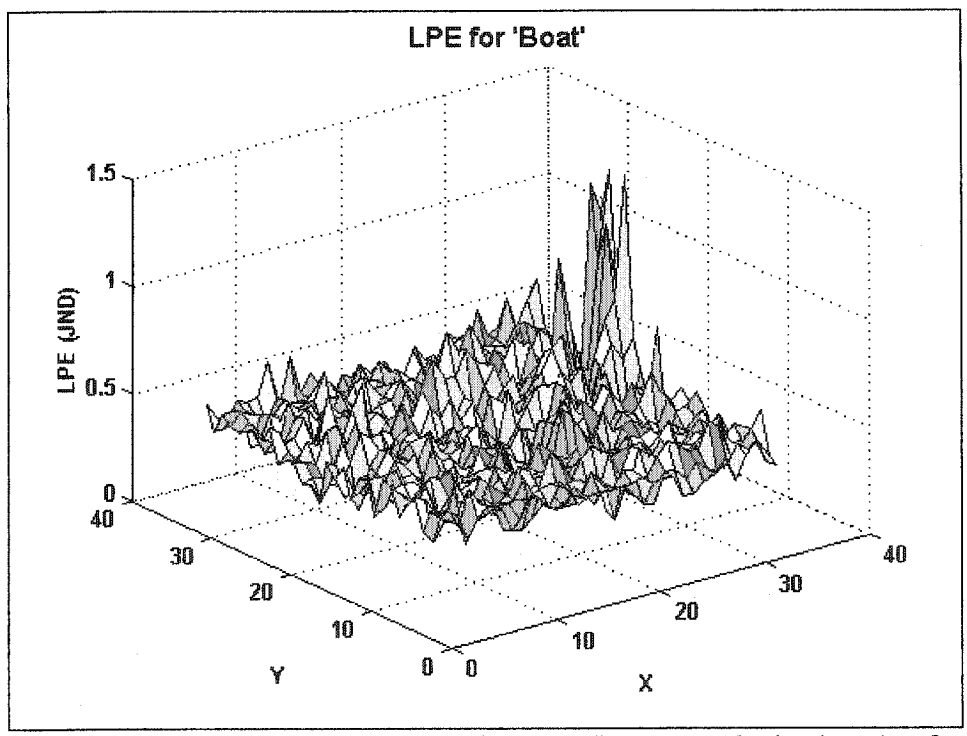


Figure 5.12 Local Perceptual Error for "Boat" watermarked using $\Delta = 2$.

5.4.3 Simulation Set 3:

This simulation set is rather similar to simulation set 1. We always embed the watermark using $\Delta = 1$. Instead of always starting at decomposition level $L_{start} = 1$ and incrementally increasing L_{end}, we take a different approach this time. We fix the number of decomposition levels we use to embed the watermark bits. In that specific simulation, we used a fixed window size, $L_{end} - L_{start} + 1$, of 4. So, the first iteration of our simulation set will embed a watermark with $\Delta = 1$ into levels from one and up to level four into all test images. The second iteration will embed a watermark with $\Delta = 1$ into levels from two and up to level five into all test images. This goes on until we approach the last iteration, where we embed a watermark with $\Delta = 1$ into levels from five and up to level eight into all test images. In other words, we use a sliding window that has a fixed size and that moves across the image's spatial frequency spectrum starting from higher frequencies and ending at lowest ones.

We decided to carry out this simulation in order to study the relative significance of choosing to embed the watermark bits within a specific frequency range of the image. Levels one to four represent high spatial frequencies, while levels five to eight represent low spatial frequencies of the image. It is known [11] that modifications in lower frequency channels of an image generally introduce more visible artifacts. On the other hand, lower frequency components of an image represent most of the image's energy, so when we use these frequencies to embed the watermark data, we get more reassurance that the watermark will be able to resist a lot of distortions. If distortions were able to render a low-frequency-embedded watermark undetectable, this usually means that the visual quality of the image was also badly damaged.

Embedding a watermark into the higher frequency spectrum is equivalent to adding Gaussian noise to the image, assuming the watermark data has the same statistical properties as Gaussian noise. Making sure that the embedded watermark data are random is very important as it makes analyzing and/or removing the watermark by an adversary much more difficult. Since Gaussian noise can usually be alleviated using a smoothing filter, this means that our watermark will also be so weak against filtering.

To summarize, a watermark that is embedded into higher frequency bands will introduce less visible distortions but will be very fragile. A low frequency range watermark will potentially be much more robust, but also has more potential for creating visual artifacts in the image. This motivates this simulation where we try to tune the telltale watermark parameters so that we can make a suitable trade-off between visual quality of the watermarked image and robustness of the watermark.

It should be noted that moving our sliding window towards the lower frequency range also means embedding less watermark bits. This is because of the fact that upper decompression levels, corresponding to lower frequencies, have fewer coefficients. To get a more subjective estimate on the effect of the sliding window on an image's quality, we would have to normalize the resulting metric results (PSNR, wPSNR, or Watson) so that we can get a measure of the distortion incurred per one embedded watermark bit. This normalization would not help much though in studying the telltale algorithm because that specific algorithm would still have to use levels of compositions that have different number of coefficients. So, it becomes more sensible to get per-level rather than per-bit results.

This simulation set generates the same set of results generated from the two previous simulation sets, i.e., PSNR, wPSNR, and Watson's LPE and TPE. It should be noted that from Simulation set 1, we are confident that using even all decomposition levels to embed the watermark bits does not severely damage the visual quality of any of the test images. We perform this test to get solid numbers that should help on weighing relative visual quality damage versus robustness of the watermark.

5.4.3.1 PSNR:

Table 5.8 shows the effect of embedding watermark bits into different frequency ranges on the PSNR of the original image compared to the watermarked one.

Table 5.8 PSNR (dB) values for watermarked images using decomposition levels sliding window.

	L1-L4	L2-L5	L3-L6	L4-L7	L5-L8
Baboon	45.17	45.12	44.81	45.27	45.38
Bear	45.20	45.20	45.50	45.20	43.91
Entrance	45.20	45.21	45.26	45.04	45.18
F16	45.15	45.12	45.13	44.97	44.34
Boat	45.17	45.18	45.25	45.39	45.54
Opera	45.16	45.07	44.97	45.09	45.73
Pills	45.16	45.07	45.06	44.92	43.93
Rose	45.14	45.16	45.11	44.94	44.63
Watch	45.14	45.18	45.29	45.17	45.24
Water	45.19	45.14	44.89	44.82	45.29

Figure 5.13 shows that most of PSNR values lie within a narrow range. Some images ("Bear", "F16", "Opera", and "Rose") tend to have their PSNR values dropping with the sliding window moving towards lower frequencies. Other test images tend to have a uniform PSNR pattern. As we can see from the above table, All PSNR values are in the range from 43.91 to 45.73 (dB). This range is deemed an acceptable one.

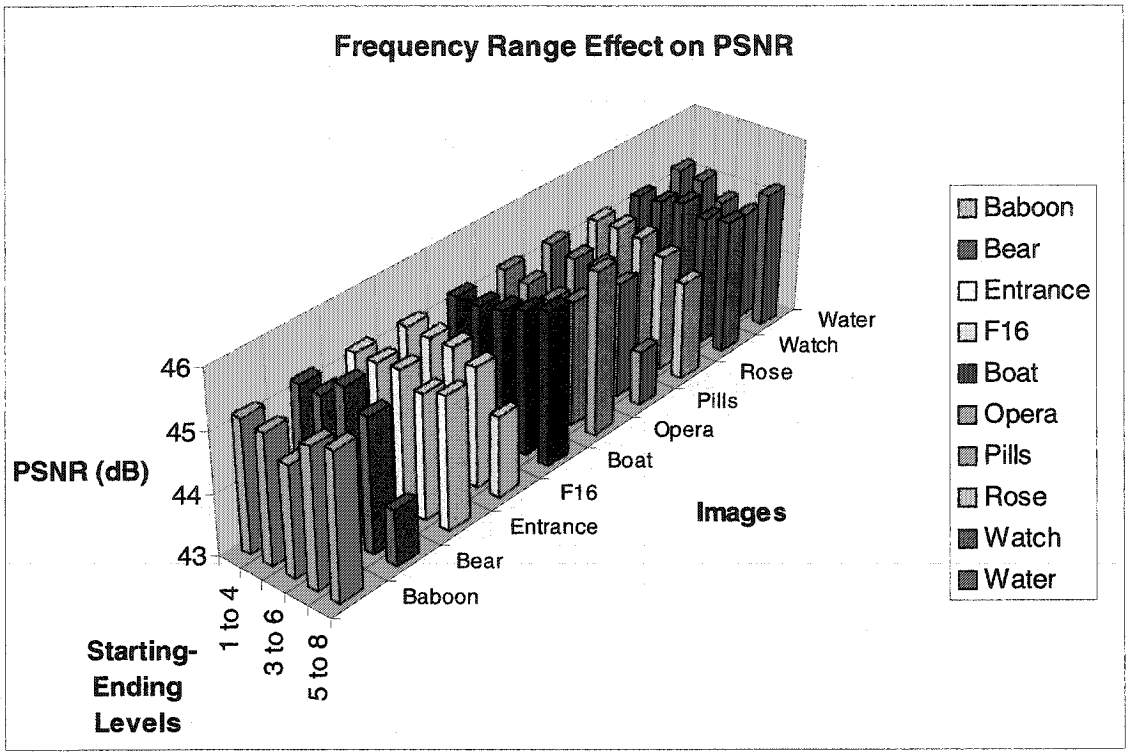


Figure 5.13 Effect of moving the sliding window across the wavelet decomposition levels on PSNR (dB).

5.4.3.2 wPSNR:

Table 5.9 and depict similar results for wPSNR. From the previous two simulations, we can predict that the wPSNR charts will be more smooth, slightly higher as wPSNR accounts for the effects of textures and shadows of the image on the final estimated value.

Table 5.9 wPSNR (dB) values for watermarked images using sliding window.

