ROBUST WATERMARKING OF IMAGES

A THESIS SUBMITTED TO
THE GRADUATE SCHOOL OF NATURAL AND APPLIED SCIENCES
OF
THE MIDDLE EAST TECHNICAL UNIVERSITY

BY

SALİH EREN BALCI

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE
IN
THE DEPARTMENT OF ELECTRICAL AND ELECTRONICS ENGINEERING

SEPTEMBER 2003

Approval of Graduate School of Natural and Applied Science.

_____
Prof. Dr. Canan ÖZGEN
Director

I certify that this thesis satisfies all the requirements as a thesis for the degree of Master of Science.

_____
Prof. Dr. Mübeccel DEMİREKLER
Head of Department

This is to certify that we have read this thesis and that in our opinion it is fully adequate, in scope and quality, as a thesis for the degree of Master of Science.

_____
Assoc. Prof. Dr. Gözde B. Akar
Supervisor

Examining Committee Members:

Prof. Dr. Kemal Leblebicioğlu _____

Assoc. Prof. Dr. Gözde B. Akar _____

Assoc. Prof. Dr. Aydın Alatan _____

Assoc. Prof. Dr. Engin Tuncer _____

Ersin Esen _____

ABSTRACT

ROBUST WATERMARKING OF IMAGES

Balcı, Salih Eren
M.Sc., Department of Electrical and Electronics Engineering
Supervisor: Assoc. Prof. Dr. Gözde B. Akar

September 2003, 117 pages

Digital image watermarking has gained a great interest in the last decade among researchers. Having such a great community which provide a continuously growing list of proposed algorithms, it is rapidly finding solutions to its problems. However, still we are far away from being successful. Therefore, more and more people are entering the field to make the watermarking idea useful and reliable for digital world. Of these various watermarking algorithms, some outperform others in terms of basic watermarking requirements like robustness, invisibility, processing cost, etc.

In this thesis, we study the performances of different watermarking algorithms in terms of robustness. Algorithms are chosen to be representatives of different categories such as spatial and transform domain. We evaluate the

performance of a selected set of 9 different methods from the watermarking literature against again a selected set of attacks and distortions and try to figure out the properties of the methods that make them vulnerable or invulnerable against these attacks.

Keywords: digital watermarking, robust watermarking

ÖZ

DAYANIKLI İMGE DAMGALAMA

Balcı, Salih Eren
Yüksek Lisans, Elektrik ve Elektronik Mühendisliği Bölümü
Tez Yöneticisi: Doç. Dr. Gözde B. Akar

Eylül 2003, 117 sayfa

Sayısal imge damgalama son 10 yılda araştırmacılar arasında büyük ilgi gören konulardan biridir. Bu büyük ekibin ürettiği algoritmalar mevcut sorunlara hızla yeni çözüm önerileri getirmekle birlikte sayısal imge damgalama teknikleri henüz bütünüyle başarılı sağlamış değildir. Bu sebeple, konu büyük ekonomik değerler de taşıdığı için, sayısal dünyada kullanılabilir, güvenli algoritmalar geliştirmek isteyen araştırmacı sayısı gün geçtikçe artmaktadır. Bu pekçok sayıdaki algoritma arasından bazıları dayanıklılık, görünmezlik, ve işlem maliyeti gibi temel işaretleme gereklilikleri açısından diğerlerinden daha üstün performans sergilemektedirler.

Bu test çalışmasında değişik damgalama algoritmalarının dayanıklılığı üzerinde çalışılmıştır. Algoritmaların değişik kategorilerden seçilmesine özen

gösterilmiştir. Seçilen 9 adet algoritma üzerinde testler gerçekleştirilmiş, bu testlerin sonuçları ile algoritmaların güçlü ve güçsüz oldukları özellikleri arasında ilişkilendirmeler yapılmaya çalışılmıştır.

Anahtar Kelimeler: sayısal damgalama, dayanıklı damgalama

To my family,
my all time supporters

ACKNOWLEDGMENTS

First of all, I want to express my sincere gratitude to my supervisor, Assoc. Prof. Gözde B. Akar for her continuous help, guidance, understanding and interest throughout this work.

I feel that I can by no means repay the endless efforts of all members of my family to keep me motivated even from the first day of my primary school education until the last minutes of this work. I especially want to thank my mother for her willingness to keep me awake by sacrificing her sleep and to my father for his continuous support that puts me back to work when I felt tired. This work could not be completed without them.

Last but not least, I must thank my colleagues, managers and directors at ASELSAN, especially Tümer Doğan and Özgür Çelikoğlu, for understanding my absence, for their support by all means, and for convincing me that I am able to finish this work at times I was less sure about it.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ABBREVIATIONS

IP	Intellectual Property
CDMA	Code Division Multiple Access
DS-CDMA	Direct Sequence Code Division Multiple Access
DCT	Discrete Cosine Transform
DWT	Discrete Wavelet Transform
LPM	Log-Polar Mapping
FMT	Fourier-Mellin Transform
DFT	Discrete Fourier Transform
PSNR	Peak Signal-to-Noise Ratio
MSE	Mean Square Error
FFT	Fast Fourier Transform
AJ	Anti-Jamming
LPI	Low Probability of Intercept
HVS	Human Visual System
CWT	Complex Wavelet Transform

CHAPTER 1

INTRODUCTION

Recent years have seen a rapid growth in the availability of digital multimedia content. Today, digital media documents can be distributed via the World Wide Web to a tremendous number of people without much effort and money. Additionally, unlike traditional analog copying, with which the quality of the duplicated content is degraded, digital tools can easily produce large amount of perfect copies of digital documents in a short period. This ease of digital multimedia distribution over the Internet, together with the possibility of unlimited duplication of this data, threatens the intellectual property (IP) rights more than ever. Thus, content owners are eagerly seeking technologies that promise to protect their rights.

Cryptography is probably the most common method for protecting digital content since it has a well-established theoretical basis and developed very successfully as a science. The content is encrypted before delivery and a key is provided to the legitimate owner (who has paid for it). However, the seller is unable to discover how the product is handled after it is decrypted by the buyer. Encryption protects the content during the transmission only. When transmitted to the receiver, data must be decrypted in order to be valuable. Once decrypted, the data is no longer protected and it becomes vulnerable. The buyer may turn out to be a pirate distributing illegal copies of the decrypted (unprotected) content.

Therefore, encryption must be complemented with a technology that can continue to protect the valuable data even after it is decrypted. This is the point

where watermarking comes in. Digital watermarking technology is receiving increasing attention since it presents a possible solution for prohibiting copyright infringement of the multimedia data in open, highly uncontrolled environments where cryptography cannot be applied successfully.

A digital watermark is a distinguishing piece of information that is adhered to the data (generally called cover or host data) that it is intended to protect. Watermarking embeds (generally hides) a signal directly into the data and the signal becomes an integral part of the data, travelling with the data to its destination. This way, the valuable data is protected as long as the watermark is present (and detectable) in it. At any given moment, the hidden signal can be extracted to get the copyright-related information. Thus, the goal of a watermark must be to always remain present in the host data. However, in practice the requirement is somewhat weaker than that: Depending on the application, a watermark is required to survive all the possible manipulations the host data may undergo as long as they do not degrade *too much* the quality of the document.

The main difference between watermarking and encryption is that encryption disguises the data and protects it by making it unreadable without the correct decryption key, while watermarking aims to provide protection in its original viewable/audible form.

Watermarking, like cryptography, needs secret keys to identify legal owners. The key is used to embed the watermark, and at the same time to extract or detect it. Only with a correct key can the embedded signal be revealed. While a single bit of information indicating that a given document is watermarked or not is sufficient sometimes, most applications demand extra information to be hidden in the original data. This information may consist of ownership identifiers, transaction dates, logos, serial numbers, etc., that play a key role when illegal providers are being tracked.

Watermarking can be used mainly for owner identification (copyright protection), to identify the content owner; fingerprinting, to identify the buyer of the content; for broadcast monitoring to determine royalty payments; and authentication, to determine whether the data has been altered in any manner from its original form.

While digital watermarking for copyright protection is a relatively new idea, the idea of data hiding dates back to the ancient Greeks and has progressively evolved over the ages. An excellent survey of the evolution of data hiding technologies can be found in [78]. The inspiration of current watermarking technology can be traced to paper watermarks which were used some 700 years ago for the purpose of dating and authenticating paper [79].

It was 1988, when the term *digital watermark* is first used by Komatsu and Tominaga [13]. Watermarks in the context of digital images first appeared in 1990 [77]. The first Information Hiding Workshop, which included digital watermarking as one of its primary topics [14], was held in 1996. Beginning in 1999, SPIE began devoting a conference focused on *Security and Watermarking of Multimedia Contents* [15]. The Copy Protection Technical Working Group (CPTWG) evaluated some watermarking systems for their possible usage as a method of protecting DVD video. The Secure Digital Music Initiative (SDMI), adopting the technology of the Verance Corporation, made watermarking a basic component of their portable device specification [16]. The same technology is also used by some Internet music distributors like Liquid Audio. Two projects, VIVA [17] and Talisman that are funded by the European Union, tested watermarking schemes for their suitability for broadcast monitoring.

In the area of image watermarking, some commercial watermarking products already emerged on the market. Digimarc Corporation's *Picture Marc* is available as a tool in Adobe Photoshop image processing program. The detector can find watermarks in the images if they were embedded into the image by the Digimarc product.

In this thesis, our aim is to investigate the robustness of some selected watermarking algorithms. In the following chapters, first, we provide a basic look into the problem of watermarking by means of its applications and requirements. Next, we set out to give the major ideas about watermarking and try to establish the general framework of the watermarking system by giving real world examples and some experimental results to strengthen the underlying ideas. We handle the basic

building blocks of classical watermarking systems; namely embedding, the channel (attacks and distortions) and detection, one by one and try to explain their properties, limitations, etc.

Finally, we present an evaluation of 9 of the existing watermarking algorithms from the literature. The algorithms that we examine differ from each other in their some basic properties such as embedding domain, embedding rules, host feature selection, etc. We give plots, figures about their robustness against a number of attacks implemented generally by Stirmark v3.1 [95, 96] and MATLAB and try to end up with generalizations and comments on the relationship between their algorithmic properties and their performance in terms of robustness.

CHAPTER 2

APPLICATIONS AND PROPERTIES

2.1. Applications

In this section, some insight to possible watermarking applications and their requirements are examined.

The major applications of watermarking are owner identification/proof of ownership [2-5], authentication (also referred as content verification, data integrity or tamper proofing) [6, 7], transactional watermarks, copy control, covert communication, and broadcast monitoring [8-12].

2.1.1. Owner Identification and Proof of Ownership

A traditional copyright notice in the form of "©*date, owner*" added on an image or a video frame is no longer a safe way of guaranteeing copyrights [20]. Although such annotations are still recommended, they can easily be cropped out or processed hence, removing or altering the ownership information. Hence, copyright violation harms the interests of the providers rather than those of customers.

Since a digital watermark, once embedded, becomes an imperceptible and inseparable part of the host data, it can be used to provide copyright marking

functionality. The intellectual property owner adds his/her copyright information in the form of a high fidelity, robust, and secure watermark. Even if the document undergoes manipulations (either intentional or unintentional) it will, ideally, still be possible to extract or detect the watermark as long as the host data is in a valuable form. Moreover, the image will look more appealing to the eye since it does not need to have a textual notice that may make it aesthetically disturbing.

The level of security required to prove ownership is higher than that required for owner identification. Proving ownership means proving that a document owns to someone and that it does not belong to anyone else. This comes from the fact that a pirate can undermine the original watermark without removing it. This is pointed out by Craver *et al.* [19]. Bob, the pirate, using his own watermarking system, might be able to make it appear as though his watermark is present in Alice's (the IP owner) original copy of the image. Thus, a third party would be unable to judge whether Alice or Bob had the true original.

The solution proposed by watermarking is to change the way the problem is stated. Instead of trying to find the original it tries to find which image is derived from another. This approach provides indirect evidence saying that the document in question is owned by Alice, rather than Bob, because Alice has the copy from which others are created.

The first known application of owner identification using watermarks belongs to the Muzak Corporation [48]. Their system encoded identification information in audio signals. They blocked the 1 kHz band of the audio signal with varying durations using a band-notch filter to encode the letters in the Morse code. The system remained in use until the early 1980's [49].

2.1.2. Transaction Tracking (Fingerprinting)

Transactional watermarks, also called fingerprints, allow an IP (intellectual property) owner or content distributor to identify the source of an illegal copy by marking each legal copy of the document with a separate, unique watermark. If a document marked with a transaction watermark is misused (distributed illegally), the owner can find

out who is responsible.

There are two well-known real-world applications for fingerprinting. One is the distribution of movie dailies. A movie daily is the result of each day's photography and they are distributed to a number of people involved. Although these dailies are highly confidential, occasionally a daily is leaked to the press. The watermark, being different in each copy, serves as a tracker to find the source of leakage.

The other application was deployed by the now-defunct company DivX. Actually, they designed and marketed a new player, which placed a unique watermark to each video it played. If someone makes copies of this film after it is viewed then the watermark would appear on all the copies identifying the player on which it was played. If those copies are sold on a black market, then the DivX could obtain one of the copies and find the adversary or at least the player of the adversary. Since the corporation ceased the business probably, no watermark could be traced [20].

2.1.3. Content Authentication

With the advance of computer tools available for digital signal processing, modifying a digital document is becoming easier while detecting that the content is modified becomes harder.

Message authentication problem has been well studied in cryptography [22]. The solution offered by cryptography is a *digital signature*, and it is a widely accepted method. Only the authorized source knows the valid key for encryption, an adversary who tries to change the message cannot create a corresponding valid signature for the modified message.

The disadvantage of digital signatures is that they must be padded as metadata to the original data as separate information before transmission. It is thus easy to lose the signatures during daily usage, even without any bad-mannered operations. Format conversion is the best example to those situations. If the signature is saved to the header fields of some data format (e.g. JPEG), then it will be

discarded when we switch to some other format with no space for a signature in the header. When the signature is lost, the work can no longer be authenticated.

The superiority of watermarking comes at this point. Since the watermark information is carried directly on the bits of the original work, we do not lose them if the header field of the digital file is removed. Some authentication marks are designed to become invalid at the slightest modification on protected data. Those marks are called *fragile watermarks*.

In signature embedding systems, signature calculation is host signal dependant since a signature is a summary of the data to be protected. However, when we embed the signature in the data, the data content is modified. Even this modification is small; the signature might no more represent the modified data. To overcome this problem it is suggested to separate the data into two parts: one for signature calculation, and one for signature embedding [28, 29, 30]. As an integral part of the host data, the watermark is modified in the same way as the original, watermarked data. Here comes one more advantage of detecting tampering using watermarks: the possibility of learning more about how the data is modified.

2.1.4. Broadcast Monitoring

Commercials are vital for the survival of radio and TV channels. Companies book a specified time of the air on a specified time of the day and introduce their products to the audience and they pay for the time they book. The price they pay also varies from time to time in a day. Booking noon hours is much cheaper than booking the evening hours, which they call prime time. Therefore, companies carefully plan and prepare their commercials and put great importance on them. That's why the companies have been using a broadcast monitoring system dating back at least 1975 [32]. It is not the commercials only that need to be monitored. Some news items may have hundreds of thousands of dollars value per hour. This makes them very vulnerable to intellectual property rights violations.

Another usage of broadcast identification data is in competitive market research [38]. A company may want to know how much the competing company is

investing on its new brand in the market and adjust its own marketing policy according to this data.

A third possible application is the detection of illegal (unauthorized) rebroadcasts of copyrighted material by pirate stations. Intellectual property owners will be more interested in these types of systems [36].

One of the available products offering watermarking based broadcast monitoring and verification solutions is Verance's ConfirMedia™ [40].

2.1.5. Copy Control

The applications that we mentioned so far have the philosophy of proving that a copyright infringement has occurred instead of trying to prevent the infringement to occur. The watermarks we mentioned are utilized after we suspect some content is modified or distributed illegally. However, with the help of an intelligent hardware we can have control over the duplication, modification, and distribution processes, which are the main sources of illegal action. Copy control technologies may serve as a deterrent against such actions.

Encryption is once again a solution to the problem. The content is distributed to legitimate users in an encrypted format and only these users have a unique key for decryption. The key is in a special format that is difficult to duplicate and distribute. For example, satellite TV broadcast companies gives its customers a smart card (very much like the SIM cards of cellular phones), which is inserted into the decoder box, serving as the key. Without the key, the decoder cannot decrypt the incoming signals and all you can see is scrambled video.