	L1-L4	L2-L5	L3-L6	L4-L7	L5-L8
Baboon	50.04	49.94	49.62	50.07	50.46
Bear	46.69	46.69	46.99	46.70	45.46
Entrance	47.56	47.55	47.57	47.43	47.63
F16	46.37	46.33	46.40	46.20	45.58
Boat	47.38	47.36	47.43	47.56	47.68
Opera	47.48	47.37	47.21	47.42	48.09
Pills	46.41	46.29	46.27	46.11	45.13
Rose	45.87	45.89	45.86	45.64	45.37
Watch	46.56	46.55	46.70	46.54	46.66
Water	47.39	47.34	47.09	47.08	47.44

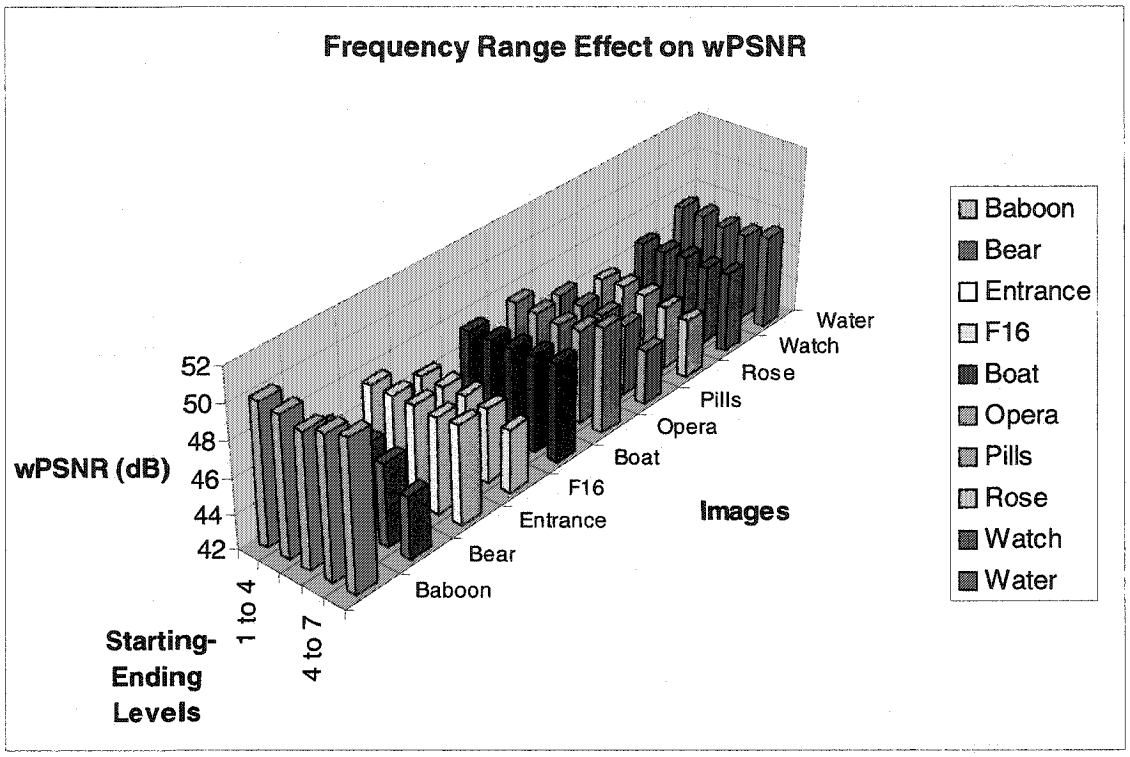


Figure 5.14 wPSNR (dB) values for watermarked images using sliding window.

Comparing PSNR and wPSNR values for this simulation, we notice how wPSNR charts are more smooth and uniform throughout almost all test images. One can attribute that to the fact that wPSNR metric is an improvement to PSNR.

Accounting for the effect of textures on the human visual system removes any irregular patterns of the wPSNR curves. We also note that the mean PSNR value for the above test is 45.09 (dB) while the mean wPSNR value is 47.09 (dB) which is higher for the same reason we just mentioned. Assume that a pixel in the watermarked image is different from its corresponding pixel in the original one by a value X, PSNR metric directly uses X to calculate the final PSNR value. wPSNR, on the other hand, scales the difference value of X down if the neighboring pixels have a structure that would help hide that

difference from a human observer's eye. So, wPSNR values tend to be higher than PSNR values for the same images.

5.4.3.3 The Watson Metric:

Table 5.10 depicts the resulting TPE, in units of JNDs, of simulation set 3. All errors are fairly below the one JND threshold.

Table 5.10 TPE in units of JNDs of simulation set 3.

	L1-L4	L2-L5	L3-L6	L4-L7	L5-L8
Baboon	0.23	0.18	0.08	0.04	0.04
Bear	0.59	0.45	0.23	0.14	0.19
Entrance	0.25	0.19	0.08	0.05	0.05
F16	0.21	0.16	0.06	0.04	0.04
Boat	0.25	0.18	0.07	0.04	0.04
Opera	0.26	0.19	0.08	0.05	0.04
Pills	0.23	0.17	0.07	0.04	0.05
Rose	0.24	0.18	0.07	0.04	0.04
Watch	0.27	0.20	0.08	0.05	0.05
Water	0.27	0.20	0.09	0.05	0.05

Figure 5.15 indicates the fact that the overall perceptual error for watermarked images drops as we embed more watermark bits towards the lower frequency range. This confirms our assumption that using lower frequency ranges (embedding watermark bits into higher decomposition levels) means that we have more potential for creating visual artifacts, but at the same time it also means we are adding less watermark energy. The overall effect of these two contradicting factors manifests itself as an overall drop of the perceptual error as we slide the window towards the higher level, lower frequency, and lesser coefficients decomposition levels.

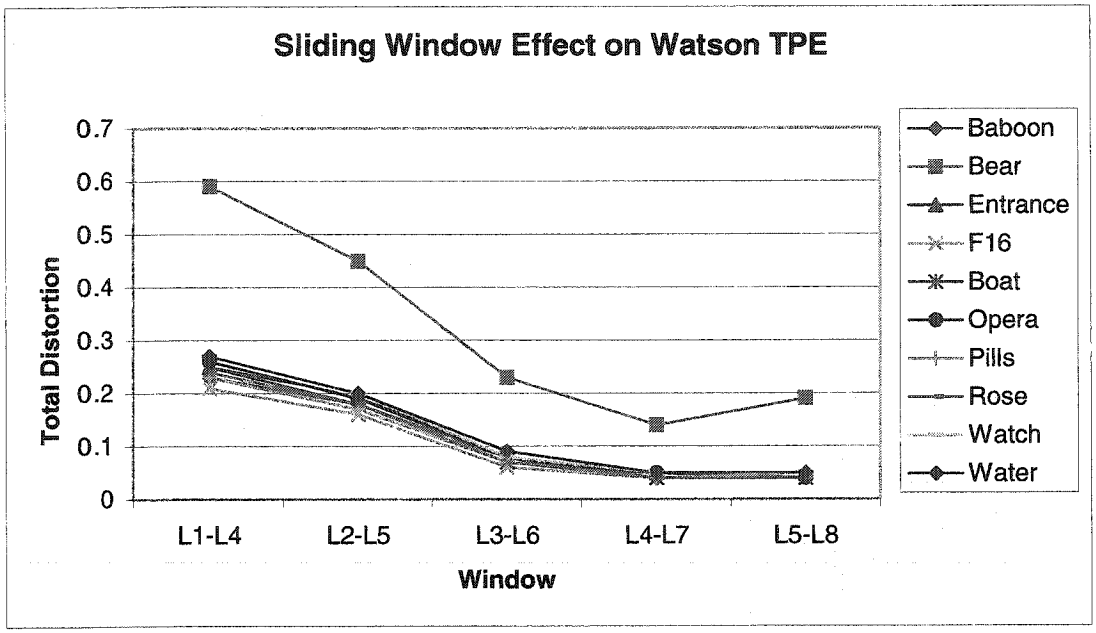


Figure 5.15 Effect of watermark frequency range on Total Perceptual Error.

5.4.4 Simulation Set 4:

The purpose of this test set is to study the robustness of the telltale algorithm against JPEG compression. We embed a watermark in each of the ten test images. We use $\Delta = 2$ and decomposition levels 2 to 5 to embed the watermark bits for all tests in that simulation set.

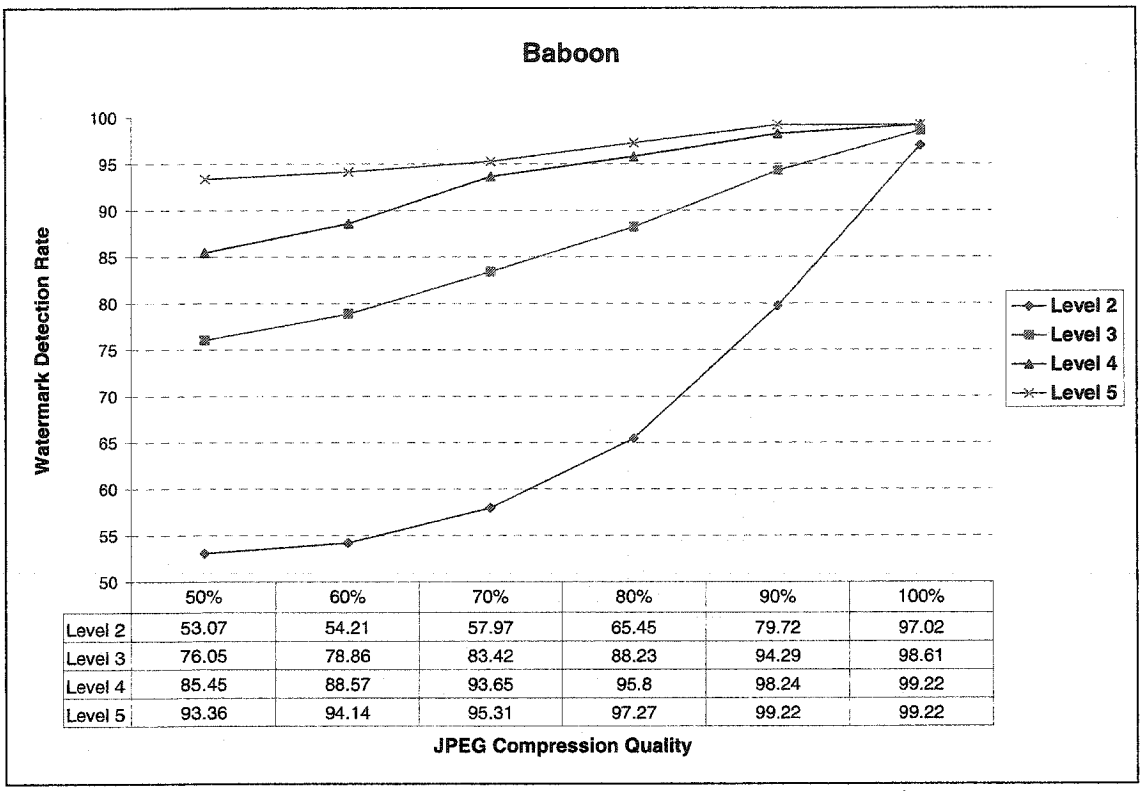
For each watermarked image, we apply JPEG compression with six different levels of quality, ranging from 50% (worst quality) to 100% (no compression, best quality) with a step size of 10%. We do not apply more JPEG compression as the visual quality of images degraded to unacceptable levels when higher JPEG compression was used. We chose $\Delta = 2$ as a comfortable watermarking scale factor based on results from previous simulation sets. We also chose decomposition levels from 2 to 5 as they have good frequency spectrum coverage. Decomposition level 1 was not used as it requires more watermark bits and thus it is more computationally expensive. A second reason not to use level 1 is that each coefficient in that level covers a spatial area of size 2×2 pixels of the image. This granularity is not critical for tamper detection as no effective tampering can be applied to such small areas. As JPEG compression possesses properties similar to that of low-pass filtering, we should expect the extraction error rate to be highest for high frequency levels and then it should improve as we move towards lower frequency decomposition levels. That is a third reason to justify our decision not to use level 1 of decomposition. We generally recommend always using decomposition level 2 as the starting level for telltale watermarking.