An encryption-based system cannot prevent pirating of the data after a legal customer with a legally received key decrypts it. That is the weakest point of a cryptographic protection mechanism.

Therefore, we need technologies that allow the media to be viewed, but prevent it from being recorded. Two examples are the Analog Protection System (APS, developed by Macrovision [43]), which modifies the video signal in such a way as to confuse the automatic gain control on VCRs, and the Copy Generation

Management System in DVDs, that consists of a pair of bits in the header of the MPEG stream, which encode the copying permissions.

There are commercial copy control softwares already in the market. Actually, what is offered by MarkAny [45] is much more than just preventing illegal copying. Their product relies on watermarks to control Open, Download, and Print functions according to user authority even after content is opened by unauthenticated user.

2.1.6. Device Control

Device control is a broad category of applications in which specially designed devices react to the watermarks they detect in content. The earliest applications of device control with watermarks date 50 years back.

In 1953, Tomberlein *et al*. [50] described a system to distribute music to offices, stores and other premises. Broadcast is watermarked to mark the beginning and end of commercials, talk and other stuff other than music so that they are ignored and not aired in the office, store, etc.

R. H. Baer of the Sanders Associates Inc. [51] was issued a patent in 1976 for a video watermark intended for interactive television applications. Another patent was awarded to Broughton and Laumeister in 1989 for another interactive television application [52]. Their technique allowed action toys to interact with television programs.

Another patent in device control area was awarded to Ray Dolby of Dolby Labs [53] in 1981. Dolby-FM was a noise-reduction technique used by the radio stations. Some radios were equipped with special decoders to fully exploit Dolby-FM. Dolby invented a watermark to be inserted into Dolby-FM broadcast that will automatically turn on the special decoder circuit in compatible radio receivers.

A more recent application is Digimarc's MediaBridge system, [54] which utilizes watermarks to make a computer respond to watermarked documents that are shown to a compatible web camera. You just hold the printed piece (contains a watermark) up to your Digimarc compatible web camera and Digimarc MediaBridge technology will take you the associated web site without typing or clicking.

2.2. Properties

Watermarks are generally desired to satisfy some requirements [85], [79], [86] like robustness, tamper resistance (security), high capacity, fidelity, low computational cost, low false positive rate, etc. However, it is probably impossible to design a watermark that excels at all of these. The properties a watermarking scheme also depend greatly on the application at hand. Therefore, the properties a watermark should have are decided according to the application for which the watermark is designed. It is not necessary to have the best tamper resistance properties in a watermark that will be used, for example, to annotate the pictures you have in your home computer. For such an application, you may desire to have a watermark that betters in imperceptibility (fidelity) and capacity. Thus, it would be unfair to evaluate the properties of two such watermarking schemes according to the same standards.

In this section, we will examine those properties in detail.

2.2.1. Robustness

We are living in a "hacking" world where it is very common that movies (actually all types of multimedia), software, documents, etc are duplicated and distributed without paying anything to their intellectual owners. This growing amount of illegal copying and distribution is the motivation that emerged the field of robust watermarking.

Robustness is an important issue for the watermarks that are not specially designed to be *fragile*. In general terms, a robust watermark is the one that resists (remains detectable after) common signal processing operations. These operations include both the ones that may be a result of everyday usage of the document (spatial filtering, lossy compression, printing and scanning (D/A/D conversion), re-sampling, cropping, etc) and intentional attacks [108, 91, 92] intended solely to remove the watermark. Video and audio watermarks must also need to be robust to many of these transformations and to some specific ones like recording to tape (D-to-A conversion) and changes in playback rate. We also need to add the combinations of those transformations to the list.

In addition, when copies of the same content exist with different watermarks, as would be the case for fingerprinting, watermark removal is possible because of the collusion[1] between several owners of copies. It is applicable when a small number of different copies (about 10) are available to an attacker.

In general there should be no way in which the watermark can be removed or altered without sufficient degradation of the perceptual quality of the host data so as to render it unusable.

The properties stated above look very demanding. Only some extreme applications, in which the signal processing between the embedding and detection is unpredictable, require robustness against every possible distortion that does not destroy the value of the cover data. Fortunately, in general, a watermark is not required to be robust against all possible manipulations. The robustness requirement is always finalized according to the application.

For some applications like broadcast monitoring or covert communication, a watermark is supposed to survive lossy compression and transmission channel effects (low-pass filtering and additive noise), that is, the processing until the receiving party detects the watermark. It does not need to survive rotation, scaling, cropping, high-pass filtering, or other types of distortions that are not likely to occur during broadcast or travelling through a communication channel. After the watermark is detected successfully at the destination, either the watermarked data is erased or it becomes worthless, so protection is no more needed for that data.

In authentication applications, robustness is completely undesirable. Instead, a fragile watermark is required. We use fragile watermarks to understand if the data has been altered since it was watermarked. Some fragile image watermarks can also spatially locate the tampered area. This way, one can understand which part of the image is no more authentic. Therefore, for authentication applications, we want the watermark to disappear or to behave in a specific manner after a modification. In some of those cases, it is more desirable to have a *semi-fragile* watermark to resist

[1] A brief explanation of collusion attack is given in Section 3.3.1

innocent operations like compression but “break” if the manipulations threaten the data integrity (such as replacing a portion of the image).

2.2.2. Watermark Payload

Capacity of a watermarking system is the maximum amount of data (generally stated in bits) that can be inserted in the cover data. When we talk about the capacity, we implicitly impose the fidelity requirement. Because embedding bits into a host signal requires modifying some of the host characteristics and the watermarked signal deviates from its original.

In order to embed large amount of data into a multimedia signal without too much affecting the fidelity, properties of the human visual system (HVS) [90] are utilized. Properties of the human visual system give us clues about the components (either in pixel domain or in a transform domain) that do not have a large effect on perceptual quality of the document. Then we can design the watermark such that it modifies those unpercepted components in large amounts and other perceptually significant components are less affected.

2.2.3. Fidelity

A high fidelity watermark is a well-hidden signal such that it does not cause a perceptible degradation in the host (cover) data.

Fidelity and quality are different terms and must not be confused. Fidelity is a measure of the similarity between signals before and after processing. Quality, on the other hand, is an absolute measure of appeal. It is possible to have high quality together with low fidelity and vice versa. You can watermark a greyscale, highly compressed, low resolution (hence low quality) video and it may well be impossible to distinguish this watermarked version form the original (hence high fidelity).

If a watermark is not specially designed to be visible (Figure 2.1) [18], then it should not degrade the perceived quality of the work. That is, the perceptual similarity between the original and watermarked versions of the document must be as

high as possible. This immediately implies the need for a good quality metric. It has been shown [88, 89] that measures based on perceptual models yields more satisfactory results than the pixel-based models.

Figure 2.1 A visible watermark. The Lena image is visibly watermarked using the METU emblem.

The capacity and fidelity requirements are also application dependant. For some applications, we need to insert as much as we can without affecting the audio or visual aesthetics while in some cases fidelity is sacrificed for capacity. Such an application is the movie dailies distributed to those involved in film production. These dailies are highly confidential, yet occasionally; a daily is leaked to the press. The watermark, being different in each copy, may serve as a tracker to find the source of leakage. Since the purpose of distributing a daily is just to inform the related people about the material shot so far, they need not be of top quality and a small visible distortion caused by a watermark will not cause a loss in usefulness. Applications demanding high quality like DVD and HDTV require watermarks with much higher fidelity.

In some modes of communication like NTSC and AM radio, signal quality reaching the end user is low. Therefore, knowing that the data will be degraded anyway before it is viewed, the embedder can ignore some small artifacts caused by

the added watermark on the outgoing signal.

It is apparent that the requirements for obtaining high fidelity and high capacity embedding are conflicting. Hence it is not possible to have them both at the maximum.

There is another term closely related to capacity. It is the *data payload* which refers to the number of bits encoded by the watermark within a unit of time or within a document. For audio, data payload refers to the number of embedded bits per second that are transmitted.

2.2.4. Security

The security of watermarking techniques can be interpreted the same way as the security of encryption systems. For a watermarking technique to be truly secure, an unauthorised party should not be able to detect or remove an embedded watermark under the assumption that (Kerckhoff's assumption [109]) the algorithms for embedding and extracting the watermark are exactly known since they are publicly available.

This requirement can be fulfilled in cryptography by the use of a secret *key*. Keys can be thought of some information (possibly random) that determine how messages are encrypted. A message encrypted by a given key can only be decrypted with the same key. Many watermarking systems are designed to use secret keys in an analogous manner. In such systems, the method by which messages are embedded in watermarks depends on a key, and a matching key must be provided to the receiver side to detect those marks.

However, the security requirements for watermarks are still somewhat different from those for ciphers. Ciphers prevent unauthorized reading and writing of documents, in that form, they can prevent certain types of attacks, but they do not provide protection for watermark removal. Removing the watermark or masking it so that it can no more be extracted is analogous to the problem of signal jamming in military communications.

To provide resistance to jamming and interference *spread spectrum*[2] communications is developed for military applications. Spread spectrum is defined as [109] a means of transmission in which the signal occupies a bandwidth in excess of the minimum necessary to send the information; the band is accomplished by a code which is independent of the data, and a synchronized reception with the code at the receiver is used for despreading and subsequent data recovery. The exact form of the spreading is a secret known only by the transmitters and receivers.

[2] Some properties of spread spectrum techniques are presented in Appendix A.1.

CHAPTER 3

THE WATERMARKING SYSTEM

Watermarking is, in essence, a form of communication where the sending side wishes to communicate a message from the watermark embedder to the watermark receiver. Therefore, it was instant and inevitable to try to fit watermarking into the traditional model of a communications system given in Figure 3.1. The encoder and decoder keys are not a part of the traditional model. They are added when secure communication is required.

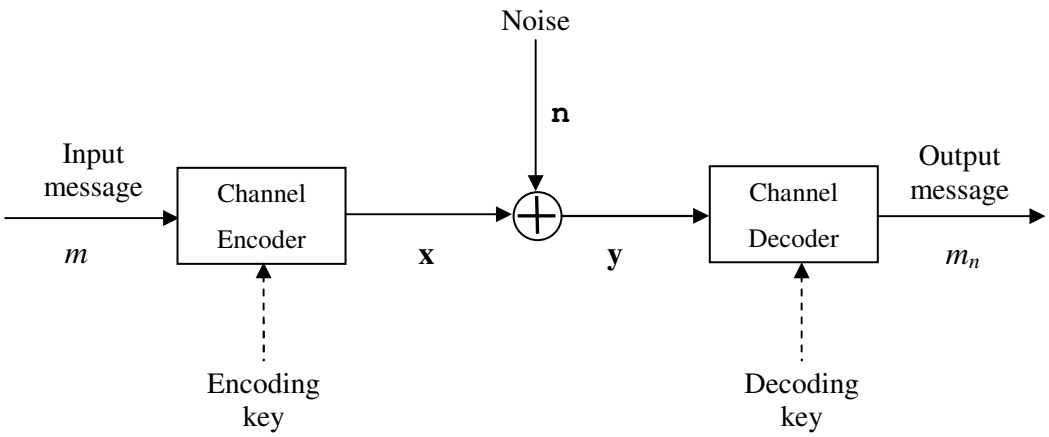


Figure 3.1 Standard model of a communications system.

When looked at as a communication task [74], the watermarking process can be split into three main steps: watermark generation and embedding (information transmission), possible attacks (transmission through the channel), and watermark retrieval (information decoding at the receiver side).

There are two ways (models) in which a watermarking system can be mapped into the communications system above. The two models differ in the way they use the cover (original) image, in which we insert the watermark.

The first model interprets the cover image, **I**, as noise. Whether this noise will be used as side *information* or not is a matter of choice. If it is used as a side information (shown as a dashed arrow in Figure 3.2, 3), then the watermark becomes a function of the cover image and the key. A watermark embedder which accepts the original image as input is called an *informed embedder*, in contrast to a *blind embedder* which produces the watermark regardless of the original image content.

The second model regards the cover image not as a part of the transmission channel but as a second message to be transmitted along with the watermark in the same signal $\mathbf{I}_w$. The model places two receivers for each component of the transmitted "composite" signal: a human being for the original image and a watermark detector for the watermark. This model of watermarking is similar to the traditional communication systems like time-division or frequency-division multiplexing which transmit multiple messages over a single line. On the receiver side, the human receiver should perceive something close to the original cover image with ideally no interference from the watermark and the watermark detector should obtain the watermark message with no interference from the cover image. In the following discussion we will take the first model as our basis.

The duty of the embedder is first to map the massage *m* to a pattern (with the help of a watermark key, *k*) suitable for adding to the cover image and then to add the pattern, **w**, to the original image in a suitable way. As a result of this process, the watermarked image, $\mathbf{I}_w$, which is going to be communicated, is produced.

After the watermarked image is transmitted, it is processed in some way, which we model as an addition of noise, **n**. However, actual processes may be

different than that. The watermarked image might go under compression, decompression, digital-analog-digital conversions, audio or visual enhancements, and even malicious attacks which intend to remove the watermark.

At the receiving side, the watermark detector may be of one of the two configurations: blind (Figure 3.2) or informed (Figure 3.3). If we are using an informed detector, the detection process consists of two steps. First, subtraction of the original image made available at the detector to obtain a noisy watermark pattern, $\mathbf{w}_n$. Then the pattern is decoded with the watermark key to extract the message as m_n. Since the original image is subtracted from the received image, we can ignore the addition of the original image at a blind embedder, and the system looks very similar to the system in Figure 3.1.

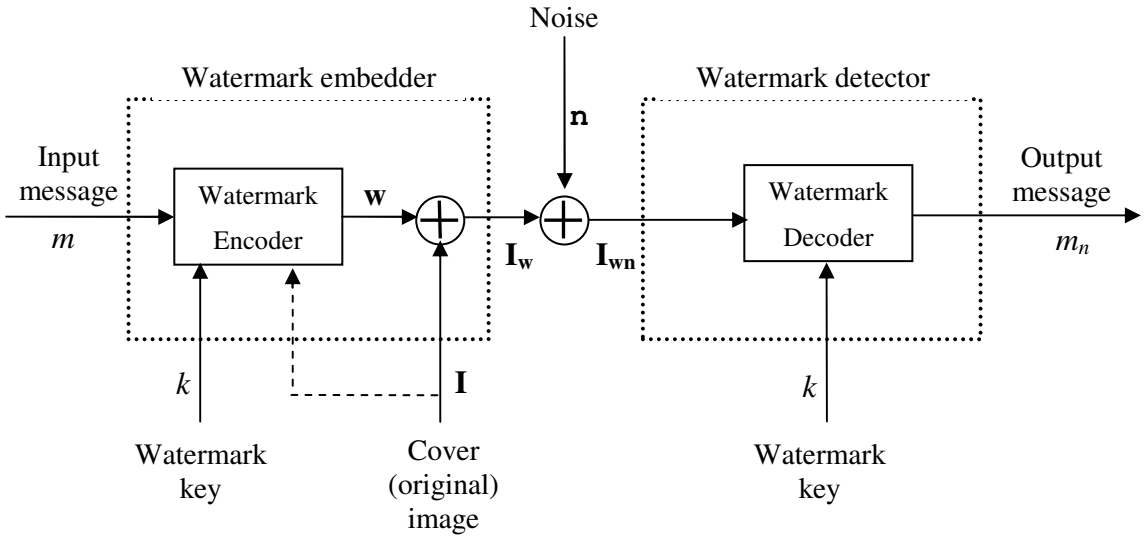


Figure 3.2 Watermarking system with a blind/informed embedder and a blind detector mapped into the communications model.

In a blind watermark detector (Figure 3.2), the cover image is unknown, and therefore cannot be removed prior to decoding. In this case, to make the analogy with Figure 3.1, we may say that the added pattern (which conveys the information to be communicated) is corrupted by the combination of the cover image, **I**, and the noise signal, **n**, added during transmission. The received watermarked image, $\mathbf{I}_{wn}$, is now viewed as a corrupted version of the added pattern, and the entire watermark detector is viewed as the channel decoder.