For each original image of the ten test images, we generated six different watermarked and JPEG compressed versions. We then fed these six versions to the watermark

extractor which reports the percentage of watermark bits that were correctly extracted in each decomposition level. If 50% or less of the extracted watermark bits were in error, the watermark is deemed undetectable. In practice, we suggest that at least 75% of the watermark bits be detected correctly to accept the watermarked image as a tamper-free one.

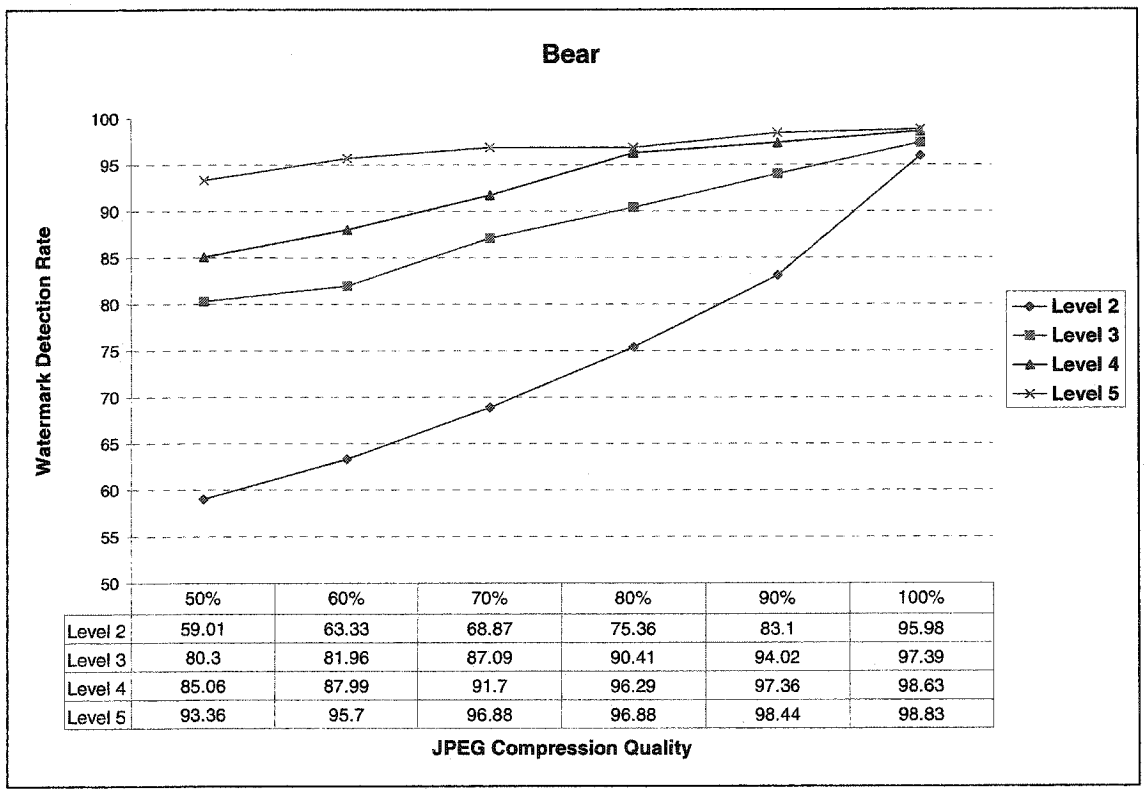
Another important issue is to make sure that the watermark bits that were in error are uniformly distributed throughout the image. This gives more evidence that no tampering of a specific spatial area of the image took place. Although the watermark bits are embedded into wavelet domain coefficients, it is one of the strengths of using the wavelet transform that we can still use information extracted from these coefficients that lie in the frequency domain and deduce conclusions about the tampering that might have taken place in the spatial domain. To verify the distribution of error bits within the watermark bits, one can either use statistical analysis or visual inspection using techniques similar to the one we suggested in Section 4.4.3.

We finally inspect the error rates for all levels to draw a conclusion regarding the watermark resistance to regular JPEG compression. The charts of Figure 5.16 depict results of our simulations for the ten test images used.



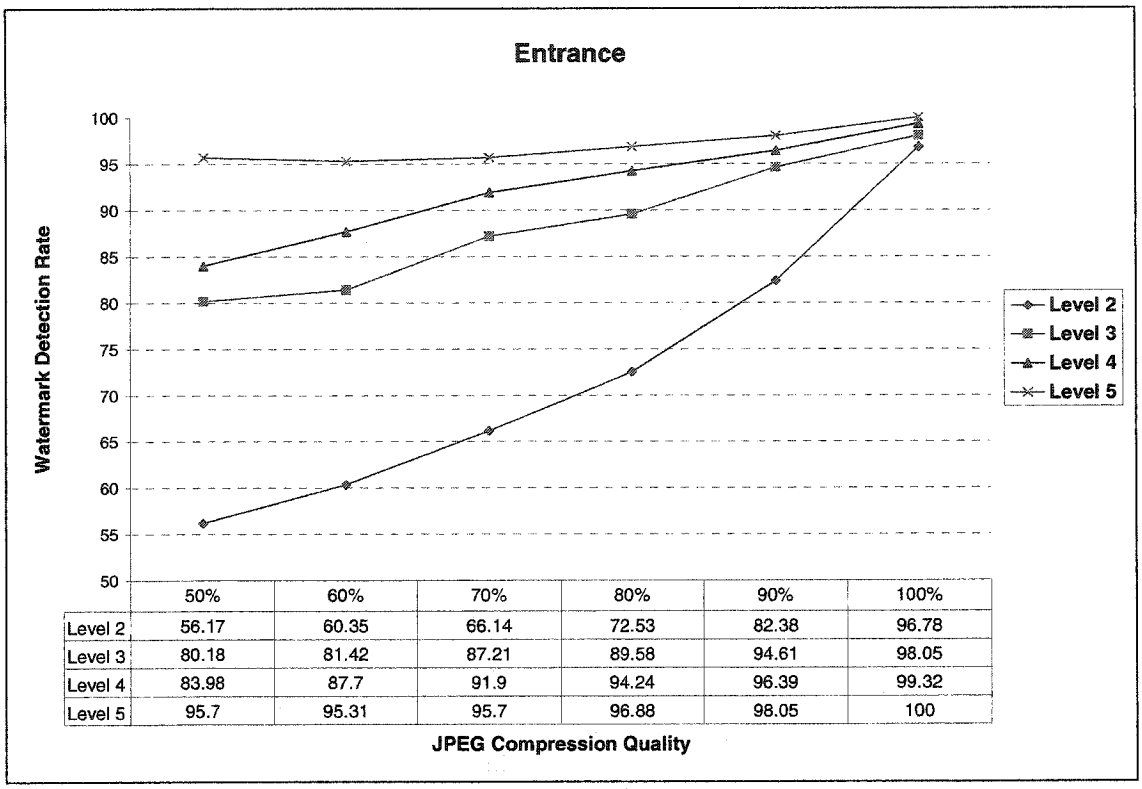
	50%	60%	70%	80%	90%	100%
Level 2	53.07	54.21	57.97	65.45	79.72	97.02
Level 3	76.05	78.86	83.42	88.23	94.29	98.61
Level 4	85.45	88.57	93.65	95.8	98.24	99.22
Level 5	93.36	94.14	95.31	97.27	99.22	99.22

Figure 5.16 Watermark resistance to JPEG compression.
(a) "Baboon" image.



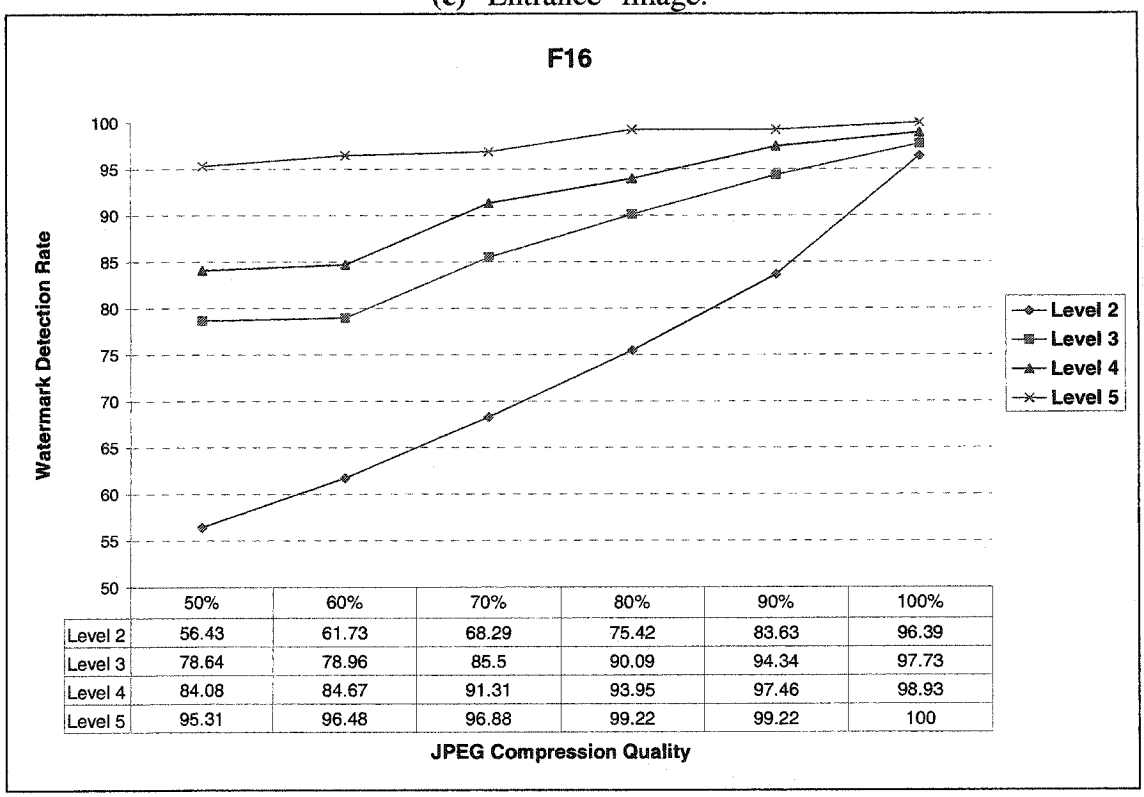
	50%	60%	70%	80%	90%	100%
Level 2	59.01	63.33	68.87	75.36	83.1	95.98
Level 3	80.3	81.96	87.09	90.41	94.02	97.39
Level 4	85.06	87.99	91.7	96.29	97.36	98.63
Level 5	93.36	95.7	96.88	96.88	98.44	98.83

(b) "Bear" image.



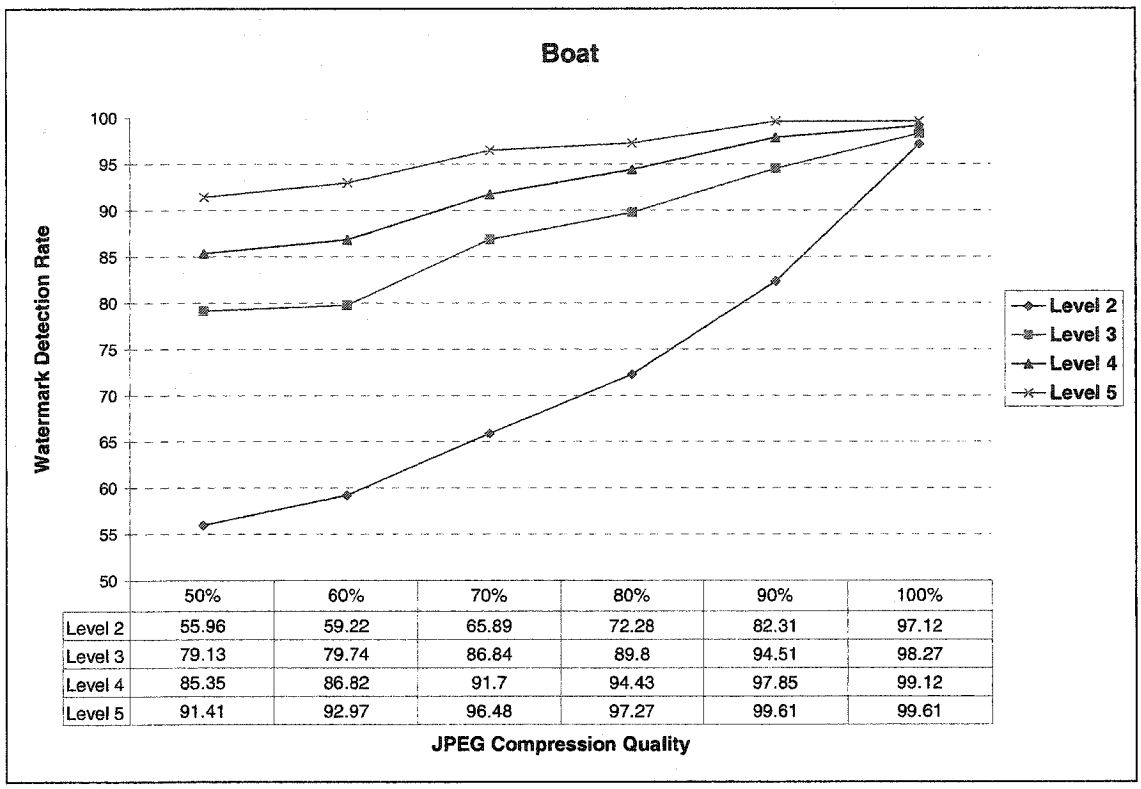
	50%	60%	70%	80%	90%	100%
Level 2	56.17	60.35	66.14	72.53	82.38	96.78
Level 3	80.18	81.42	87.21	89.58	94.61	98.05
Level 4	83.98	87.7	91.9	94.24	96.39	99.32
Level 5	95.7	95.31	95.7	96.88	98.05	100

(c) "Entrance" image.



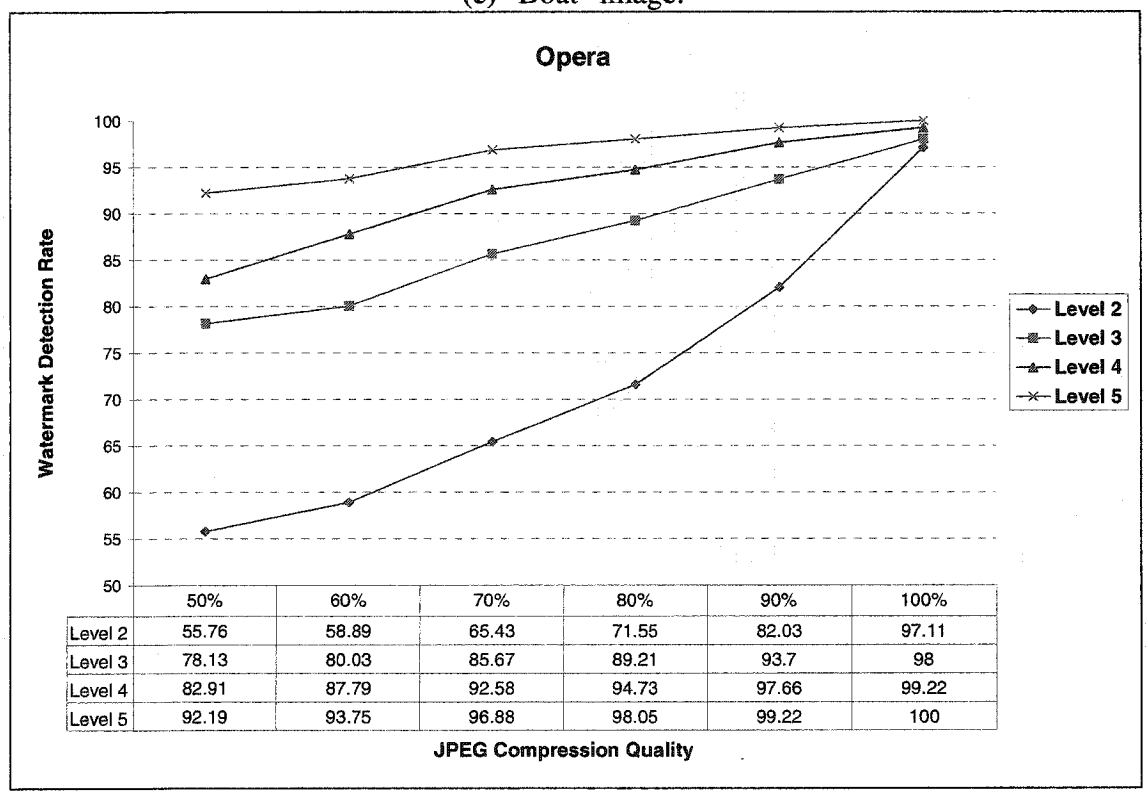
	50%	60%	70%	80%	90%	100%
Level 2	56.43	61.73	68.29	75.42	83.63	96.39
Level 3	78.64	78.96	85.5	90.09	94.34	97.73
Level 4	84.08	84.67	91.31	93.95	97.46	98.93
Level 5	95.31	96.48	96.88	99.22	99.22	100

(d) "F16" image.