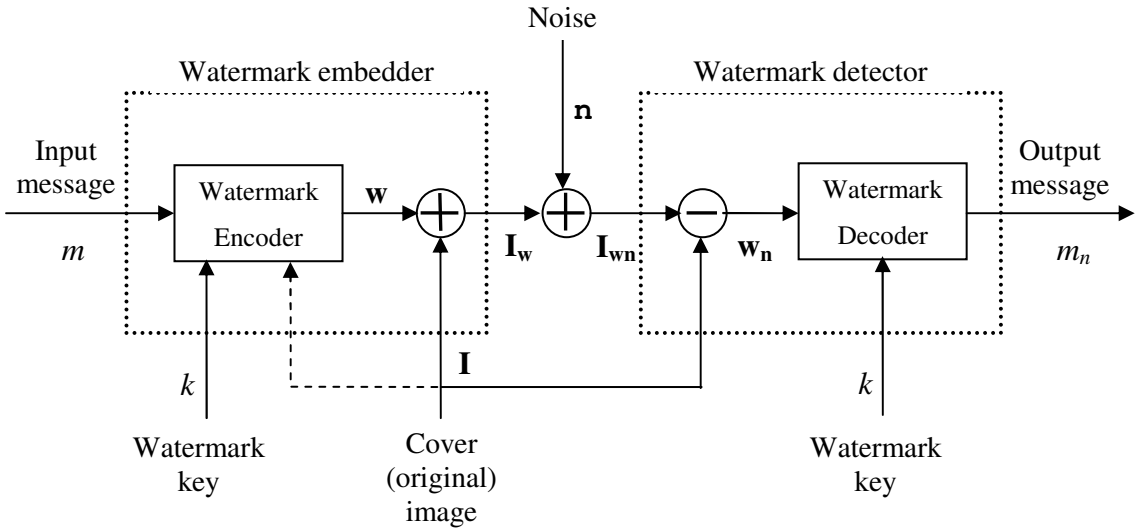


Figure 3.3 Watermarking system with a blind/informed embedder and an informed detector mapped into the communications model.

Generally, watermark encoder and decoder blocks in Figures 3.2 and 3.3 are huge ones containing many sub-blocks. Most watermarking schemes utilize the properties of some transform domain for higher robustness, invisibility or capacity. Therefore, the encoder and decoder blocks contain forward and reverse transform blocks and also further image processing and filtering blocks to improve the algorithm performance.

Keeping this generalized system in mind, we can start to examine each block separately and in detail.

3.1. Watermark Embedding

The first step in the designing of a watermarking system is the definition of the embedding procedure. This is a crucial task, since watermark properties highly depend on the way the watermark is inserted within the data.

3.1.1. Where to hide?

If watermarking can vaguely be described as “hiding a set of data under the coverage of some other set of host data”, then first question that comes to mind is where to hide it.

Although some of the early watermarking schemes like LSB modification techniques [55, 120] utilized the spatial (pixel) domain of the image to embed a watermark, the most successful algorithms make the watermark embedding in a suitable transform domain other than original image space. Then the transform coefficients are modified instead of directly changing the pixel values. The transforms commonly used for watermarking purposes are the discrete cosine transform (DCT), discrete Fourier transform (DFT, magnitude and phase [118]), discrete wavelet transform (DWT), and the fractal transform [81]. There are also less familiar transform domains used such as Fresnel transform [80], the complex wavelet transform (CWT), and Fourier-Mellin transform [75].

We can generalize the total watermark embedding process in the two steps [74]: first extracting a set of features (host features) from the host image, and then by modifying them according to the watermark content. The choice of the host features and the definition of the embedding rule have implications on watermark robustness and imperceptibility, which are the main concerns and challenges of the watermark embedding process.

The joint achievement of watermark imperceptibility and robustness requires that the main properties of Human Visual System (HVS) are utilized. HVS is a deep topic commonly used for perceptual coding of multimedia signals [76]. The main reason of utilizing HVS is to find a way of hiding more information with more

energy without disturbing the visual quality of the image. If the characteristics of the HVS are not taken into account, too weak a watermark would be inserted due to the invisibility requirement.

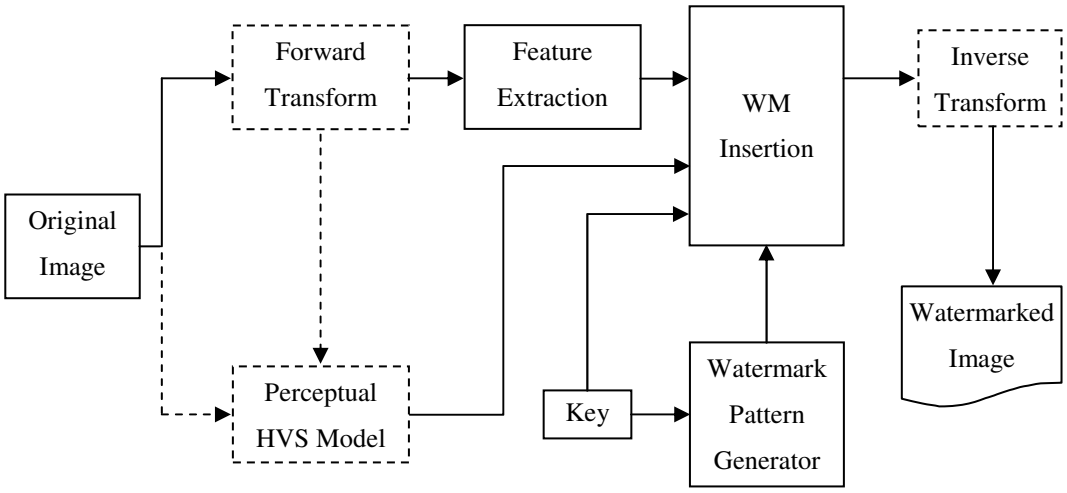


Figure 3.4 Detailed block diagram for watermark embedding. Dashed arrows and blocks suggest optional paths or processes that can be skipped.

Taking into account the discussion unto now, we give in Figure 3.4 a more detailed and specific block diagram for the watermark embedding operations.

The exploitation of the HVS properties can be pursued either implicitly, by properly choosing the embedding domain (like DWT) and the embedding strategy (like modifying low- to mid-frequency range); or explicitly, by using a visual masking method that shapes the watermark pattern according to the HVS considerations.

3.1.1.1. Selecting Host Features

Once again restricting our scope to the image case, we will question the set of host features that is most suitable for embedding watermark information. While doing that

we have to keep in mind the two important requirements for effective watermarking[3]. First one is robustness to signal processing alterations that intentionally or unintentionally attempt to remove or alter the watermark information. Secondarily, many watermarking applications require a scheme where the watermark modifications do not alter the perceptual quality of the host signal[4].

The first choice in selecting the host features is the watermark embedding domain. The watermark can be applied directly to the original signal space (spatial domain) or in some transform domain which presents some good perceptual characteristics and/or offers robustness to certain signal processing operations. Embedding the watermark in the original signal space is desirable for the sake of low complexity, low cost and low delay.

Block based DCT became a popular embedding domain since it is a basic component of image and video compression standards such as JPEG, MPEG and ITU H.26x family of coders. If we choose an embedding domain that matches that of standard compression systems, we can design the watermarking scheme to avoid adding the watermark information to the coefficients that are likely to be removed or coarsely quantized, resulting in a scheme robust to compression.

Furthermore, the sensitivity of the HVS to the DCT basis images has been extensively studied, which resulted in the recommended JPEG quantization table [124]. These results can be used for predicting and minimizing the visual impact of the distortion caused by the watermark.

Using that kind of 'matching' transforms, makes it possible to embed the watermark real-time on a compressed bitstream. Especially for some video applications, where the video will most likely be in some compressed form such as MPEG2, such a capability is usually very desirable.

---

[3] The term "watermarking" alone, in this text, will correspond to robust watermarking. Not all watermarks are desired to be robust. Some applications require fragile watermarks. We will explicitly state the term "fragile" if we specifically talk about those kind of watermarks.

[4] There are also some applications where visible watermarks are used. Actually, historically all watermarks were visible. The most common recent application is the TV channel logos visibily embedded over valuable television broadcast materials.

Another transform common to watermarking and compression techniques is the Discrete Wavelet Transform[5] (DWT) which is the basis of zero-tree wavelet coding (EZW, to be included in the upcoming image and video compression standards such as JPEG2000), SPIHT, and MPEG-4. This makes DWT a popular tool for watermarking.

Advantages of DWT are numerous. First of all is its multiresolution characteristics and hierarchical structure. In the case when the received data is not distorted significantly, the cross correlations with the whole size of the image may not be necessary and much of the computational load may be saved.

Another advantage of DWT is that it inherently separates the perceptually significant and insignificant components of an image. Human eyes are not very sensitive to the small changes in the edges and textures of an image but are very sensitive to the changes in the smooth areas of the image. With the DWT, the edges and textures are confined to the high frequency subbands, such as HH, LH, HL, etc (Figure 4.4). Therefore, modifying the large coefficients in these bands for watermark embedding does not generally create visual disturbances on the image.

Some image transforms show immunity or invariance to some kind of image processing operations. The obvious example is the shift-invariance of the amplitude of the Discrete Fourier Transform (DFT) coefficients and many watermarking techniques use DFT amplitude modulation because of this property. Because cyclic translation of the image in the spatial domain does not affect the DFT amplitude, the watermark embedded in this domain will be translation invariant. Furthermore, DFT divides the image into frequency bands and the watermark can be embedded directly into the significant middle frequencies since the modulation of the lowest frequency coefficients results in visible artifacts while highest frequency regions are very vulnerable to noise, filtering and lossy compression. However, the symmetry of the Fourier coefficients must be preserved to ensure that the image data is still real-valued after the inverse transform (IDFT). That is, if the coefficient $\left|\mathbf{I}(u,v)\right|$ in an

[5] A brief review of DWT is presented in Appendix A.2

image with $N \times M$ pixels is modified, its counterpart $\left|\mathbf{I}(N-u, M-v)\right|$ must also be modified in the same way.

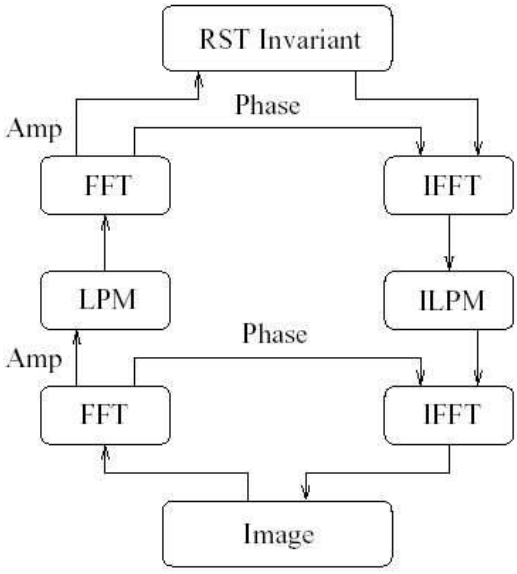


Figure 3.5 Diagram of RST watermarking scheme involving log-polar mapping.

Similarly, applying a watermark in the Fourier-Mellin Transform (FMT) domain results in a watermark that is invariant to image translation, scale and rotation [75]. FMT is Fourier Transform followed by a non-linear, irreversible (lossy) transformation called log-polar mapping (Figure 3.5). The main idea behind log-polar mapping is to find a presentation in which rotation and scaling operations are converted to linear shifts. This transformation maps the spatial coordinate axis (x, y) to polar axis (μ, θ) using the forward (Eqn 3.1, upward arrows in Figure 3.5) and inverse (Eqn 3.2, downward arrows in Figure 3.5) transformation equations:

$$\begin{aligned} \mu &= \frac{1}{2}\ln\left(x^2 + y^2\right) \\ \theta &= \tan^{-1}\frac{y}{x} \end{aligned} \tag{3.1}$$

$$x = e^{\mu} \cos \theta$$
$$y = e^{\mu} \sin \theta \qquad (3.2)$$

The forward transform converts a scaling in the form of $(\lambda x, \lambda y)$ into a translation (shift) expressed in terms of polar coordinates as $(\mu + \log \lambda, \theta)$, and a rotation of δ degrees is converted to a shift in the θ axis stated as $(\mu, \theta + \delta)$. Hence, if we apply Fourier transform to the log-polar representation, we obtain a rotation and scale invariant domain because of the shift invariance property of Fourier transform.

The problem in this theoretically elegant method lies in its implementation. When applied on a digital image, the transformations require a lot rounding because of the trigonometric and logarithmic operators. This rounding causes a large amount of loss in the data which results in failure to successfully inverting the transformation.

3.1.1.2. Spread Spectrum Coding

In the frequency domain, the idea of redundant embedding leads to the well-known spread spectrum paradigm. In a spread spectrum system, messages are encoded into symbols which are transmitted as pseudo-random sequences of 1s and 0s. These sequences are spread across a wide range of frequencies. Thus, if the signal is distorted by some process like noise or filtering that damages only certain bands of frequencies the message will still be recoverable.

Spread spectrum communications have two characteristics that are important to watermarking. First, the signal energy inserted into any *one* frequency is too small to create a visible artifact. Second, the watermark is dispersed over a large number of frequencies, so that it becomes robust against many common signal distortions. More information about the characteristics of spread spectrum techniques is available in the Appendix.

3.1.2. What to hide?

In most of the systems proposed so far, the watermark consists of a pseudo-random sequence of independent and identically distributed samples. Such a form is the result of the approach that we called “first model” at the beginning of this chapter. This model approaches the watermarking problem as transmission of a weak signal over a very noisy channel, a problem that is commonly handled by spread spectrum techniques. The pseudo-random sequence is initiated by a secret key to achieve system security, and all elements in the sequence can be regenerated any time the key is known.

This key-dependant random sequence can be used as the watermark itself, as is the case for 1-bit watermarking, or it can be modulated by another series of bits which convey the information (possibly in a coded way) to be communicated. In the former case, the decoder is only asked to decide upon the watermark presence.

In some applications it may be suitable to have a watermark which corresponds to another image, possibly a logo or a serial number. When the watermark is extracted from the attacked image, it is not required to have 100% of the bits matching because sophisticated pattern-recognition capabilities of human eye and brain may still detect the logo even if it is distorted. An example to logo watermarking is given in Section 3.2.3.

3.1.2.1. Basic Scheme for 1-bit Watermarking

The most straightforward way to add a watermark to an image in the spatial domain is to add a pseudorandom noise pattern to the luminance values of its pixels. There are many methods [55-69], [94] developed on this principle. In general the noise pattern consists of the integers randomly selected from {-1, 0, 1}. The pattern is generated based on a key, which is generally the seed of the random number generator. The only constraints are [71] that the energy in the pattern is more or less uniformly distributed and that the pattern is not correlated with the host image content. The formulation for this embedding method is given in Eqn 3.3.

$$\mathbf{I}_w(x,y) = \mathbf{I}(x,y) + k.\mathbf{w}(x,y) \qquad (3.3)$$

To create the watermarked image pixels $\mathbf{I}_w$(x,y), we multiply the pseudorandom pattern by a scalar gain factor (watermark strength parameter) k ($0<k<1$) and add to the host image pixels.

We may extend the idea to transform domain also. That is, we add the modulated pseudorandom pattern to the transform coefficients, **F**, instead of pixel values, **I**. We can do even a better job by making the watermark content-dependent (Eqn 3.4). Such a watermark in transform domain will be more robust since it hides itself in the strong components of the representation, which are harder to modify. We will look into different embedding formulas later.

$$\left|\mathbf{F}_w(x,y)\right| = \left|\mathbf{F}(x,y)\right| \cdot \left(1 + k.\mathbf{w}(x,y)\right) \qquad (3.4)$$

3.1.2.2. Multiple-Bit Techniques

This basic technique stores only 1-bit of information into the image. During the detection either the watermark is detected (logic-1 output) or it is not (logic-0 output). There are several ways to increase the *payload* of this basic technique.

The simplest way to embed a string of bits $b_1b_2\ldots b_L$ in an image is to divide the image into L subimages $\mathbf{I}_1\mathbf{I}_2\ldots\mathbf{I}_L$ of size mxn and to add a random watermark pattern (of the same size) to each subimage $\mathbf{I}_i$ (Figure 3.6) after modulating the pattern according to the corresponding bit value b_i [62], [60], [65]. The bits may modulate the patterns in several ways. We may add the random[6] pattern of size mxn to the subimage if the watermark bit equals one and leave the subimage unaffected if corresponding bit is zero (or -1).

[6] Since computers can only generate pseudorandom numbers we practically mean 'pseudorandom' when we use the term 'random'.

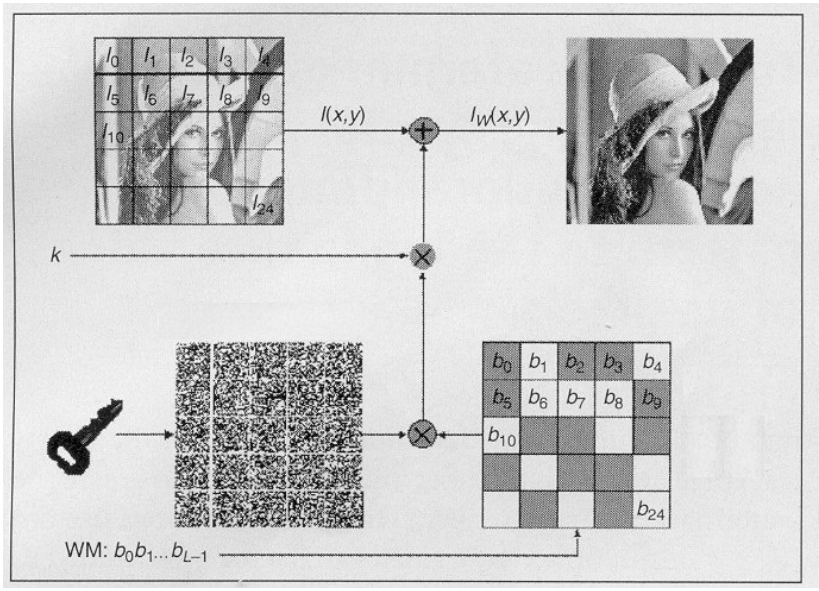


Figure 3.6 Embedding multiple (in this case 25) bits to an image [71].

Using a form of Direct Sequence Code Division Multiple Access (DS-CDMA) spread spectrum communications, we can also achieve multiple bits embedding [93]. In this technique, we generate a separate pseudorandom pattern of {-1, 1} for each bit of the message to be embedded. That is, if our message is $b_1b_2...b_L$, then we have L stochastically independent pseudorandom random patterns that have the same size as the image, call them $\mathbf{v}_1\mathbf{v}_2\ldots\mathbf{v}_L$. Each pattern, $\mathbf{v}_i$, is also modulated by its corresponding bit, b_i. We use the pattern $+\mathbf{v}_i$ if b_i represents a 0 and $-\mathbf{v}_i$ if b_i represents a 1. The summation of all random patterns $\pm\mathbf{v}_i$ constructs the watermark. Figure 3.7 shows a 1-dimensional example of this technique to generate a 7-bit watermark. In 2-dimension the signal and watermark vectors are possibly replaced by the $m\times n$ blocks of host image and random –1s and 1s respectively.

We can also scale down this summation before embedding to fit it within certain limits using Eqn 3.5 below.

$$\mathbf{w} = k.\left(\sum_{j=1}^{L}\mathbf{v}_j\right) \tag{3.5}$$

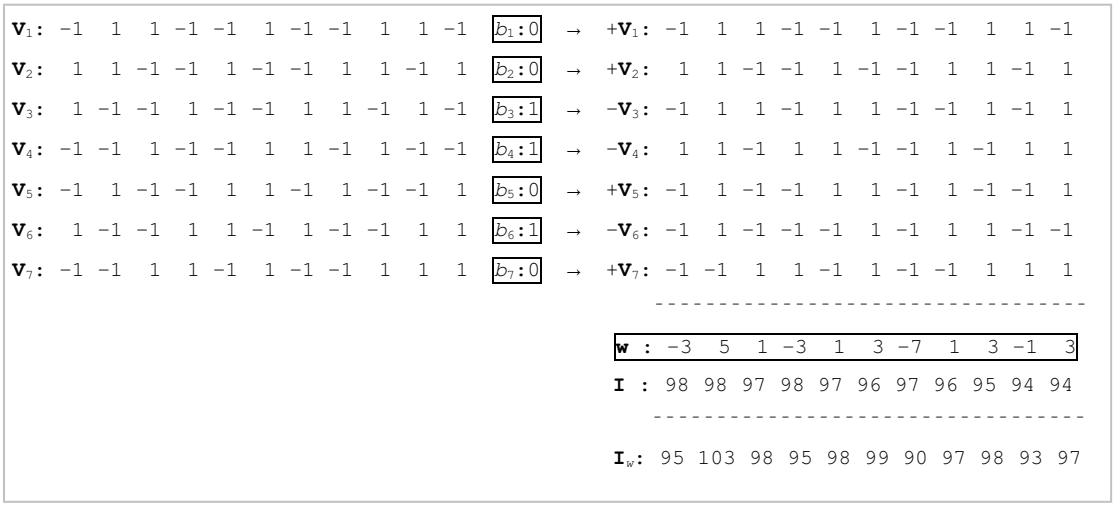


Figure 3.7 Generation of a 7-bit DS-CDMA watermark [71].

This idea of embedding multiple bits can also be extended to any transform domain. In that case, instead of working with blocks of image pixels one applies the algorithm on blocks of transform coefficients.

3.1.2.3 Redundant Embedding

When a work is distorted, it is unlikely that all coefficients (or pixels) in its representation are affected equally. We can give cropping as an example in spatial domain and filtering as an example in transform domain. Such a filter will only affect the coefficients outside its passband while changing almost all pixel values.

One general strategy for surviving such "local" distortions is to embed the watermark redundantly over several coefficients. If some of these coefficients fail to carry the watermark (get too much distorted), the remaining coefficients may still be sufficient to detect the watermark.

In spatial domain, "tiling" is a way of redundant embedding that provides robustness against cropping. Tiling can also be used as an alternative option in DS-CDMA technique described in Section 3.1.4.2. One can apply tiling in DS-CDMA by using the same random patterns for all blocks of the original image.

In general, we can say that a watermark is embedded redundantly, in some domain, if the watermark can be detected in several subsets of coefficients.

3.1.2.4. Anticipating Lossy Compression and Filtering

Not all watermarks that have been embedded in an image in spatial domain can survive the lossy JPEG compression since they usually consist of low-power, high-frequency noise which is coarsely quantized by the JPEG algorithm. These watermarks can also be affected severely by low-pass operations like linear or median filtering.

The robustness to JPEG compression can be improved by anticipating the losses beforehand thereby designing the watermark accordingly. In [62] such an idea has been mentioned. The watermark pattern to be embedded, W, is first compressed using the JPEG algorithm. The energy of the resulting pattern is increased to compensate for the energy lost in the high frequencies through the compression. Finally, this pattern is added to the image to generate the watermarked image. The idea is to filter out in advance all the energy from the watermark that would anyway be lost when the watermarked image is compressed. It is claimed in [62] that a watermark formed in this way is invariant to further JPEG compression of the same or higher quality factor.

In another blockwise embedding method [65] a different gain factor, k, is calculated for each 32×32 block. To adjust the gain, each block is tested for successful detection under a given JPEG quality factor which should be relatively lower than the factor to which the final watermark is required to be robust. If the watermark cannot be recovered from a block then the gain is increased iteratively.

3.1.2.5. Anticipating Geometrical Distortions

In a watermarking system it is really the embedding side that must be “strong” and “wise”. The success in correctly deciding on a watermark existence greatly depends on how elegant the embedding is. The main reason is that, the number of available tools is more at the embedding side and most of the time there is not much to do at

the detecting side other than correlation.

Geometrical transformations are the most challenging distortions for robust watermarking techniques. They hardly affect the image quality but they make most of the watermarks undetectable, especially if the watermark is not designed to survive geometric transformations. Most common ones are shifting, rotation, scaling, cropping and combinations of these four. Since they typically affect the synchronization between the pseudorandom pattern and the watermarked image, the synchronization must be retrieved before the detector performs correlation.

The main approaches to handle global geometric distortions are to recover the synchronization (undoing the distortion) or finding invariant host features that can be modified like the DFT amplitude which is invariant to shifting.

The techniques following first approach suggest placing markers (also called grid or template [24]) inside the image that tells us the original orientation of the image before the distortion. In [23] such a grid is designed using the sinusoidal signals which appear as peaks in FFT at certain frequencies. Since the original locations of these peaks are known the distortion can be estimated and inverted before detection.

For the second approach Fourier-Mellin Transform (FMT) has been suggested in [75]. Although theoretically perfect, it could not be implemented successfully in practice. FMT is defined as DFT followed by log-polar mapping (LPM). First, a DFT provides shift invariance and then LPM operation converts rotation and scale into translation along the horizontal and vertical axis. Finally, after a second DFT operating on the LPM result we obtain a rotation, scale, and translation invariant domain in which we can embed, for example, a CDMA watermark. For the successful operation of this technique the embedding scheme is modified to include templates to determine the scale and orientation of the watermarked image.

3.1.3 How to hide?

3.1.3.1 Additive and Multiplicative Embedding

The simplest approach to watermark embedding is the *additive* one which calculates the watermarked feature using the following formula

$$y_i = x_i + k m_i \tag{3.6}$$

where x_i is the i^{th} component of the selected feature set, m_i is the i^{th} element of the watermark vector, and k is the parameter controlling the embedding strength, or in other words, watermark power. Although not stated explicitly, this parameter can be used as an adaptive one, changing its value according to the mask the embedder uses to conceal the watermark.

Another common formula is the *multiplicative* formula stated as:

$$y_i = x_i + k m_i x_i \text{ or } y_i = x_i(1 + k m_i). \tag{3.7}$$

Additive formula has been especially popular among spatial domain watermarking algorithms. In this case, the embedding strength is adjusted according to the local characteristics of the host image. The advantage of additive watermarking comes under the condition or assumption that the selected host features are Gaussian distributed and attacks are limited to the addition of white Gaussian noise. Under these conditions, a correlation based detector operates at optimum, that is, either the overall error probability or the probability of missing the watermark (false negative) can be minimized.

The multiplicative formula is generally favoured by the techniques operating in the full-frame DCT or DFT. The main reason for such a choice is to utilize the masking effect of the significant frequency components in the image spectrum. It is known, in fact, that it is more difficult to perceive a disturbance at a given frequency if the image already contains such a frequency component. In other words,

embedding a watermark whose energy is proportional to the energy of the cover image at that frequency helps us to achieve invisibility without sacrificing watermark strength.

Another advantage of multiplicative embedding is its security against averaging. Since the watermark is image dependent it is more difficult to estimate the watermark by averaging a set of watermarked images.

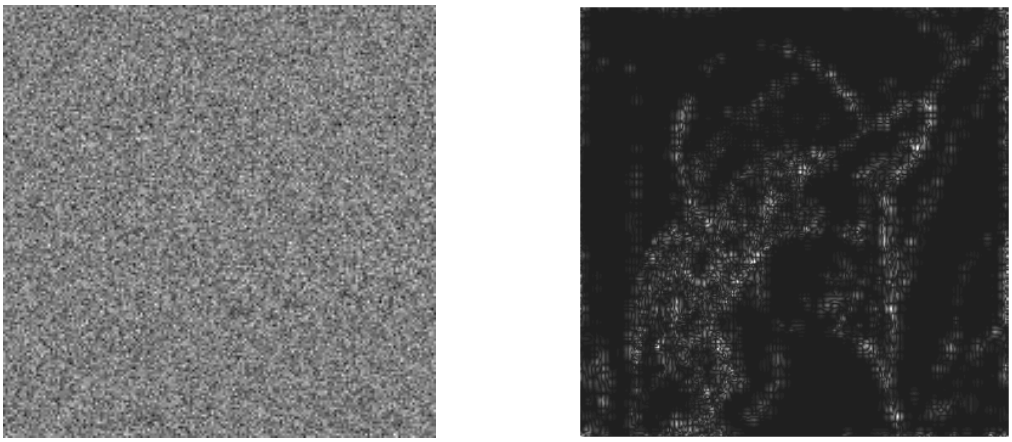


Figure 3.8 Difference images between original and watermarked Lena images for additive (left) and multiplicative (right) watermark embedding.

In Figure 3.8 it is easier to see how multiplicative embedding differs from additive. In the additive case all image pixels are affected from the modification in a random manner whereas watermark embedded using a multiplicative formula over DWT coefficients leaves the smooth regions of the image almost unaffected but emphasizes the watermark power at textured areas (feathers on the hat) and edges (contours of the hat).

There is also a third category of watermarking techniques which use neither additive nor multiplicative formulas but impose a different non-linear relationship on the modified coefficients or pixels. The majority of these types of algorithms can be classified as *quantization-based algorithms*.

3.1.3.2 Quantization-based Algorithms (Substitution Embedding)

The process of mapping a large –possibly infinite– set of values to a much smaller set is called quantization.

A quantizer system has two mappings: an encoder mapping and a decoder mapping. The encoder divides the range of possible input values into a number finite number of intervals (bins) and all values that fall into the same interval are represented by the same symbol (codeword). Since there could be (infinitely) many distinct samples in each interval, the encoder mapping is irreversible. The decoder generates a reconstruction value for each symbol generated at the encoder.

To use those encoder and decoder mappings to hide some information, the features to be watermarked are modified (quantized to some value) according to the watermark bit. Figure 3.9 shows an example to this type of watermarking in which the image pixel is mapped to the nearest reconstruction point of one of the quantizers according to the message bit to be embedded. The X's and O's represent reconstruction points of two separate quantizers each corresponding to one of the bits (0 or 1). This process is called *quantization index modulation* [119].

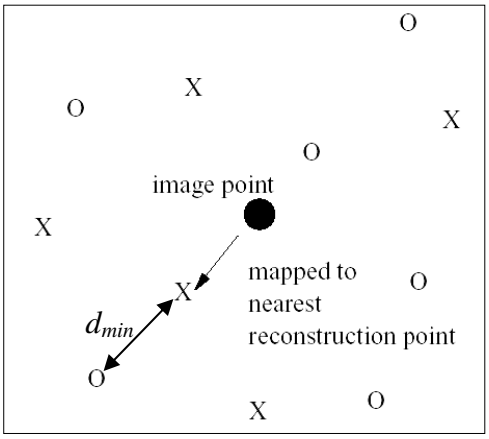


Figure 3.9 Quantization index modulation. Mapping the image pixel to the nearest reconstruction point according to the message bit $m \in \{0,1\}$.

The minimum distance $d_{\min}$ between sets of reconstruction points of different quantizers determines the robustness of the embedding. $d_{\min}/2$ is, intuitively the maximum amount of noise that can be tolerated by the system. That is, if the image point quantized to a X-point is subjected to some amount of noise such that its value shifts toward the O-point more than $d_{\min}/2$, then the decoding quantizer will decide that O-point is the value decided by the encoder and the watermark bit will be incorrectly detected.

3.1.4 Perceptual Adaptation of Watermarks

In the beginning of our discussion we stated that a watermarking system and a communication system has very similar parts and one can be mapped to the other in terms of the components involved in the overall process. One of the similarities is that in both systems the medium imposes some constraints on the signal carrying the information to be transferred through it. In the communications case, we usually have maximum average or peak power constraints associated with physical limitations of transmission devices. In a watermarking system, we have perceptual constraints related to the fact that the watermarked signal should be perceptually indistinguishable from the original.

Watermarks are supposed to be imperceptible. This requirement raises two important questions. How do we measure a watermark's perceptibility? And how can we embed watermarks in a host signal such that it cannot be perceived. Several researchers have investigated the human visual system (HVS) to answer these questions [121, 123, 127, 128, 100].

3.1.4.1 Perceptual Evaluation and Models

In the watermarking literature, most of the imperceptibility claims are based on automated evaluation criteria such as signal-to-noise ratio (SNR) and mean squared error (MSE) or on a single observer's judgements on a small number of trials. These empirical methods are not sufficient for proper comparison of watermarking

algorithms. Experiments using real human observers with a large number of subjects and a large number of trials can only give a true assessment of invisibility of a specific watermark. Although these experiments provide very accurate information they are very expensive to conduct and are not easily repeated. Therefore the algorithmic quality measures are frequently used.

The simplest metric or distance function that gives a measure of the distance between the original and watermarked image is the mean squared error which is defined in Eqn 3.8 as:

$$\frac{1}{N}\sum\left(\mathbf{I}_w[i]-\mathbf{I}[i]\right)^2 \qquad (3.8)$$

Although peak SNR and MSE are often used to find a quick and rough measure of watermark's fidelity impact, they actually provide a poor estimate of the true fidelity and can be misleading [122]. The three images in Figure 3.10 show such an example. The image on the left is a 1 pixel shifted version of the 256×256 Lena image (middle) and on the right is the same image with additive Gaussian noise of high amplitude. Clearly, a human observer will find the image on the right more similar to original. However, the MSE measure does not say so. The calculated mean square error for shifted version is 266.6469 while for the noisy version it is 255.4736.

Figure 3.10 (Left) Lena image with additive Gaussian noise (MSE = 255.4736); (Middle) Original Lena image; (Right) Lena image shifted 1 pixel to the right (MSE = 266.6469).