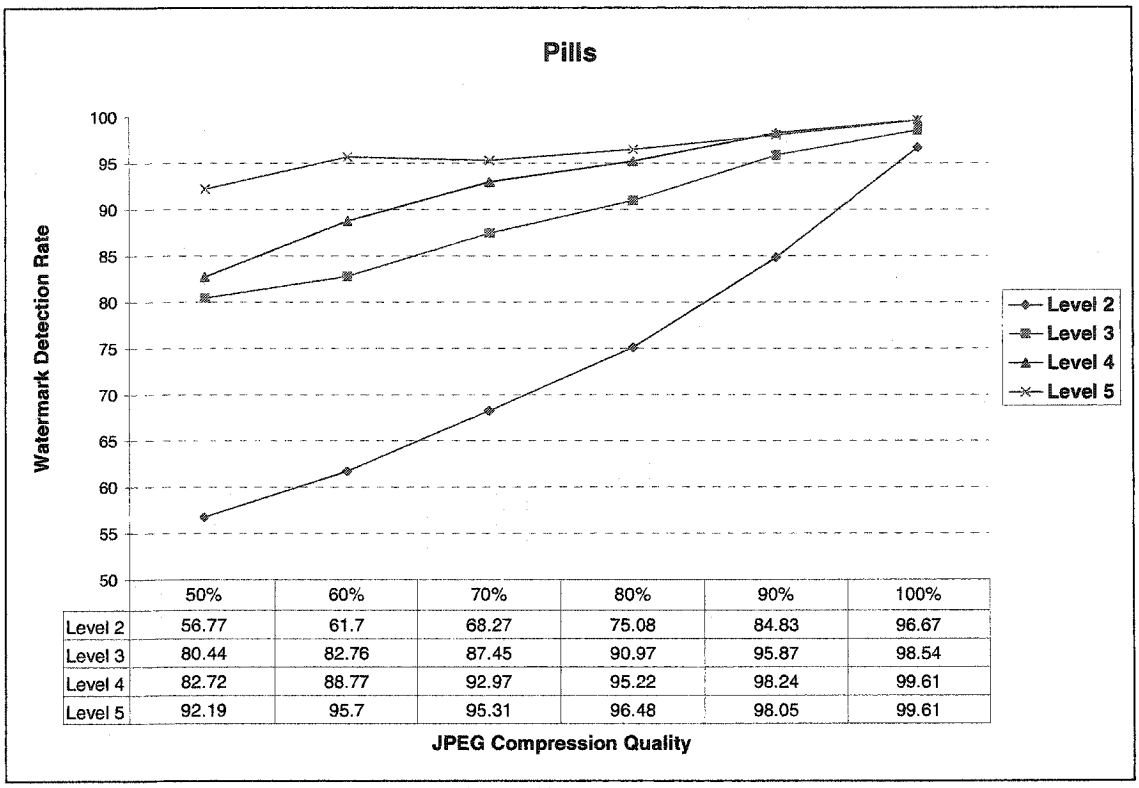
	50%	60%	70%	80%	90%	100%
Level 2	55.96	59.22	65.89	72.28	82.31	97.12
Level 3	79.13	79.74	86.84	89.8	94.51	98.27
Level 4	85.35	86.82	91.7	94.43	97.85	99.12
Level 5	91.41	92.97	96.48	97.27	99.61	99.61

(e) "Boat" image.



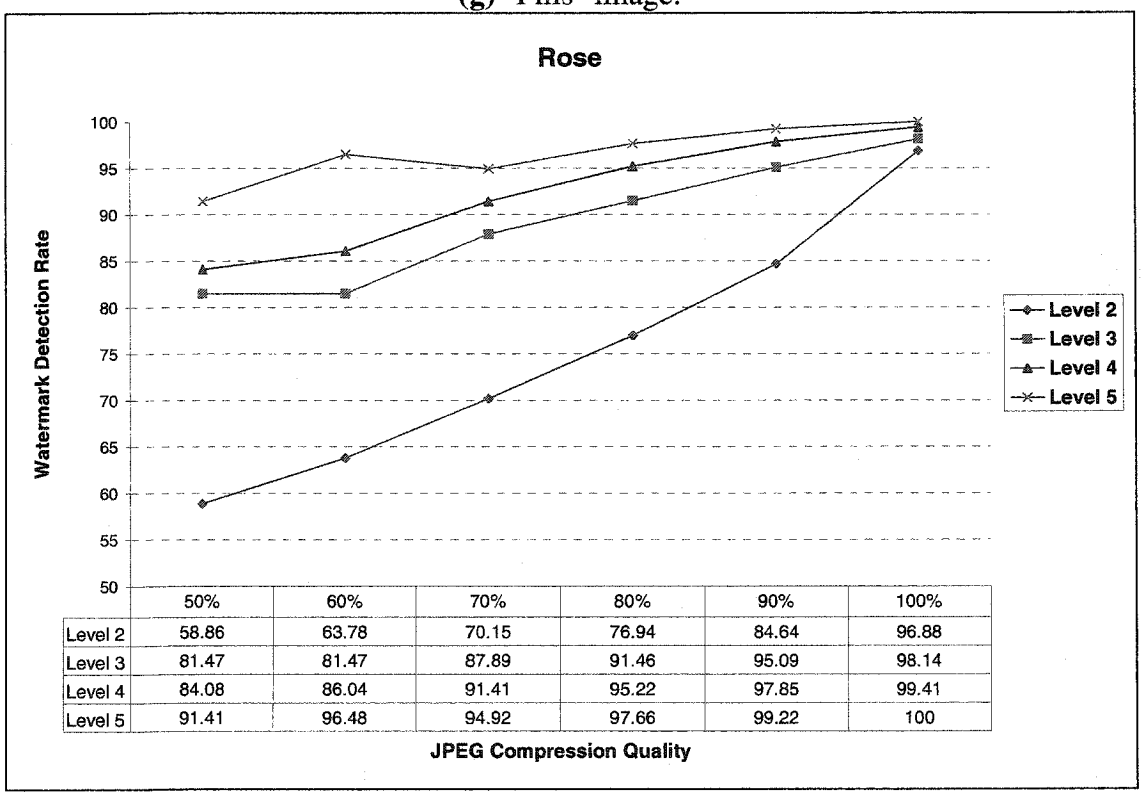
	50%	60%	70%	80%	90%	100%
Level 2	55.76	58.89	65.43	71.55	82.03	97.11
Level 3	78.13	80.03	85.67	89.21	93.7	98
Level 4	82.91	87.79	92.58	94.73	97.66	99.22
Level 5	92.19	93.75	96.88	98.05	99.22	100

(f) "Opera" image.



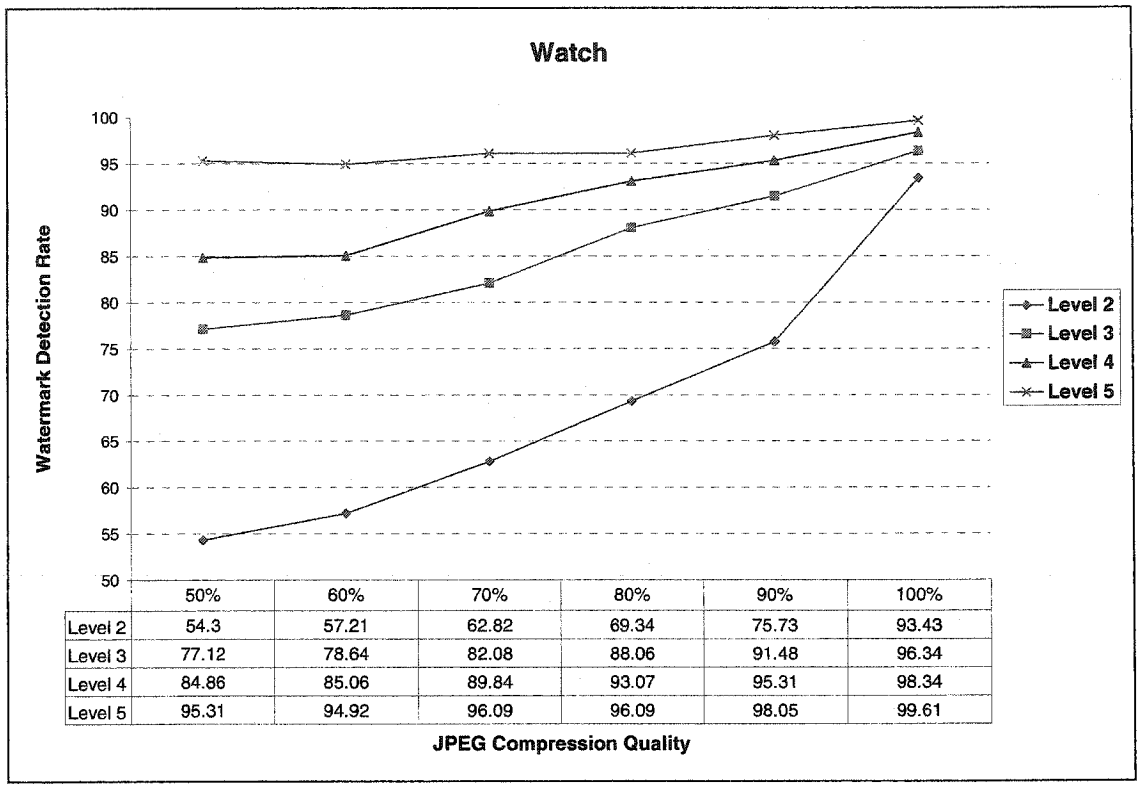
	50%	60%	70%	80%	90%	100%
Level 2	56.77	61.7	68.27	75.08	84.83	96.67
Level 3	80.44	82.76	87.45	90.97	95.87	98.54
Level 4	82.72	88.77	92.97	95.22	98.24	99.61
Level 5	92.19	95.7	95.31	96.48	98.05	99.61

(g) "Pills" image.