Different models are available in the literature and Watson model [47] is one of these built for measuring visual fidelity. Watson model consists of a sensitivity function, two masking components based on luminance and contrast masking, and a pooling component. Although this model is far better than MSE at estimating the perceptual effects of noise added to images, it still relies on proper synchronization and can overestimate the effects of shifts. Anyway, there are watermarking methods that utilizes those perceptual models [37].

One of the most important reasons that the error sensitivity based methods cannot work effectively is that they treat any kind of image degradation as certain types of *errors*. However, human eye is highly adapted to extract structural information from the viewing field, not pixel-wise differences and large errors do not always mean large structural distortions. Therefore, a measurement of structural distortion should be a good approximation of perceived image distortion. Based on this philosophy, Wang and Bovik proposed another metric called *image quality index* (Q) [125, 126, 97].

Figure 3.11 shows a set of experimental results. For all 5 of the test images (b)-(f) the MSE value is very similar. However, they are perceptually much different from each other as anyone might agree. We see that quality index calculation results in a more successful perceptual evaluation.

To find the quality index (Eqn 3.9), first, the original (subscript-x) and test (subscript-y) images are subjected to a 8×8 sliding window and for each position of the window the formula below is calculated, where bars over letters designate average and σ stands for the variance of the pixel values within the window.

The sliding window calculations results in a quality map of the image where the dynamic range of the map is [-1, 1]. The best value is achieved if and only if $y_i = x_i$ for all i. The overall quality index value is the average of the quality map. The formulation can be stated as:

$$Q = \frac{\sigma_{xy}}{\sigma_x \sigma_y} \cdot \frac{2\overline{x}\,\overline{y}}{\overline{x}^2 + \overline{y}^2} \cdot \frac{2\sigma_x \sigma_y}{\sigma_x^2 + \sigma_y^2} \qquad (3.9)$$

Perceptual models are used not only to measure the perceptual impact of a watermark but also to control it during the watermark embedding process. Most watermarking systems attempt to shape the added pattern according to some perceptual model to achieve automatic adjustment of the embedding strength and obtain a desired perceptual distance.

Figure 3.11 Original 512×512 Lena image (a) and 5 different distorted versions: (b) contrast stretched, MSE=255 Q=0.9372 (c) Gaussian noise contaminated, MSE=255 Q=0.3891 (d) impulsive noise contaminated, MSE=255 Q=0.6494 (e) blurred image, MSE=255 Q=0.3461 (f) JPEG compressed, MSE=215 Q=0.2876.

3.1.4.2 Perceptually Shaping the Watermark

Perceptually shaping the watermark by exploiting the basic characteristics of the HVS, the watermark energy can be increased thereby improving its robustness. The perceptual knowledge may either implicitly (intrinsic in the transform domain) or

explicitly be applied in the embedding process. The watermark is locally scaled, attenuating some areas and amplifying others, so that the watermark is better hidden by the host image.

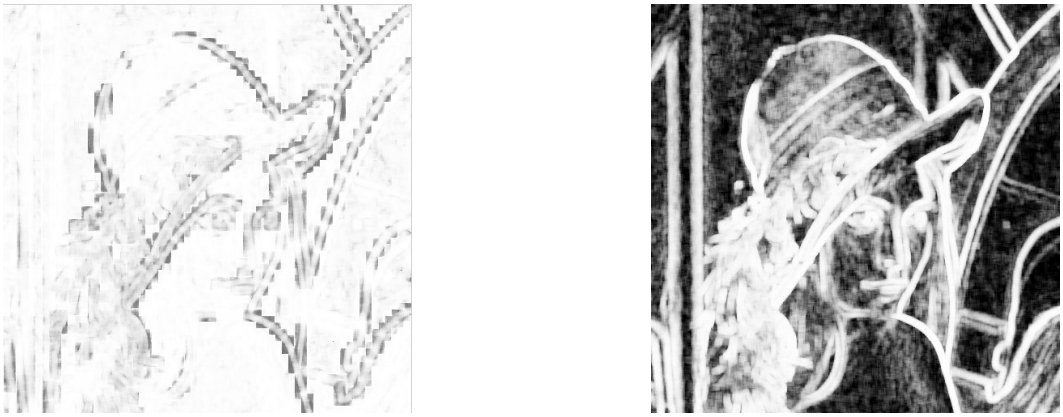


Figure 3.12 On the left is a quality map image of a highly watermarked image calculated using Wang and Bovik [125] method. The watermarking method is the spatial logo embedding technique with a contrast mask based on local image variance applied. The map successfully reveals the modified high variance regions. When there is no mask applied (right) the watermark effects the image almost in a reverse manner.

Watermark shaping is generally achieved by the masks involved in embedding formulas (Eqn 3.10). The masks are multiplied with the watermark pattern before it is added to the image representation (in a spatial or transform domain).

$$\mathbf{f}_w(x,y) = \mathbf{f}(x,y) + Msk(x,y) \cdot k \cdot \mathbf{w}(x,y) \qquad (3.10)$$

A simple perceptual mask can be constructed in the spatial domain by exploiting the local variance characteristics of an image. Lets consider the spatial watermarking method we used in Section 3.1.4 to embed logos into the image. If we adjust the random pattern amplitude according to the variance of each block we make an HVS-aware embedding. This is clearly seen in Figure 3.12. The quality map on

the left identifies the modified regions of a watermarked image as low-fidelity (dark regions). The smooth areas of the image are not affected at all while textured areas and edges received considerable modification. This was expected since the mask attenuates the watermark power over smooth areas while amplifying it for textured regions and the map shows us that these areas are different then the original.

If we do not apply the local variance mask, then the quality map (right) is almost an inverse of the first one. This time, smooth areas show low-fidelity while textured regions are found as very close to original. This is also expected. The quality map actually behaves like our eye. The more or less equally distributed watermark power hardly creates any perceivable distortion in textured regions, since they are high fidelity. However, same watermark power becomes more effective over smooth regions and the given quality calculation method finds out that these areas are low fidelity.

There are other ways of obtaining a spatial domain mask. In [36] the masking image is generated by filtering the image with a Laplacian high-pass filter and by taking the absolute values of the resultant filtered image. One can even use the output of a simple Prewitt edge detector as a spatial mask.

Selecting the features or the transform space for watermark embedding is often based on perceptual knowledge and choosing a space where we can differentiate perceptually significant and insignificant components of the original host signal. We have mentioned that DCT is widely studied for its perceptual characteristics. There are masks defined in terms of DCT coefficients like in [70], where the squared sum of the 8×8 DCT AC coefficients is used to generate a masking image. Experiments show that [71] a perceptually invisible watermark modulated with a gain factor locally adapted to such a mask can contain twice as much energy as a watermark modulated with a fixed gain factor.

Masking and watermark embedding domains can be different. For example, a spatial mask can be defined even if the watermark is going to be embedded in another domain like DFT, DCT or DWT. In this case, the nonspatial watermark is first embedded in the image $\mathbf{I}$, resulting in a temporary image $\mathbf{I}_{wt}$. The watermarked

image $\mathbf{I}_w$ is now constructed by mixing the original image and this temporary image by means of a masking image *Msk* [106, 103] as in Eqn 3.11. Here, the masking image must be scaled to values in the range from zero to one.

$$\mathbf{I}_w(x,y) = (1 - Msk(x,y)) \cdot \mathbf{I}(x,y) + Msk(x,y) \cdot \mathbf{I}_{wt}(x,y) \quad (3.11)$$

3.2. Watermark Detection

The goal in a watermarking system is basically introducing some information into a medium and then trying to extract it as reliably as possible. If we think of a watermark embedder as the transmitter in a communication chain then a watermark extractor will be the receiver.

In Figure 3.13 we give a somewhat detailed block diagram of watermark detection process. The detection may serve two different purposes: deciding whether the image under test contains a watermark (if it is watermarked by a 1-bit scheme), and extracting a message that the watermark might carry (if it is watermarked by a multi-bit scheme).

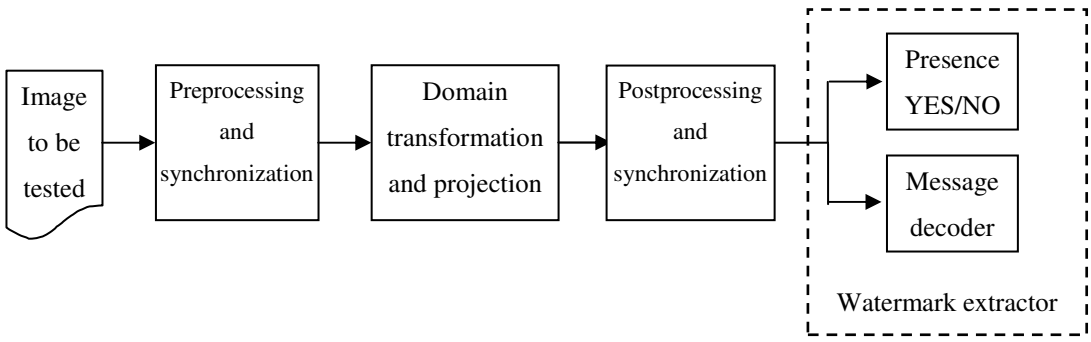


Figure 3.13 Diagram of a generic watermark detection process [41].

3.2.1. Detecting 1-bit Watermarks

We have started analyzing the watermark embedding process with a very basic scheme which could be used to embed a single bit of information into an image in

spatial domain. For the detection of that watermark we calculate the correlation between the received image (test image, suspected to be watermarked) and the embedded (reference) pattern (pseudorandom noise). The randomness of the reference pattern plays a crucial role here, because pseudorandom sequences generated with different seeds have very low cross-correlation values (Figure 3.14).

The ownership information lies in the key which is generally the seed of the random number generator. If the correct key is not known, then the correlation values would be much lower than the values with a correct key. Hence, a high correlation value, which means the watermark is detected, occurs only when the watermarked image is correlated with a pattern with the correct seed (Figure 3.15).

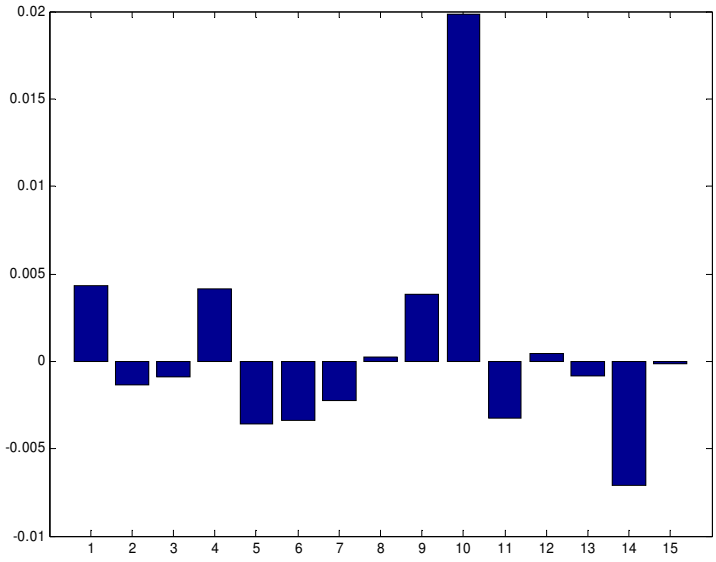


Figure 3.14 There are fifteen superimposed plots on the graph above. They are the correlation of 14 1D normally distributed random sequences (generated with separate seeds) with a reference sequence (generated with seed 10), each of length 100. The positive peak in the middle belongs to the autocorrelation trace of the reference sequence.

If we had a high correlation between two patterns generated from different seeds, then we would have a high false positive decision rate, which is an undesired

merit for a watermarking system. As an experiment, we watermark the 256x256 Lena image by simply adding a uniformly distributed pseudorandom {-1, 1} pattern with seed 10. Then we calculated correlation coefficients between the watermarked image and 15 different random patterns generated with seeds 1 to 15.

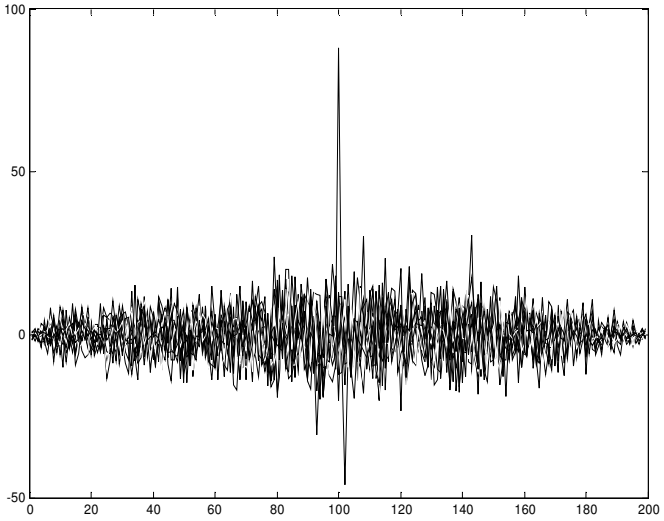


Figure 3.15 The correlation coefficients between the watermarked Lena image and 15 different random patterns generated with separate seeds. 10th pattern is the embedded pattern, so it is detected as a peak on the correlation coefficients.

The correlation coefficient corresponding to the pattern of seed 10 is largest among the other fourteen (Figure 3.15). This experiment shows that the image and the random patterns are weakly correlated. This is also a requirement since we want the correlation value, which we use to decide about watermark existence, to depend only on the random pattern added to the image.

To be convinced that the watermark is detected with the correct key, the recovered (possibly distorted) watermark pattern is correlated with original, undistorted random patterns having various (generally around 1000) different seeds/keys and it is required that one pattern with correct seed (having the same seed as the embedded random pattern) results in a high correlation value.

The majority of the systems proposed in the watermarking literature, exploit a correlation-based detection measure for detecting the watermark. There are several forms of correlation used in watermark detection. The most basic one is the *linear correlation*. Other common types are *normalized correlation* and *correlation coefficient*[7].

In some systems the correlation value between the suspected and reference patterns are computed explicitly, while some methods define ways of detection sounding different but that are mathematically equivalent to correlation.

Whether the watermark is detected or not is not generally based on a human observer's decision on the availability of a peak in correlation plots. It is common to set a threshold, T, on the detection value. If it exceeds the threshold the watermark detector automatically decides that the test image contains the watermarked. In this case, the selection of the threshold plays an important role on the detector performance. Because a suboptimal threshold may cause a high percentage of false negatives (P_{fn}) or false positives (P_{fp}).

Kim and Moon [33] uses a threshold (S) calculation formula (Eqn 4.13) calculated from the embedding strength (α) and the mean of the amplitude of the watermarked coefficients (V_i') which are in the wavelet domain (M is the number of the watermarked coefficients):

$$S = \frac{\alpha}{2M} \sum_i \left| V_i' \right|.$$

3.2.2. Detecting Multiple-bit Watermarks

We embedded multiple bits by dividing the image into several blocks (subimages) and making the embedding into each block, say $\mathbf{I}_j$ under control of a single bit b_j. For the detection, the detector again calculates the correlation between the subimage and corresponding random pattern. It assigns the value 1 to the

[7] See Appendix A.3 for a brief description of each

constructed watermark bit if the correlation exceeds a certain threshold T, otherwise the watermark bit is assumed to be zero.

To avoid using a threshold, we can add two different random patterns P_0 and P_1 for watermark bit 0 and 1, respectively. This time detector correlates each subimage with both of the random patterns. Then the bit value corresponding to the pattern that gives the highest correlation with the watermarked image is decided as the received bit. We can use this method in a wiser way by selecting the patterns P_0 and P_1 such that they differ only in sign, that is $P_0 = -P_1$ [62]. Then in this case, the detector only has to calculate the correlation between the subimage and one of the patterns, say P_0. If the correlation is positive then bit is decided as 0, if it is negative then the received watermark bit is assigned to 1.

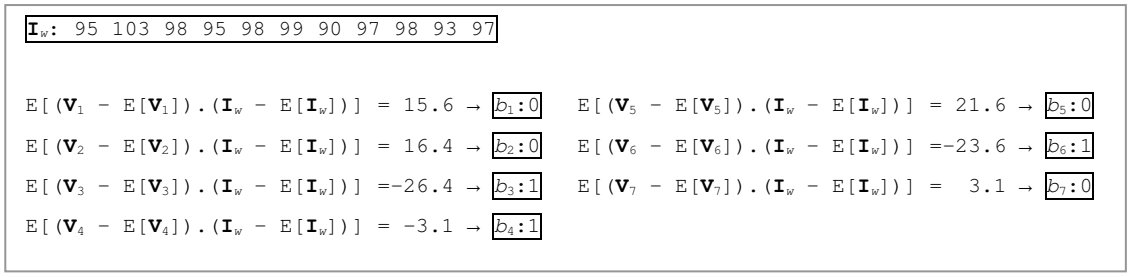


Figure 3.16 Extraction of the bits from the CDMA watermark.

Our other alternative for embedding multiple bits was to use the DS-CDMA technique. In this case, the bits b_i are extracted by calculating the correlation between the normalized (mean subtracted) image and the corresponding pseudorandom pattern $\mathbf{v}_i$. If the correlation is positive, the bit value is decided as 0, otherwise the watermark bit assumed to be one. In Figure 3.16 the detection scheme is illustrated again on the 1-dimensional example. All bits $b_1 \ldots b_7$ are correctly extracted.

This form of spread spectrum is resistant to cropping (provided that is synchronised), non-linear distortions of amplitude and additive noise. Also, since it has good statistical properties it can be mistaken for noise and go undetected by an eavesdropper.

Figure 3.17 shows the detection results for a logo embedded using the basic multi-bit embedding technique in spatial domain. The logo is a binary image of size 32×128 (4096 bits). When the watermarked image is JPEG-compressed with quality of 70 and 30 we can detect about 83% and 76% of the bits correctly, respectively. For a watermarking technique which decides on the existence of the watermark according to the number of correctly detected bits, these percentages might mean that the watermark is not detected. However, when the bits make up an image as in this case, the extracted bits make sense and we can say that the watermark is detected. In the 30% compression case, although it is heavily corrupted the logo can still be recognized. Although the example given makes the embedding in spatial domain, it is possible to do it in a transform domain like block-DCT.

Figure 3.17 Original logo (above left), extracted logo after JPEG-70 compression (above right), extracted logo after JPEG-30 compression (below).

3.2.4. Detection Errors

During the detection process, the watermark detector can make two types of errors. It can detect the existence of a watermark although there is none (false positive), or the detector can reject the existence of the watermark even though there is one (false negative).

In order to decrease the error probabilities P_{fn} and P_{fp}, the first thing to do is to increase the power (σ_W^2) in the watermark pattern, since for an image consisting of N pixels and detection threshold T those probabilities are (as calculated in [1]):

$$P_{fp} = \frac{1}{2} erfc\left(\frac{T\sqrt{N}}{\sigma_W \sigma_I \sqrt{2}} \right) \text{ and } P_{fn} = \frac{1}{2} erfc\left(\frac{(\sigma_W^2 - T)\sqrt{N}}{\sigma_W \sigma_I \sqrt{2}} \right)$$

where $erfc(x) = \frac{1}{\sqrt{2\pi}} \int_x^{\infty} e^{-t^2/2} dt$.

If the watermark pattern only consists of integers {-1, 1} with a uniform distribution, the errors P_{fn} and P_{fp} can be minimized by increasing the strength parameter *k*. However, it cannot be increased unboundedly since larger *k* decreases the quality of the watermarked image.

The probability function for the detection process is presented in Figure 3.18. In that figure, the peak on the left corresponds to the detection value distribution for non-watermarked images. It is seen that some non-watermarked images result in a detection values above threshold *T*, which means a watermarked is detected. Thus, resulting in a false positive. The peak on the right is the detection value distribution for watermarked images. In this case, for some watermarked images correlation gives a value less than the threshold, causing a false negative.

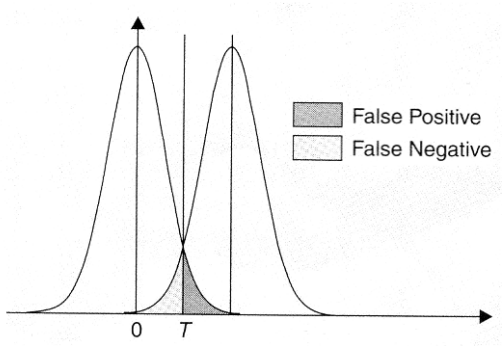


Figure 3.18 Errors in detection by correlation [71].

3.2.5. Improving Watermark Detection

In Section 3.1.4.2 and 3.2.3 we mentioned about using the pixels of a binary logo

image as the bit stream of the spatial domain multi-bit watermarking technique and in Section 3.2.1.1 it is claimed that increasing watermark power by increasing the strength parameter improves detection. Figure 3.19 shows a plot of the increase in the correctly recovered bits as the gain/strength parameter increases. The gain parameter k directly affects the embedding power since the embedded sequence is multiplied by it before being added to the host image.

However, increasing the power of the watermark results in a loss in the watermarked image quality. So, robustness is traded with fidelity (invisibility) as we expected. With k equal to 5 the image quality index between the watermarked and original image is 0.717, PSNR is 34.036 dB and MSE is 25.672 which are moderate figures and the resultant image is acceptable.

Another strategy to obtain a better detection value is to apply a matched filtering before correlation [39]. The filtering helps remove the interference between the host image content and the watermark especially in the low-frequency bands. The following convolution kernel F_{edge} can be used to filter out the low frequencies (Eqn 3.12).

$$F_{edge} = \frac{1}{2} \cdot \begin{bmatrix} -1 & -1 & -1 \\ -1 & 10 & -1 \\ -1 & -1 & -1 \end{bmatrix} \qquad (3.12)$$

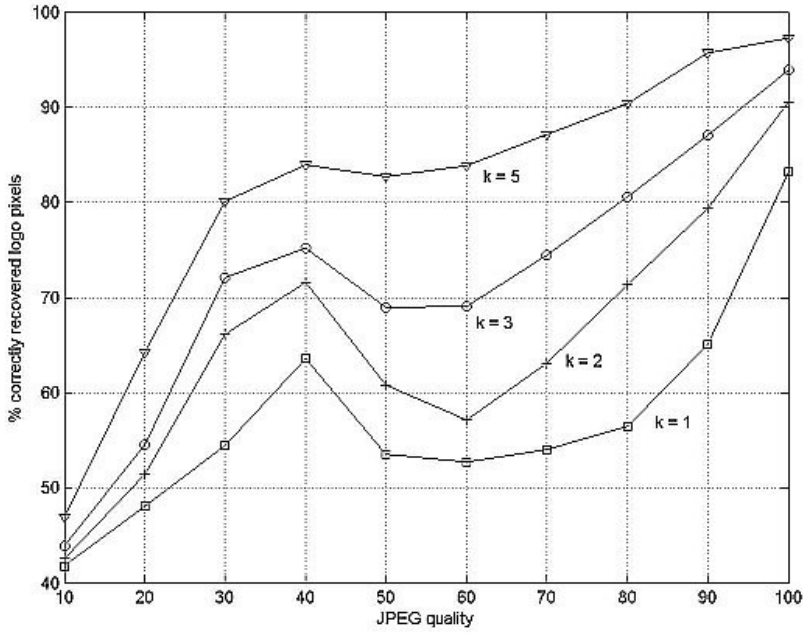


Figure 3.19 The percentage of correctly recovered pixels vs. JPEG quality for 4 different values of *k* parameter.

The experimental results also verify the positive effect of filtering as in Figure 3.20 which gives how the percentage of correctly recovered bits increases if the filter is applied to a watermarked image watermarked with strength *k* equal to 3.

The logo watermarking method that we borrowed from Section 3.1.4.2 embeds each logo pixel to a block of host image. In the experiments above, that block size is always 8×8 pixels. Hence, 64 pixels (512 bits) of host image are used to hide a single bit of information (logo pixel). We can increase the probability of detecting each information bit correctly by embedding it in larger blocks, that is, by decreasing the payload. This requires either a larger host image or a smaller logo image. Therefore, we verify that payload and robustness are also conflicting requirements. Figure 3.21 shows the results with a 16×16 logo, that is the subimage block size is 32×32 instead of 8×8. The increase in the robustness is significant.

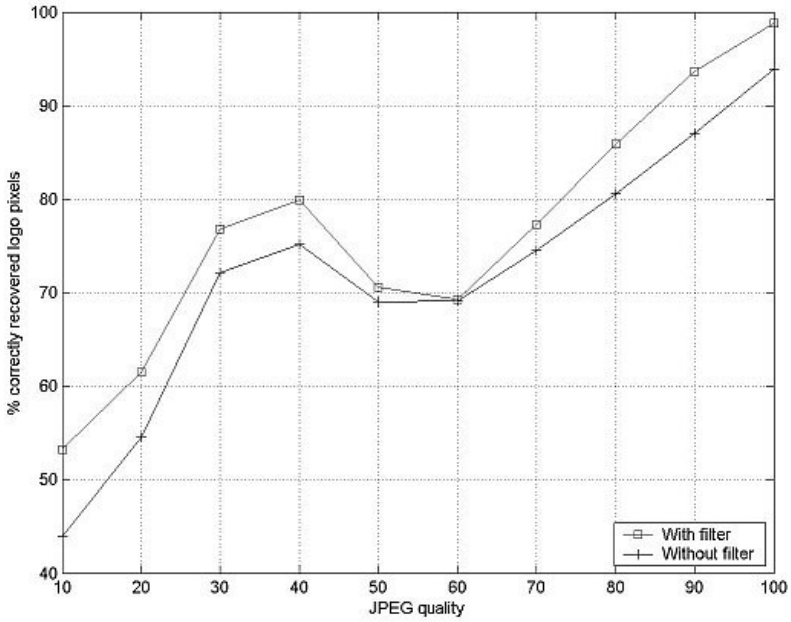
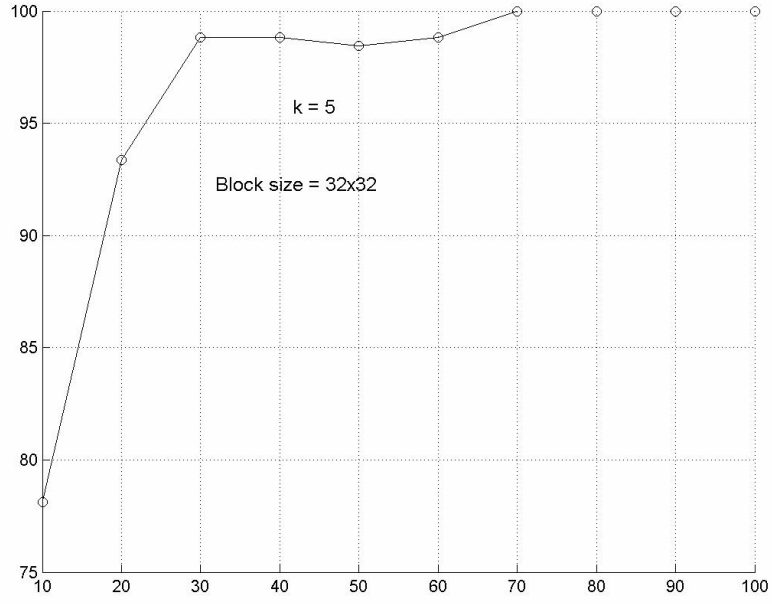


Figure 3.20 The effect of applying a high-pass filter before watermark detection (*k*=3).



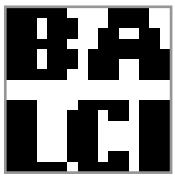


Figure 3.21 (Left) The increase in the robustness of the logo watermarking scheme by decreasing the payload. (Right) The 16x16 binary logo image, the payload.

3.3. Attacks on Digital Watermarks

In watermarking terminology, an *attack* is any processing that may impair detection of the watermark or communication of the information conveyed in the watermark. The processed watermarked data is then called *attacked data*.

Robustness is an important aspect of any watermarking scheme. Its notion is clear: A watermark is robust if it cannot be impaired without also rendering its carrier (cover, host) data useless. Hence, an attack can only be said to be successful to defeat watermarking scheme if it impairs the watermark beyond acceptable limits while maintaining the perceptual quality of the attacked data.

Since the complete theoretical analysis of the robustness of an algorithm against different attacks is too complicated researchers use the results of experiments commenced by some benchmarking tools like Stirmark [96] and Unzign [35].

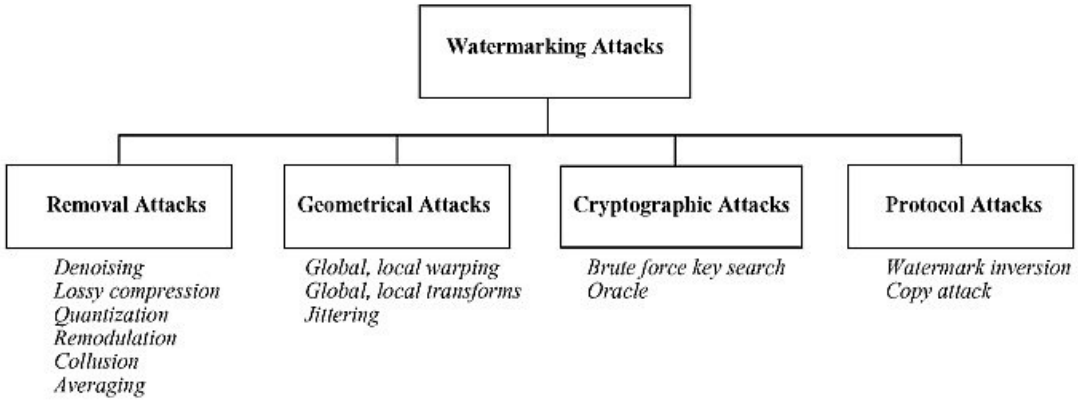


Figure 3.22 Classification of watermark attacks [91].

The literature of watermarking attacks is also huge. The wide class of existing attacks can be divided into four main categories: removal attacks, geometrical attacks, cryptographic attacks and protocol attacks. Figure 3.22 summarizes the different attacks [91]. We will mention about them briefly.

3.3.1. Removal Attacks

Removal attacks aim at complete removal of a watermark from the cover data. Denoising stems from the idea that watermark can be treated as a noise of some statistical properties. Therefore, it can be estimated from the available copy of watermarked data. Image denoising is mostly based on maximum likelihood (ML), maximum aposteriori probability (MAP) or minimum mean square error (MMSE) criteria. Lossy compression has been shown to have roughly the same influence on noise removal as denoising.

A relatively new method called remodulation attack predicts the watermark via subtraction of the median filtered version of stego (watermarked) image from the stego image itself. It is additionally high-pass filtered, truncated and then subtracted from the stego image with a constant amplification factor of 2. However, this scheme performs well only for high-pass watermarks. When the watermark spectrum is well-matched with the host image the attack shows poor performance. Hence, here we see the importance of perceptual adaptation methods since they provide a way of invisibly embedding the watermark into the lower frequency bands of the image.

Other attacks in this group are statistical averaging and collusion attacks. Many instances of a given data , each time signed with a different key or different watermark are averaged to compute the attacked data. If the number of available data sets is large enough the embedded watermark may not be detected anymore assuming that on average it will yield zero mean. With the collusion attack, many instances of the same data are available, but this time the attacked data is generated by taking only a small part of each data set and rebuilding a new attacked data set. Some countermeasures for the video watermarking application are suggested in [25] by Deguillaume.

3.3.2. Geometrical Attacks

In contrast to the removal attacks, geometrical attacks intend not to remove the embedded watermark itself, but to distort it through spatial or temporal alterations of

the stego data. The attacks are usually such that the watermark detector loses synchronization with the embedded information.

The most well-known integrated software versions of these attacks are Unzign and Stirmark. Unzign introduces local pixel jittering and is very efficient in attacking spatial domain watermarking schemes. Stirmark introduces both global geometrical and local distortions. The global distortions are rotation, scaling, change of aspect ratio, translation and shearing that belong to a class of general affine transformations. The line/column removal and cropping/translation are also integrated in Stirmark.

Although robustness to global affine transformations is more or less a solved issue, the local random alterations integrated in Stirmark still remains an open problem almost for all techniques. The so-called random bending attack exploits the fact that the human visual system is not sensitive against shifts and local affine modifications. Therefore, pixels are locally shifted, scaled and rotated without significant visual distortions.

Figure 3.23 shows how random bending, one of the toughest-to-survive geometrical attacks implemented by Stirmark, affects an image. In a natural image it is almost impossible to observe the effect. Hence, although you see no distortion at all the detector fails to detect the watermark.

3.3.3. Cryptographic and Protocol Attacks

Cryptographic attacks are brute force methods to find the secret information through exhaustive search. Since many watermarking schemes use a secret key it is very important to use keys with a secure length. The protocol attacks aim at general watermarking framework. The two examples given are watermark inversion introduced by Craver et al. [19], and the copy attack [26].