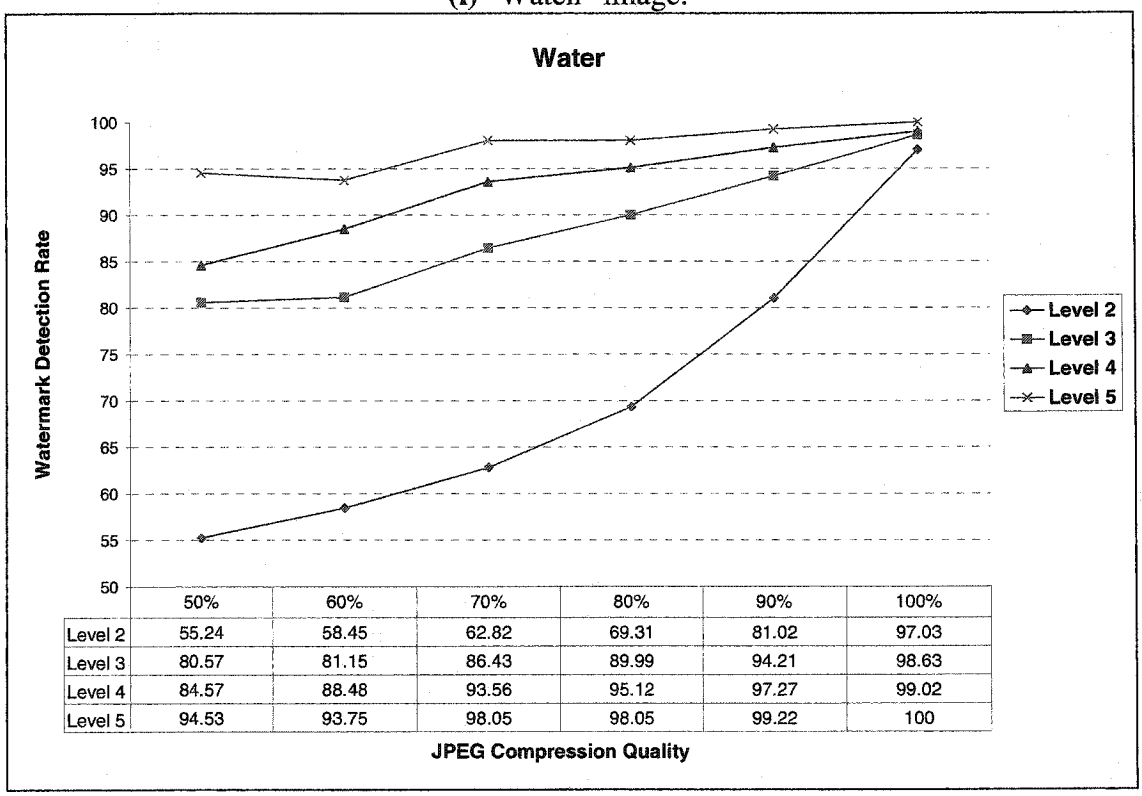
	50%	60%	70%	80%	90%	100%
Level 2	58.86	63.78	70.15	76.94	84.64	96.88
Level 3	81.47	81.47	87.89	91.46	95.09	98.14
Level 4	84.08	86.04	91.41	95.22	97.85	99.41
Level 5	91.41	96.48	94.92	97.66	99.22	100

(h) "Rose" image.



	50%	60%	70%	80%	90%	100%
Level 2	54.3	57.21	62.82	69.34	75.73	93.43
Level 3	77.12	78.64	82.08	88.06	91.48	96.34
Level 4	84.86	85.06	89.84	93.07	95.31	98.34
Level 5	95.31	94.92	96.09	96.09	98.05	99.61

(i) "Watch" image.



	50%	60%	70%	80%	90%	100%
Level 2	55.24	58.45	62.82	69.31	81.02	97.03
Level 3	80.57	81.15	86.43	89.99	94.21	98.63
Level 4	84.57	88.48	93.56	95.12	97.27	99.02
Level 5	94.53	93.75	98.05	98.05	99.22	100

(j) "Water" image.

From the results above, we conclude that watermark bits embedded into coefficients of higher frequency levels tend to have higher extraction error rates. We also observe that decomposition level 2 tends to have unacceptable error rates of approximately 45% for all test images. This only happens for the worst quality JPEG compression of 50%. For other JPEG quality ratios, watermark extraction rates are generally higher than 75% which is deemed acceptable and should not generally trigger any tampering alarms.

Extraction errors that appear when 100% JPEG quality is used are due to the fact that our simulations actually make two-way conversions of the images from the ".BMP" format to the ".JPG" format and back to ".BMP" format in order to perform JPEG compression. We consider this type of error is equivalent to random mild distortions that a digital image can undergo through its lifetime. We observe that the lowest frequency decomposition level coefficients (level 5) was the most resistant to this type of distortion.

In general, our simulations show telltale algorithm demonstrates relatively good resistance against distortions imposed by mild to average JPEG compression.

5.4.5 Simulation Set 5:

The purpose of this simulation set is to test the watermark behaviour when the watermarked image is slightly rotated. Using $\Delta = 2$ and decomposition levels 2 to 5, we watermarked our test images following that with a rotation of each image by one degree clockwise. Gaps created by rotation were filled by black pixels.

As shown in Table 5.11 and

Figure 5.17, the watermark is undetectable at nearly all decomposition levels. The wavelet transform is sensitive to rotation. Watermarks designed to be robust against rotation might use the Fourier-Mellin (log polar) transform which is invariant to rotation.

In general, we should expect that any loss of synchronization will render the watermark undetectable. That includes RST operations.

Table 5.11 Watermark detection rate for images rotated by 1°.

	Level 2	**Level 3**	**Level 4**	**Level 5**
Baboon	49.81	51.51	51.76	51.17
Bear	50.05	49.95	54.39	46.48
Entrance	50.73	49.93	52.54	57.81
F16	49.69	50.02	55.66	66.41
Boat	49.78	50.42	52.25	54.30
Opera	51.10	50.85	49.90	55.86
Pills	50.23	50.78	50.98	51.17
Rose	50.26	51.34	54.49	54.69
Watch	50.16	49.46	51.95	59.76
Water	50.81	52.17	53.03	51.17

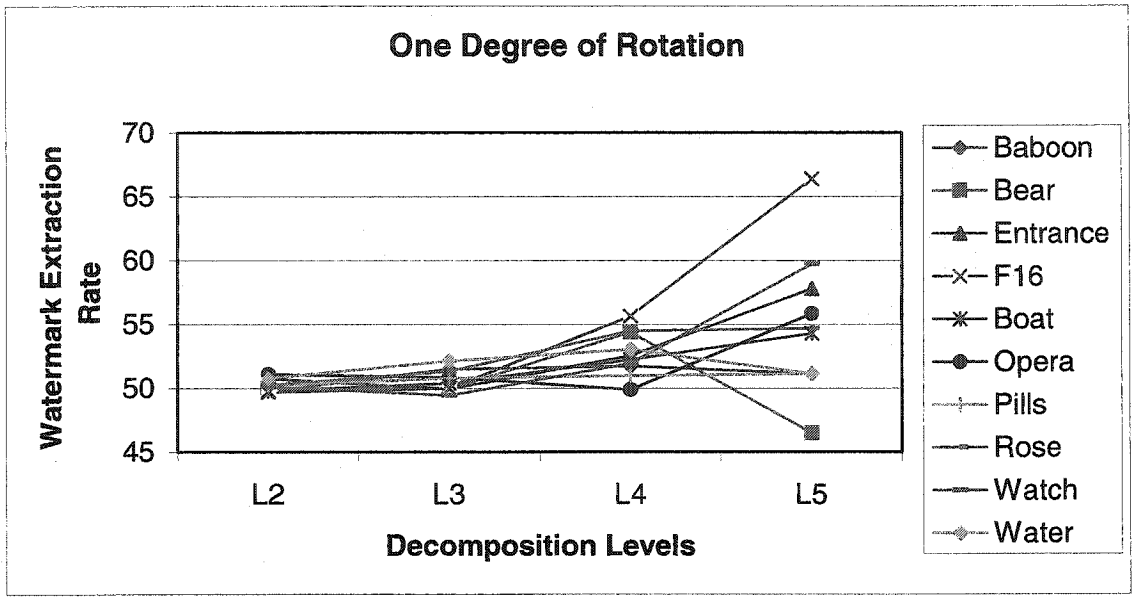


Figure 5.17 Watermark resistance to 1° rotation.

5.4.6 Simulation Set 6:

In this simulation, we add noise to the ten watermarked images, then we attempt to extract the watermarks from the noisy images. We use $\Delta = 2$ and decomposition levels 2 to 5 as parameters of the embeddor. The Gaussian noise applied to the image has zero mean ($\mu = 0$) and a variance of 0.001 ($\sigma^2 = 0.001$). This generates PSNR values in the vicinity of 30 (dB) for all test images.

Table 5.12 and Figure 5.18 show that the damage caused by Gaussian noise has the same pattern for all test images. Watermark bits embedded into lower frequency coefficients were able to survive the distortions caused by Gaussian noise with detection rate of decomposition level 5 approaching 90% for all test images.

Table 5.12 Watermark detection rate for noisy watermarked images.

	Level 2	**Level 3**	**Level 4**	**Level 5**
Baboon	49.92	60.99	79.88	89.06
Bear	50.71	63.09	79.98	87.50
Entrance	50.57	61.62	79.39	89.84
F16	49.89	62.21	78.52	88.67
Boat	49.89	62.69	80.37	89.84
Opera	50.60	62.04	78.61	89.84
Pills	50.09	60.74	81.15	91.02
Rose	50.10	61.16	78.22	89.06
Watch	50.26	59.28	78.61	87.11
Water	50.11	60.72	80.76	90.23

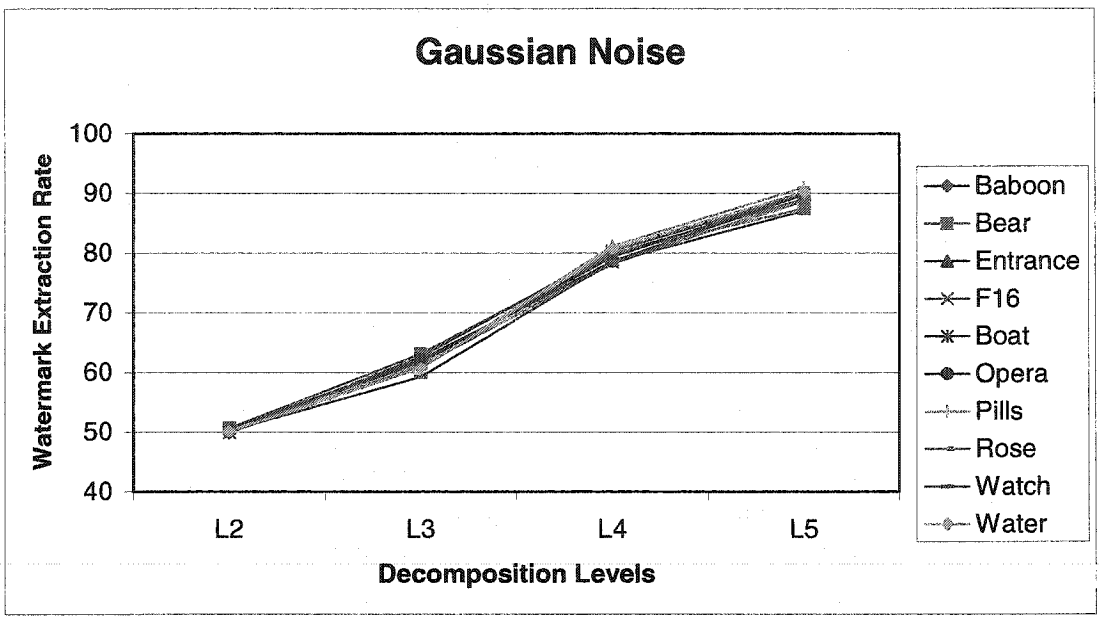


Figure 5.18 Watermark resistance to Gaussian noise distortions.

Table 5.13 PSNR values for noisy watermarked images.

Baboon	Bear	Bear	F16	Boat	Opera	Pills	Rose	Watch	Water
30.03	30.26	30.09	30.05	30.04	30.06	30.03	30.04	30.05	30.04

The watermark was able to withstand the zero-mean Gaussian noise that caused the noisy images to have PSNR values of around 30 (dB) as shown in Table 5.13. Starting from level 4, most of the watermark bits were successfully detected.

We also observe that there is not much of variation in detection rate among various test images. Lower frequency decomposition levels had performed well as the average energy added by the noise signal for larger image blocks is close to zero.

5.5 Summary

Six simulation sets were performed. The goal was to evaluate the telltale watermarking algorithm. Various simulations to measure the perceptual impact of adding the telltale watermark to test images indicated that we can use $\Delta = 2$ as the maximum scaling factor for watermark embedding. Using larger values for Δ heavily damaged perceptual quality of watermarked images.

In the first three simulation sets, we also analysed and discussed features of three different quality metrics (PSNR, wPSNR, and the Watson metric). Our simulation sets indicated that the features of an image significantly affects the perceptual performance of the watermarking algorithm.

Simulation set 3, where we used a sliding window that has a width of four decomposition levels, has indicated that the spectral range where we embed the watermark bits has no significant effect on the perceptual quality of the watermarked images. Combining that with the fact that using lower frequency levels to embed the watermark increases robustness against distortions, we conclude that it is more efficient to embed the watermark starting at second or third decomposition level.

The last two simulation sets measure the watermark resistance against mild rotation and additive Gaussian noise respectively. Rotation de-synchronizes the watermark detector and causes detection rates to drop to around 50% (indicating a detection failure) throughout all decomposition levels. Telltale watermark has no tolerance of RST distortions. Additive Gaussian noise only affects higher frequency decomposition levels (level 2 and 3) but generally the watermark is still detectable.

6 Conclusion

6.1 Thesis Summary and Conclusions

Semi-fragile telltale watermarks are essential to copyright monitoring and tamper detection of multimedia content. Watermarks that use the wavelet domain perform well in detecting any spatial or spectral tampering of watermarked images. This is due to the space-scale nature of the wavelet transform.

Wavelet-based semi-fragile watermarks are also tolerant of mild distortions such as JPEG compression and AWGN. Lower frequency decomposition levels demonstrate more resistance to such distortions. One important feature of these watermarks is that the original unwatermarked image is not required at watermark extraction time. This increases the applicability of such watermarks.

The use of perceptual quality metrics helps in tuning the algorithm parameters by setting acceptable perceptual limits. Another advantage of using perceptual models is highlighting the fact that the performance varies according to the features of the watermarked image. This implies that adaptive watermarks should perform better in general.

There are however some drawbacks using the wavelet domain for watermarking. These watermarks are not tolerant to timing errors. Synchronization is critical during watermark extraction. However, it should be emphasized that, by definition, semi-fragile watermarks are not intended to be robust against all types of distortions. Instead, a semi-fragile

telltale watermark is considered to be good if it can detect tampering and tolerate mild distortions.

6.2 Thesis Contributions

In this work, we present a performance analysis of image watermarking using a telltale wavelet-based semi-fragile watermark. The perceptual impact of watermark embedding was studied, and suitable range for algorithm parameters were suggested. The ability of the watermark to detect tampering was verified. The robustness of the algorithm against mild distortions has been studied by measuring watermark detection rates for various images and distortions.

A brief review of various techniques used for digital watermarking, attacks, benchmarking methods, and perceptual metrics mechanics have also been presented.

The following novel techniques have been proposed in this work:

- Using level-based analysis of decomposition levels coefficients in our simulations has proven to be more effective in distinguishing tampering from incidental distortions. This technique is different from that used in the original algorithm where the tamper threshold is calculated based on the overall detection rate of all decomposition levels.
- Applying a novel sliding-window technique to the watermark embeddor has been shown to improve the overall performance of the telltale algorithm. It allows for more flexibility in tuning the telltale algorithm so that an acceptable trade-off between sensitivity to tampering and robustness against innocent distortions is achieved. The original algorithm authors suggested that the watermark is embedded starting at

decomposition level 1. Our simulations indicate that levels 1 and 2 are less resistant even to minor distortions. Level 1 also requires more watermarking bits to completely cover the image and it has no specific advantage for tamper detection. The large number of watermark bits raises the issue of how will the watermark be transferred to the extractor for watermark verification. The smaller the size of the watermark, the more applicable the algorithm. It should be noted that the issue of how the watermark can be transferred to the extractor is a well-known challenge in cryptography, where two parties have to agree on a shared secret (encryption key or password) in order to establish a secure communication channel. Since that secure channel is not established at the beginning, the two parties have to use other techniques to agree on a shared secret through an unsecure channel without the risk of an adversary knowing the key and being able to eavesdrop. One example of a solution to that problem is The Diffie-Hellman key agreement protocol that uses exponentiation to allow two users to exchange a secret key over an insecure channel without any prior shared secrets.

- Detailed perceptual analysis of watermarked images using Watson's model TPE and LPE has helped to evaluate the perceptual performance of the telltale algorithm. Our simulations have experimentally determined that the highest value for the scaling factor Δ should be close to 2 in order not to visually damage the watermarked images.

6.3 Directions for Further Research

While good performance of telltale watermarks has been verified, there is room for more improvement. The ability of a telltale watermark to self-restore areas of the image that

have been tampered with is a valuable feature that can be added to such watermarks. This can be done by building a Hidden Markov Model (HMM) of the original image. The wavelet transform of real world signals is known to be a sparse one, where very few coefficients have large values while most other coefficients have values close to zero. This can be modeled as a mixture Gaussian distribution. Wavelet coefficients are also known to have *clustering*, where neighbouring coefficients of a large coefficient tend to be also large, and *persistence*, where large coefficients tend to propagate across decomposition levels. Using this information, a HMM can be built. We can use the parameters of that HMM as the watermark bits (after encrypting them to keep a noise-like profile). When the watermark is extracted, the HMM is constructed and any modified areas of the image can be re-built, with much lower resolution, using that HMM.

The use of cocktail watermarks promises to improve performance. For instance, consider a telltale algorithm that embeds watermark bits into the wavelet domain, while at a second iteration uses Fourier-Mellin domain to embed more watermark data. The distortions introduced by the Fourier-Mellin domain watermark will be dealt with by the wavelet domain extractor as mild noise. The result is a telltale watermark that is robust against mild RST distortions as well as JPEG compression and AWGN.

References

[1] J. Daemen, V. Rijmen. The Block Cipher Rijndael. Smart Card Research and Applications, LNCS 1820, J.-J. Quisquater and B. Schneier, Eds., Springer-Verlag, 2000, pp. 288-296.

[2] J. Daemen and V. Rijmen. Rijndael, the advanced encryption standard. Dr. Dobb's Journal , Vol.~26, No.~3, March 2001, pp. 137-139.

[3] RSA Laboratories' Frequently Asked Questions about Today's Cryptography, Version 4.1.

[4] S. Katzenbeisser, F. A. P. Petitcolas. Information Hiding: techniques for steganography and digital watermarking. Artech House, 1999.

[5] F. Petitcolas, R. Anderson, and M. Kuhn, Information hiding: A survey. Proceedings of the IEEE: Special Issue on Identification and Protection of Multimedia Information, 87(7), July 1999, pp. 1062-1078.

[6] R. Anderson. Stretching the Limits of Steganography. Information Hiding, volume 1174 of Lecture Notes in Computer Science, May 1996, Berlin, Germany, pp. 39-48.

[7] K. Tanaka, Y. Nakamura, and K. Matsui, Embedding Secret Information into a Dithered Multilevel Image. Proceedings of the IEEE Military Communications Conference, 1990, pp. 216 -220.

[8] B. Schneier. Applied Cryptography, 2nd edition. John Wiley & Sons, 1996.

[9] D. Kundur and D. Hatzinakos. Digital Watermarking for Telltale Tamper Proofing and Authentication. Proceedings of the IEEE, 87(7), 1999, pp. 1167-1180.

[10] B. Pfitzmann. Information hiding terminology. Information hiding: first international workshop, vol. 1174 of Lecture Notes in Computer Science, Isaac Newton Institute, Cambridge, England, May 1996, Springer-Verlag, Berlin, Germany, pp. 347-350.

[11] I. J. Cox, M. L. Miller, J. A. Bloom. Digital Watermarking. Morgan Kaufmann, 2002.

[12] M. Kankanhalli, et al. Adaptive Visible Watermarking of Images. Proc. ICMCS'99, June 1999, Centro Affari, Florence, Italy.

[13] F. C. Mintzer, et al. Toward on-line, worldwide access to Vatican Library materials. IBM Journal of Research and Development, vol. 40, No. 2.

[14] M. Kutter and F.A. P. Petitcolas. A fair benchmark for image watermarking systems. Electronic Imaging '99. Security and Watermarking of Multimedia Contents, vol. 3657, Sans Jose, CA, USA, 25-27 January 1999. The International Society for Optical Engineering, pp. 226-239.

[15] L. G. Brown. A Survey of Image Registration Techniques. ACM Computing Surveys, 24(4), 1992, pp. 325-376.

[16] C. Xu, J. Wu, and Q. Sun. Audio Registration and Its Application to Digital Watermarking. Security and Watermarking of Multimedia Contents, SPIE-3971, 2000, 393-401.

[17] D. Kundur and D. Hatzinakos. Towards a telltale watermarking technique for tamper proofing. Proc. IEEE Int. Conf. On Image Processing, vol. 2, 1998, pp. 409-413.

[18] N. Doraswamy and D. Harkins. IPSec: The New Security Standard for the Internet, Intranets, and Virtual Private Networks 1/e. ISBN: 0-13-011898-2. Prentice Hall. July 1999.