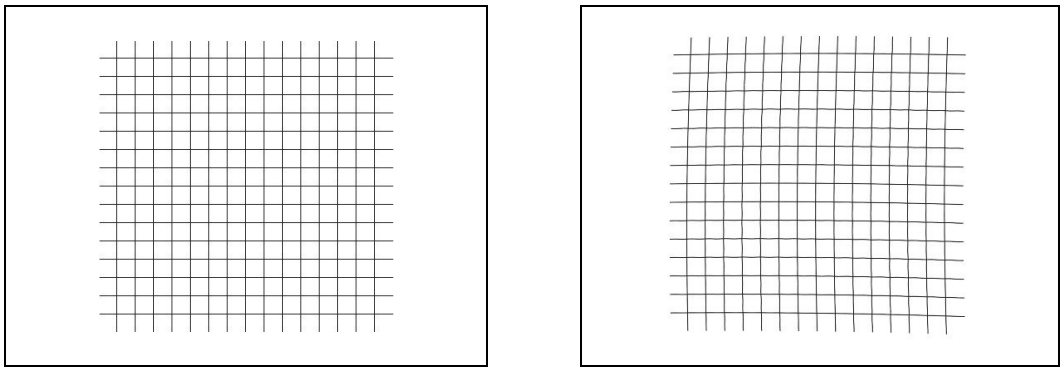


Figure 3.23 The effect of Stirmark random bending attack can be better seen on a grid.

To resist watermark inversion, the copyright protection algorithm watermarks need to be non-invertible. Otherwise, an attacker who has a copy of the stego data can claim that the data contains also the attacker's watermark by subtracting his own watermark, thereby creating an ambiguity. Non-invertibility requires that it should not be possible to extract a watermark from non-watermarked image. As a solution, it is proposed to make watermarks signal-dependant by using a one-way function.

CHAPTER 4

ALGORITHMS AND EXPERIMENTS

The field of digital watermarking has been very popular since it emerged and it is a rapidly developing branch of image processing with several conferences, workshops and more than 100 papers published each year. In this chapter we will look in more detail to 9 selected algorithms among this enormous diversity of approaches to the problem of watermarking and discuss their performance in terms of robustness against a set of attacks.

The selected algorithms are previously implemented by Peter Meerwald [119] as DOS-based executables. As a part of the thesis work we prepared a software that provides a graphical user interface to use these algorithms and also the Stirmark, so that one can do the whole watermarking process (embedding-attacks-detection) on a Windows panel instead of typing command line options. The results and created files, whose names can be entered into the program, are stored in folders specified by the user.

In general, watermarking algorithms can be classified according to many properties. Most common are their working domain (spatial or frequency), availability of the original host image for detection (blind or non-blind), human perception (visible or invisible), target document type (text, audio, image or video), and the purpose (copyright protection—robust, temper detection—fragile, data hiding, etc).

The algorithms that we will examine in the following sections differ from each other in their working domain (spatial, DCT or DWT), detection mechanism (blind or non-blind) and payload (single bit or multiple bits). This way, we will be covering most of the available watermarking methodologies, since these are the major categories the robust watermarking algorithms can be classified into.

We will give plots, figures about their robustness against a number of attacks implemented by Stirmark v3.1 [95, 96], IrfanView and Adobe Photoshop and try to end up with generalizations and comments on the relationship between their algorithmic properties and their performance (robustness).

4.1. Spatial Domain Technique (Bruyn)

The spatial domain technique described below is developed by V. Darmstaedter, J. F. Delaigle. J. J. Quisquater, B. Macq [94]. It is an improved version of the original method by Bruyndonckx, Quisquater, and Macq [3].

This is a multi-bit embedding scheme working in the spatial domain. The watermark is a bitstream carrying the copyright information.

Embedding method is block-based. Each bit of the watermark is spread over one 8×8 block of the spatial image. The main purpose of the choice is that 8×8 corresponds to the size of JPEG blocks. So the effects of JPEG compression affect each embedded bit independently.

For a 512×512 image this block size allows 4096 bits to be embedded. Since this capacity is more than sufficient for the copyright information, the information can be embedded redundantly using simple repetition coding or other codes like BCH with error-correction capability thereby increasing the robustness.

The embedding algorithm first classifies the pixels in each block into *zones* of homogeneous luminance values. Before deciding on the zones the block is first classified according to the one of the 3 contrast types (Figure 4.1) it belongs to.

1. Hard contrast
2. Progressive contrast
3. Noise contrast

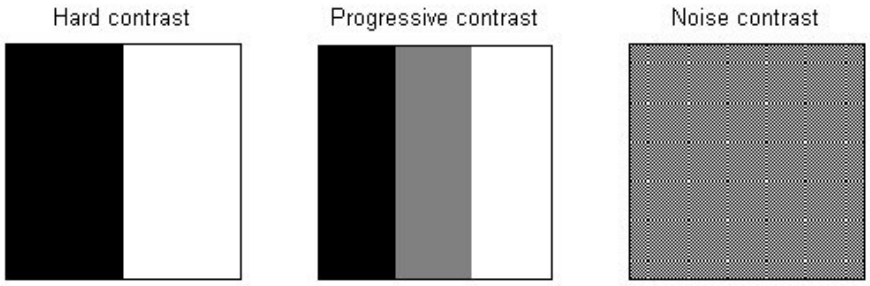


Figure 4.1 Example block images corresponding to each contrast type.

The contrast type of the block is decided according to the metrics computed from a slope function $S(x)$, which is the slope of $F(x)$, an increasing function of the luminance of the block pixels where $F(1)$ is the lowest luminance value in the block and $F(n^2)$ is the highest luminance value, n being the block size.

Let S_{max} be the maximum slope of F, located at index $x = \alpha$. If S_{max} is lower than a given threshold T_1 the block has a noise contrast, otherwise it has either progressive or hard contrast. In this case, two indexes β_+ and β_-, closest to α and respectively higher and lower satisfying $S(\alpha) - S(\beta_+) > T_2$ and $S(\alpha) - S(\beta_-) > T_2$, are defined where T_2 is another threshold value. The block is a hard contrast type if $\alpha = \beta_+ = \beta_-$ (there is a step in $F(x)$ at $x = \alpha$) and it is progressive otherwise.

After the contrast type is decided it is now possible to classify the pixels $p(i, j)$ into zones according to the following the rules below:

- For progressive and hard contrasts;

  if $p(i, j) < F(\beta_-)$, $p(i, j)$ belongs to zone-1,

  if $p(i, j) > F(\beta_+)$, $p(i, j)$ belongs to zone-2.

- For noise contrasts the pixels are separated in two groups having the same dimension;

  if $p(i, j) < F(n^2/2)$, $p(i, j)$ belongs to zone-1,

  if $p(i, j) > F(n^2/2)$, $p(i, j)$ belongs to zone-2.

The blocks are also divided into two *categories* (*A* and *B*) to "find a room" for embedding the watermark bit because a direct modification on the zones is reported to be neither robust nor acceptable for the invisibility. The subdivision is determined by a *grid* whose structure is defined before the embedding. Figure 4.2 shows two different grids as an example.

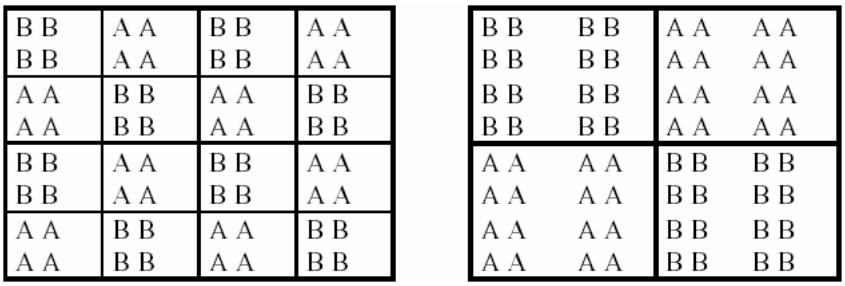


Figure 4.2 Category grids for an 8×8 block: 2×2 grid (left), 4×4 grid (right).

It is also noted that the grid that one uses to watermark the image must be kept secret. It is easier to remove the watermark once the knowledge of the grid is revealed. For increased security the grid may be changed for every block according to a secret key.

From the steps unto now, 4 different groups of pixels are defined depending on the zone (1 or 2) and the category (*A* or *B*), namely 1A, 1B, 2A, and 2B. The number of pixels in each group will be denoted by n_{1A}, n_{1B}, n_{2A}, and n_{2B}.

Next, 6 mean values are computed from these subdivisions of pixels: m_{1A}, m_{1B}, m_{2A}, m_{2B}, m_1, and m_2. The last two are the zone means, that is the means combined from the two categories in each zone. The embedding of a bit b in the block is performed through the relationship between categories luminance mean values. The embedding rule is given by 4 formulas (Eqns 4.1-4.4) and two constraint equations (Eqn 4.5 and 4.6). The constraints are required to conserve the mean value within a zone in order to make the embedding invisible to the eye.

° if $b=0$: $m_{1B}^{*}-m_{1A}^{*}=l$ (4.1)

$m_{2B}^{*}-m_{2A}^{*}=l$ (4.2)

° if $b=1$: $m_{1A}^{*}-m_{1B}^{*}=l$ (4.3)

$m_{2A}^{*}-m_{2B}^{*}=l$ (4.4)

$\left(n_{1A}\cdot m_{1A}^{*}+n_{1B}\cdot m_{1B}^{*}\right)/\left(n_{1A}+n_{1B}\right)=m_1$ (4.5)

$\left(n_{2A}\cdot m_{2A}^{*}+n_{2B}\cdot m_{2B}^{*}\right)/\left(n_{2A}+n_{2B}\right)=m_2$ (4.6)

where the quantities with a superscript asterisk denote the mean values after the pixels are modified and l is the embedding level.

To extract the embedded bits the size of the blocks (here we assume it is 8×8) and the grid(s) must be known. The pixels of the suspected image are divided into 8×8 blocks and the pixels in each block are classified into zones and categories just like the embedding procedure. Then the following values are computed:

$$\Sigma_1=m_{1A}-m_{1B} \text{ and } \Sigma_2=m_{2A}-m_{2B}$$

and the following cases appear:

1. $\Sigma_1\cdot\Sigma_2>0$: $b=1$ if $\Sigma_1>0$ and $b=0$ if $\Sigma_1<0$
2. $\Sigma_1\cdot\Sigma_2<0$: find $\Sigma=\left(n_{1A}+n_{1B}\right)\cdot\Sigma_1+\left(n_{2A}+n_{2B}\right)\cdot\Sigma_2$;
   $b=1$ if $\Sigma>0$ and $b=0$ if $\Sigma<0$,
   however result is uncertain
3. $\Sigma_1\cdot\Sigma_2\approx 0$: find $\Sigma=\max\left(\left|\Sigma_1\right|,\left|\Sigma_2\right|\right)$;
   $b=1$ if $\Sigma>0$ and $b=0$ if $\Sigma<0$,
   however result is uncertain.

4.2. DCT Domain Techniques

4.2.1. DCT Algorithm 1 (Cox)

The algorithm examined below is described by Cox, Kilian, Leighton, and Shamoon [102]. It is probably the most well-known paper on watermarking subject. Its reputation comes from the ideas suggested by it: the relation between spread spectrum communication and watermarking[8] and embedding the watermark to perceptually significant components of an image. These ideas are largely accepted and they affected almost all researchers after it is published.

The watermark is a Gaussian sequence of real numbers selected from $N(0,1)$. Alternative distributions like {1,-1}, {0,1} or [0,1] are also possible but authors state that such distributions leaves the image vulnerable to attacks using multiple watermarked documents.

n significant coefficients are altered according to Eqn 4.7. n is at the same time the length of the watermark $X = x_1, \cdots, x_n$ and v_i are the DCT coefficients except the DC coefficient.

$$v_i' = v_i(1 + \alpha_i x_i) \qquad (4.7)$$

α_i is a scaling parameter and it is coefficient adaptive since different spectral components may be more or less tolerant to modification. One way of determining the value of α_i is to create a degraded version $\mathbf{I}^*$ of the original image $\mathbf{I}$ and choose α_i to be proportional to the deviation in that degraded coefficient, namely $\delta_i = \left| v_i^* - v_i \right|$, or proportional to the average of δ_i's. However, the authors used a constant value $\alpha = 0.1$ for all their experiments in the paper.

Extraction of the watermark assumes that the original (cover) image is available at the detector. After the possibly corrupted watermarked image $\mathbf{I}^*$ is

[8] See Appendix A.1 for more information on the relationship between spread spectrum communications and watermarking.

received at the detector, it is DCT transformed and the supposed-to-be-watermarked coefficients are selected to form $X^* = x_1^*, \cdots, x_n^*$.

Before applying the popular similarity function in Eqn 4.8 applying some pre-processing to X^* is proposed to enhance the ability to detect the watermark.

$$sim\left(X, X^*\right) = \frac{X^* \cdot X}{\sqrt{X^* \cdot X^*}} \tag{4.8}$$

The pre-processing on X^* aims to decrease the value of the term $X^* \cdot X^*$ so that the similarity peaks more significantly if watermark exists. The way of pre-processing offered by the authors is a simple transformation like $x_i^* \leftarrow x_i^* - E\left(X^*\right)$ or $x_i^* \leftarrow sign\left(x_i^* - E\left(X^*\right)\right)$. Both of these transformations yield superior values for similarity.

4.2.2. DCT Algorithm 2 (Koch)

Our second DCT-based algorithm presented below is developed by E. Koch and J. Zhao [2]. The proposed approach, called Randomly Sequenced Pulse Position Modulated Code (RSPPMC) copyright labelling, is rooted in the well-known fact that typical digital images of people, buildings and natural settings can be considered as non-stationary statistical processes, which are highly redundant and tolerant of noise.

The watermark is some sequence of binary values, $w_i \in \{0,1\}$. The algorithm pseudorandomly selects 8×8 blocks of DCT coefficients of the image. Within each block b_i, two coefficients in the middle frequencies are again pseudorandomly selected.

In a later work by Koch and Zhao [107] that is based on this paper, a way of rejecting certain blocks or coefficient pairs is introduced. Such an operation is performed to improve the watermark transparency of the method but it also decreases the amount of payload.

For watermark embedding first each block is quantized using the JPEG quantization matrix and a given quantization parameter *Q*, in a manner reflecting acceptable information loss in the image for the application. Then two coefficients, namely $c_i(m_1,n_1)$ and $c_i(m_2,n_2)$ in the block b_i are selected and the absolute difference between the two is computed as follows:

$$\Delta_i = \left| c_i(m_1,n_1) \right| - \left| c_i(m_2,n_2) \right|$$

The code pulses, i.e. high or low, representing the binary code being embedded, are superimposed on the selected locations. In order to embed 1 bit of watermark information (code pulse), w_i, the selected coefficient pair is modified such that the distance becomes:

$$\Delta_i = \begin{cases} \geq q & if\ w_i = 1 \\ \leq -q & if\ w_i = 0 \end{cases},$$

where *q* is the parameter controlling the embedding strength. The quantized data is decoded; and then, inversely transformed to produce the labelled image data.

For the detection, the difference $\Delta_i^* = \left| c_i^*(m_1,n_1) \right| - \left| c_i^*(m_2,n_2) \right|$ is calculated from the test image. The detected bit is decided high (1) if $\Delta_i^* > 0$, or low (0) if $\Delta_i^* < 0$. In order to introduce a noise margin into the detection to take into account the JPEG quantization process, the tests can be modified as to decide 1 if $\Delta_i^* > p$, or 0 if $\Delta_i^* < -p$.

It is reported in the paper that a copyright label code could be embedded in several images, using pulses with sufficient noise margins to survive common processing, such as lossy compression, colour space conversion, and low pass filtering.

This scheme is later extended in [107] by the author to enforce a relationship between three coefficients instead of two. This allows encoding the watermark bit in a more robust way and provides a technique to skip blocks that are not suitable for

watermark embedding.

4.3. DWT Domain Techniques

4.3.1. DWT Algorithm 1 (Corvi)

This algorithm is presented by Marco Corvi and Gianluca Nicchiotti [21]. It is a non-blind detection system utilizing wavelet coefficients for watermark embedding.

The embedding media is the LL-subband of the wavelet decomposition. As a watermark, a Gaussian sequence of pseudo-random real numbers is created. Since all LL-subband coefficients are used for watermark embedding, the size of the watermark pattern is equal to the size of the LL-subband of the host image.

Embedding rule is very simple and it is formulated as below:

$$f_w(m,n) = f_{mean} + [f(m,n) - f_{mean}](1 + \alpha W),$$

where f_{mean} is the average value of the coefficients. The embedding rule is a modified version of the multiplicative formula, hence the perceptual adaptation is inherently performed both by the HVS-matching characteristic of the wavelet transform and the automatic fitting of the watermark pattern to the wavelet coefficients provided by multiplication operation.