[19] D. Kilburn. Dirty Linen, Dark Secrets. AdWeek, volume 38, no. 40, October 6, 1997, pp. 35-40.

[20] ConfirMedia™ Broadcast Monitoring and Reporting System. 2001 Verance Corp. San Diego.

[21] S. Craver, N. Memon, B. Yeo, M. M. Yeung. Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications. IEEE Journal on Selected Areas in Communications, 1998, pp. 573-586.

[22] G. L. Friedman. The Trustworthy Camera: Restoring Credibility to the Photographic Image. IEEE Transactions on Consumer Electronics, 39 (4), 1993, pp. 905-910.

[23] R. B. Wolfgang and E. J. Delp. A Watermark for Digital Images. Proceedings of the 1996 International Conference on Image Processing, volume 3, 1996, pp. 219-222.

[24] F. Mintzer, G. W. Braudaway, and M. M. Yeung. Effective and Ineffective Digital Watermarks. Proceedings of the IEEE International Conference on Image Processing (ICIP '97), Santa Barbara CA, USA, vol. III, October 1997, pp. 9-12.

[25] S. Pereira. Robust Digital Image Watermarking. Computer Vision Group -CUI - University of Geneva, Geneva, Switzerland, 2000.

[26] H. Maître. Image Watermarking: Why is Watermarking a Hard Problem. Korea-France Workshop on Multimedia, Seoul, Korea, 1998, pp. 44-52.

[27] G. C. Langelaar, I. Setyawan, and R. L. Lagendijk Watermarking Digital Image and Video Data: A state-of-the-art Overview. IEEE Signal Processing Magazine, Vol. 17, No. 5, September 2000, pp. 20-46.

[28] R. J. Anderson, F. A. P. Petitcolas. Information Hiding: An Annotated Bibliography. Computer Laboratory, University of Cambridge.

[29] F. A. P. Petitcolas and R. J. Anderson. Evaluation of copyright marking systems. Proceedings of IEEE Multimedia Systems'99, vol. 1, 7-11 June, 1999, Florence, Italy, pp. 574-579.

[30] F. A. P. Petitcolas et al. A public automated web-based evaluation service for watermarking schemes: StirMark Benchmark. In proceedings of Electronic Imaging 2001, Security and Watermarking of Multimedia Contents, vol. 4314, San Jose, CA, U.S.A., 22–26 January 2001. The Society for imaging science and technology (I.S.&T.) & the international Society for optical engineering (S.P.I.E.), pp. 575-584.

[31] S. Voloshynovskiy, S. Pereira, V. Iquise, and T. Pun. Attack modeling: Towards a second generation benchmark. Signal Processing, Special Issue: Information Theoretic Issues in Digital Watermarking, May, 2001.

[32] A. B. Watson. DCT quantization matrices visually optimized for individual images. Human Vision, Visual Processing, and Digital Display IV, Proc. SPIE, 1993, pp. 1913-14.

[33] V. Solachidis, A. Tefas, N. Nikolaidis, S. Tsekeridou, A. Nikolaidis, and I. Pitas. A benchmarking protocol for watermarking methods. 2001 IEEE Int. Conf. on Image Processing (ICIP'01), Thessaloniki, Greece, 7-10 October, 2001, pp. 1023-1026.

[34] R. C. Gonzalez and R. E. Woods. Digital Image Processing, 2e. Prentice Hall, 2002.

[35] I. Avcıbaş, B. Sankur, K. Sayood. Statistical Evaluation of Image Quality Measures. Journal of Electronic Imaging, Vol. 11, April, 2002, pp. 206-223.

[36] F. Hartung and M. Kutter. Multimedia watermarking techniques. Proceedings of the IEEE, Vol. 87 (7), July 1999, pp. 1079 -1107.

[37] M. Kutter, F. Jordan, and F. Bossen. Digital signature of color images using amplitude modulation. Proc. Electronic Imaging 1997 (EI 97), San Jose, CA, Feb. 1997, pp. 518-526.

[38] D. M. Green and J. A. Swets. Signal Detection Theory and Psychophysics. Huntington, New York: Robert E. Krieger Publishing Co., 1974.

[39] G. K. Wallace. The JPEG still picture compression standard. IEEE Transactions on Consumer Electronics. Volume: 38 Issue: 1, Feb. 1992, pp. xviii -xxxiv.

[40] I. J. Cox. Spread Spectrum Watermark for Embedded Signaling. United States Patent 6,069,914, 2000.

[41] A. Watson, G. Yang, J. Solomon, J. Villasenor. Visibility of Wavelet Quantization Noise. IEEE Transactions on Image Processing, vol. 6, no.8, August, 1997, pp. 1164-1175.

[42] E. Terhardt. Calculating Virtual Pitch. Hearing Research. vol 1, 1979, pp. 155-182.

[43] J. L. Mannos and J. J. Sakrison. The Effects of a Visual Fidelity Criterion on the Encoding of Images. IEEE Transactions on Information Theory. IT-4, 1974, pp. 525-536.

[44] Z. Wang, A. C. Bovik and L. Lu. Why is Image Quality Assessment So Difficult? IEEE International Conference on Acoustics, Speech, & Signal Processing. May 2002.

[45] A. J. Ahumada Jr. and H. A. Peterson. Luminance-Model-Based DCT Quantization for Color Image Compression. Human Vision, Visual Processing, and Digital Display III, B. E. Rogowitz, ed. Proceedings of the SPIE. vol. 1666, 1992, pp. 365-374.

[46] B. Coşkun, U. Naci, Ö. Ekici, and B. Sankur. A Comparative Assessment of Semi-Fragile Watermarking Methods. SPIE Conf. 4518, Multimedia Systems and Applications IV, August 19-24, Denver, USA.

[47] M. W. Marcellin, M. J. Gormish, A. Bilgin, and M. P. Boliek. An overview of JPEG-2000. Proceedings, Data Compression Conference, March 2000, Snowbird, Utah, pp. 523-544.

[48] M. Ramkumar and A. N. Akansu. Information theoretic bounds for data hiding in compressed images. IEEE Second Workshop on Multimedia Signal Processing, 1998, pp. 267 -272.

[49] A. Westfeld and A. Pfitzmann. Attacks on Steganographic Systems. Information Hiding. Third International Workshop, IH'99, Dresden, Germany, September/October, 1999, Proceedings, LNCS 1768, Springer-Verlag Berlin Heidelberg, 2000, pp. 61-76.

[50] M. Goljan, J. Fridrich, and R. Du. Distortion-free Data Embedding for Images. Fourth International Information Hiding Workshop, 2001, pp. 27-41.

[51] C. Lu, S. Huang, C. Sze, and H. Liao. Cocktail watermarking for digital image protection. IEEE Transactions on Multimedia, Volume: 2 Issue: 4, December 2000, pp. 209-224.

[52] E. Lin, C. Podilchuk, and E. Delp. Detection of Image alterations Using Semi-Fragile Watermarks. SPIE-3971, 2000, pp. 152-163.

[53] D. Storck. A New Approach to Integrity of Digital Images. IFIP Conference on Mobile Communication, 1996, pp. 309-316.

[54] S. Bhattacherjee and M. Kutter. Compression-tolerant Image Authentication. IEEE International Conference on Image Processing, volume 1, 1998, pp. 435-439.

[55] M. Wu and B. Liu. Watermarking for image authentication. IEEE International Conference on Image Processing, October 1998, pp. 437-441.

[56] G. Strang and T. Nguyen. Wavelets and Filter Banks, Revised Edition. Wellesley-Cambridge Press, 1997.

[57] Y. T. Chang. Wavelet Basics. Kluwer Academic Publishers, 1995.

[58] X. Kang. Digital Color Halftoning. November 1999. Wiley-IEEE Press.

[59] M. Holliman. Counterfeiting Attacks on Oblivious Block-wise Independent Invisible Watermarking Schemes. IEEE Transactions on Image Processing, 2000.

[60] J. Fridrich and M. Goljan. Protection of digital images using self embedding. Proceedings of NJIT Symposium on Content Security and Data Hiding in Digital Media, (Newark, NJ), May 1999.

[61] F. Hartung, J. K. Su, and B. Girod. Spread Spectrum Watermarking: Malicious Attacks and Counterattacks. Proceedings SPIE, Security and Watermarking of Multimedia Contents, vol. 3657, Jan. 1999, pp. 147-158.

[62] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. A Secure, Robust Watermark for Multimedia. Proceedings of the First International Workshop on Information Hiding, Univ. of Cambridge, Lecture Notes in Computer Science, LNCS 1174, May 1996, pp. 243-246.

[63] I. J. Cox, J. Kilian, F. T. Leighton, and T. Shamoon. Secure Spread Spectrum Watermarking for Multimedia. IEEE Transactions on Image Processing, vol. 6, Dec. 1997, pp. 1673-1686.

[64] J. Fridrich. Image Watermarking for Tamper Detection. Proceedings of ICIP '98, Chicago, October 1998, pp. 404-408.

[65] C. S. Lu, S. K. Huang, C. J. Sze, and H. Y. M. Liao. A New Watermarking Technique for Multimedia Protection. Multimedia Image and Video Processing, L. Guan, S. Y. Kung, and J. Larsen, Eds. Boca Raton, FL: CRC, 2000 .

[66] M. A. Al-Mohimeed. Wavelet-based digital watermarking. In Proceedings of SPIE, Security and Watermarking of Multimedia Contents III, volume 4314, San Jose, CA, USA, January 2001, pp. 418-423.

[67] S. Voloshynovskiy, A.Herrigel, N.Baumgärtner, and T.Pun. A stochastic approach to content adaptive digital image watermarking. International Workshop on Information Hiding, Dresden, Germany, 29 September-1 October 1999, Lecture Notes in Computer Science, Ed. Andreas Pfitzmann, pp. 211-236.

[68] J. Fridrich. Methods for Detecting Changes in Digital Images. IEEE Workshop on Intelligent Signal Processing and Communication Systems. Melbourne, Australia. November 1998, pp. 173-177.

[69] J. L. Mannos, and J. J. Sakrison. The Effects of a Visual Fidelity Criterion on the Encoding of Images. IEEE Transactions on Information Theory, IT-4, 1974, pp. 525-536.