The modification made in the original embedding formula makes the DC component of the LL-subband is unmodified since coefficient mean is subtracted prior to embedding.

Since the original image is assumed to be available at the detector, for the detection the LL-subband of the original image is subtracted from the tested image's LL-subband to obtain a watermark estimate. Then this estimated pattern is checked to be similar to the original watermark pattern.

4.3.2. DWT Algorithm 2 (Dugad)

This algorithm is developed by Dugad, Ratakonda, and Ahuja [31]. It is based on modifying the detail subbands of the wavelet transform representation of the image.

The watermark is a Gaussian sequence of pseudo-random real numbers, x_i, with zero mean and unit variance matching the size of the detail subbands after the host image is decomposed using a 3-level wavelet transform with Daubechies 8-tap filters.

The low subband is left out and all coefficients whose magnitude is above a given threshold T_1 in other (detail) subbands are selected for modification. Watermark is added to those detail coefficients only. T_1 controls the amount of watermark added to the image. This approach is different than Cox's [102] or Piva's [103] which fix the number of coefficients to be watermarked. Hence, they add same amount of watermark to two different images which possibly have different properties, like the amount of smooth or textured areas.

No explicit visual masking is performed for embedding. The detail subbands of DWT already contain edge and detail information. Hence masking is actually inherent in this technique. The embedding equation (Eqn 4.9) is similar to that in [102] where *i* runs over all detail DWT coefficients larger than T_1. In the paper α is taken as 0.2.

$$V_i' = V_i + \alpha |V_i| x_i \qquad (4.9)$$

Detection is performed using correlation. Only the coefficients above a detection threshold $T_2 > T_1$ are considered. T_2 is required to be strictly larger than T_1 for robustness since some coefficients, which were originally below T_1, may become greater than T_1 during image manipulations.

Detection process is blind (without the original image). The correlation z between the detail DWT coefficients $\hat{V}_i$ of the corrupted (attacked) watermarked image and the reference watermark is computed as in Eqn 4.10 where *i* runs over all

detail DWT coefficients larger than T_2 and M is the number of such coefficients.

$$z = \frac{1}{M}\sum_i \hat{V}_i x_i \qquad (4.10)$$

The detection threshold S is computed using the same equation (Eqn 4.13) as in Kim's paper [33]. The denominator is 2M instead of 3M as used in [103] because the number of samples over which the correlation is computed is generally smaller in this method and because no explicit masking is involved.

4.3.3. DWT Algorithm 3 (Kim)

The algorithm designed by J. R. Kim and Y. S. Moon [33] is the third wavelet based watermarking technique we are going to examine.

The watermark pattern to be added is a Gaussian distributed random vector is generated from a uniform distributed vector using Box-Muller transform. This pattern is added to perceptually significant coefficients that are determined using a level-adaptive thresholding scheme defined in Eqn 4.11. The threshold T_i for the ith DWT decomposition level depends on the maximum absolute coefficient C_i in that ith level subbands, namely LH_i, HL_i, and HH_i.

$$T_i = 2^{(\lfloor \log_2 C_i \rfloor - 1)} \qquad (4.11)$$

After the thresholds are calculated for each subband, the watermark is embedded to the selected coefficients adaptively according to the multiplicative formula $V_i' = V_i + \alpha V_i X_i$, where V_i' and V_i represent watermarked and original selected coefficients respectively, X_i is the i^{th} element of the random vector, and α is a small value like 0.04 for LL subband, in which coefficient values are considerably larger than other bands. For other subbands, the scale factor is properly tuned according to the decomposition level. Since the mean of wavelet coefficients is reduced by half as one level goes up, scale factor α is increased twice. In the paper,

scale factors 0.1, 0.2, and 0.4 are used for level 3, 2, and 1 respectively. This adjustment of the scale (gain) factors improves the performance of robustness and invisibility.

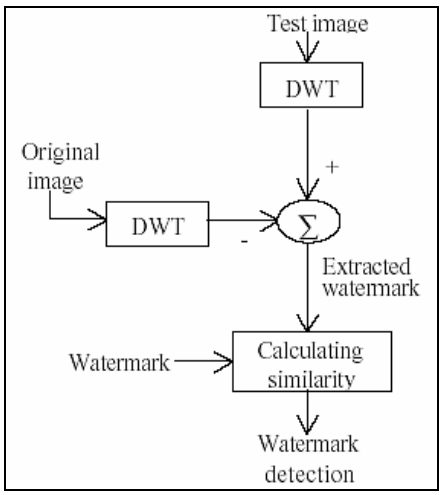


Figure 4.3 Block diagram representation for detection process in DWT-Algorithm 3.

Detection requires original image. As illustrated in Figure 4.3, first original and watermarked images are DWT transformed. Then, the wavelet subbands of the original image are subtracted from those of watermarked image. This gives us the extracted watermark. In order to decide on the detection result, we calculate the similarity between the original and the extracted watermark using Eqn 4.12 and apply a threshold *S* to that similarity value.

Following two formulas calculates the percentage of the remained watermark after the attack and the threshold, S, level respectively. In the first formula (4.12) X^* is the extracted watermark and in the second formula (4.13), *M* is the number of modified coefficients and α is the scale factor used in watermark insertion.

$$sim(X, X^*) = \frac{\frac{X \cdot X^*}{\sqrt{X^* \cdot X^*}}}{\frac{X \cdot X}{\sqrt{X \cdot X}}} \times 100 \qquad (4.12)$$

$$S = \frac{\alpha}{2M} \sum_{i=1}^{M} \left| \hat{V}_i \right| \qquad (4.13)$$

4.3.4. DWT Algorithm 4 (Wang)

In this algorithm introduced by Houng-Jyh Wang, Po-Chyi Su, and C. C. Jay Kuo in [46], the watermark is a sequence of length N_w consisting of real numbers between -1 and 1. We will denote the k[th] element of that watermark sequence by W_k, where $k \in [1 \cdots N_w]$.

The image is first DWT transformed to a desired level of resolution and then the watermark is embedded only to a selected number of significant coefficients in the detail subbands. The approximation subband remains unmodified. The coefficients in the subband *s* are denoted by $C_s(x, y)$.

Coefficient selection is motivated by the principle for the design of the multi-threshold wavelet codec (MTWC) [104, 105]. This method gives a perceptual weighting for different significant wavelet coefficients, and sets a limit on the bound of fidelity loss after watermark casting. The significant coefficient selection starts with assigning an initial threshold value computed by $T_s = \beta_s \frac{\max_{\forall x,y}[C_s(x,y)]}{2}$ to each of the detail subbands and initially all coefficients are unselected. β_s is the weighting factor of subband *s*. Then the algorithm proceeds as follows:

1. Select the subband with the maximum value of T_s.
2. For the selected subband, examine the set off all yet unselected coefficients, and out of that set select the coefficients greater than the current threshold of that subband.

3. Modify each one of the selected coefficients by adding the next watermark symbol W_k according to the following formula:

$$C'_{s,k}(m,n) = C_s(m,n) + \alpha_s T_s W_k ,$$

where C' is the watermarked image coefficient and $\alpha_s \in (0.0,1.0]$ is a scaling factor to adjust the trade-off between robustness and image fidelity (increasing/decreasing α_s will result in increased/decreased robustness but also increased/decreased perceptual disturbance).

4. Update the subband's threshold $T_s^{new} = \frac{T_s}{2}$.
5. Repeat steps 1 to 4 until there are no watermark symbols left to cast.

The detection mechanism makes use of the original image (hence non-oblivious). Let C^* be the coefficients of the received image which has probably gone through various attacks. The difference between the C^* and the original unmarked coefficients C in the selected significant coefficient position can be written as:

$$E^*_{s,k}(m,n) = C^*_{s,k}(m,n) - C_s(m,n)$$

Then the similarity between C^* and C is calculated as:

$$sim(C^*, C) = N_w \frac{\sum_{k=1}^{N_w} E^*_{s,k}(m,n) \cdot E_{s,k}(m,n)}{\left\| E^*_{s,k}(m,n) \right\| \left\| E_{s,k}(m,n) \right\|}$$

where $E_{s,k}(m,n) = \alpha_s T_s W_k$ is the amount of modification added by the embedding formula or in other words the original watermark.

In the same paper a blind (oblivious) detection scheme by truncating selected significant coefficients to some specified value is also introduced.

4.3.5. DWT Algorithm 5 (Xia)

This algorithm by Xiang-Gen Xia, Charles G. Boncelet, and Gonzalo R. Arce [44] is a 1-bit watermarking scheme based on adding a key-dependent Gaussian random sequence through an embedding formula and its detection by correlation.

The noise sequence, $N(m,n)$, is a zero mean and unit variance Gaussian sequence. First, the image $x(m,n)$ is decomposed into several subbands with a typical pyramid structure of DWT, which is shown in Figure 4.4 below.

All coefficients, $y(m,n)$, except the ones in the lowest frequency LL-band (LL_3 for Figure 4.4) are watermarked by adding the pattern $N(m,n)$according to the equation below:

$$\tilde{y}(m,n) = y(m,n) + \alpha y^2(m,n)N(m,n)$$

DWT coefficients at the lowest resolution remain unchanged. Then the two-dimensional IDWT of the modified coefficients combined with the unchanged coefficients is computed to obtain $\tilde{x}(m,n)$. For this resultant image to have the same dynamic range as the original image, it is modified according to Eqn 4.14. The resultant image $\hat{x}(m,n)$ is the final watermarked image.

$$\hat{x}(m,n) = \min[\max(x(m,n)), \max\{\tilde{x}(m,n), \min(x(m,n))\}] \quad (4.14)$$

The original image is assumed to be available to the decoder for watermark detection. The decoding method proposed is hierarchical and is described as follows: First, the received and original images are decomposed into four bands, i.e. LL_1, LH_1, HL_1, and HH_1 only.

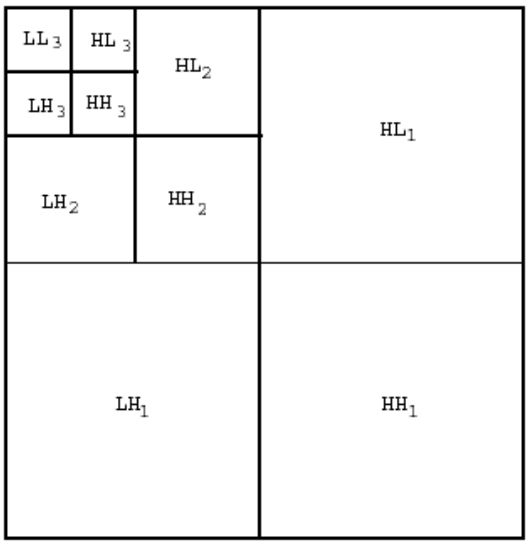


Figure 4.4 Representation of the pyramid decomposition in DWT.

Comparison starts at the HH_1 band. The signature added in the HH_1 band and the difference of the DWT coefficients in HH_1 bands of the received and original images are compared by calculating their cross correlations. If there is a peak in the cross correlations, the signature is said to be detected. Otherwise, the signature added in the HH_1 and LH_1 bands are compared to the difference of the DWT coefficients in the HH_1 and LH_1 bands, respectively. If there is a peak, the signature is detected. Otherwise, we consider the signature added in the HL_1, LH_1, and HH_1 bands. If there is still no peak in the cross correlations, we continue to decompose the original and the received signals in the LL_1 band into four additional subbands LL_2, LH_2, HL_2 and HH_2 and so on until a peak appears in the cross correlations. Otherwise, we decide that the signature can not be detected.

4.3.6. DWT Algorithm 6 (Xie)

In this paper by Xie and Arce [42], authors extend their previous work on digital image signatures [98, 99]. They use a method that they call “etching algorithm” to embed the watermark into the entire low frequency band of the wavelet decomposed

image. A non-overlapping 1×3 horizontal window is slided over the subband coefficients. The elements the window overlaps are denoted as b_1, b_2, b_3, which are coefficients at the coordinates $(i-1, j), (i, j), (i+1, j)$.

These 3 coefficients are sorted and the one in the middle (the median of three) is modified using a nonlinear transformation. Let $b_{(1)}$ denote the lowest of the three coefficients, $b_{(2)}$ the median and $b_{(3)}$ the highest of coefficients.

First, the range between $b_{(3)}$ and $b_{(1)}$ is divided into intervals of length Δ (Eqn 4.15). This creates regions R_k whose boundaries can be denoted as $[l_{k-1}, l_k)$. For a median coefficient $b_{(2)} \in R_k$ and watermark bit x the transformation for engraving a single watermark bit is done according to Eqn 4.16.

$$\Delta = \alpha \cdot \frac{\left(|b_1| + |b_3|\right)}{2} \tag{4.15}$$

$$b'_{(2)} = \begin{cases} l_k & k \text{ is odd and } x = 1, \text{ or } k \text{ is even and } x = 0 \\ l_{k-1} & k \text{ is even and } x = 1, \text{ or } k \text{ is odd and } x = 0 \end{cases} \tag{4.16}$$

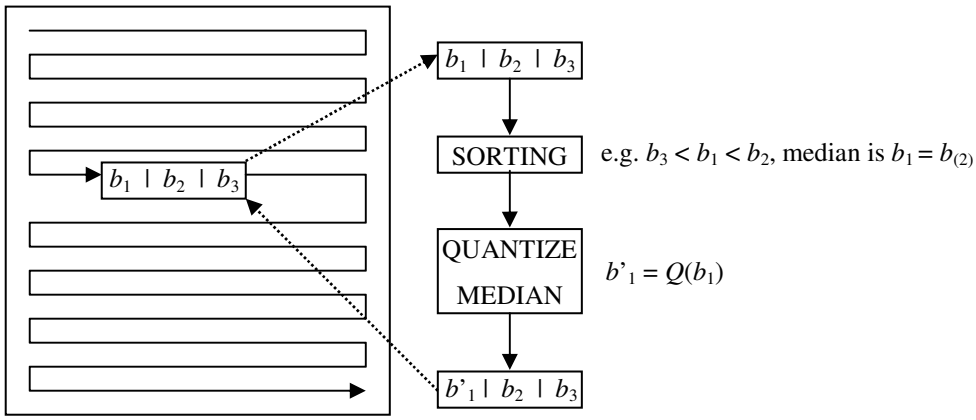


Figure 4.5 A pictorial summary of the Xie's bit engraving method.

Figure 4.6 gives a pictorial representation of this transformation formulated above. Detection is blind. The sliding window is applied to the suspected image coefficients. The median of the sliding window is determined and quantized to obtain a reconstruction point. The bit value associated with that reconstruction point becomes the extracted watermark bit.

The authors proposed to adopt the SPIHT compression algorithm [100] into this watermarking method. The lowest bit level at the end of coding is denoted by m. The watermark capacity can be greatly increased if multiple bits can be engraved in each window location. This is possible if the range $b_{(3)} - b_{(1)}$ is significantly larger than the threshold 2^m. In that case the length-Δ intervals are further splitted into smaller regions.

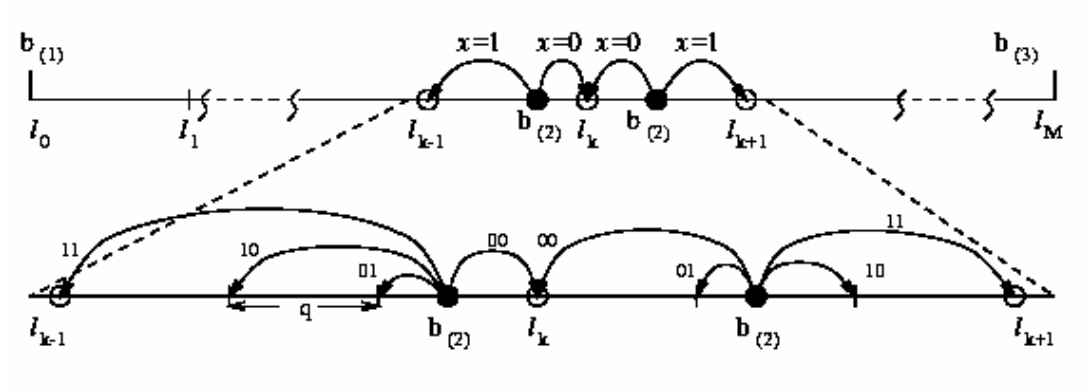


Figure 4.6 A representation of the single bit (above) and multi-bit (below) engraving method as given in Xie's paper [42].

4.4. Experiments and Results

In this part of this thesis, we will give the results of the experiments that we performed on the 9 algorithms presented in Section 4.3. We performed experiments on 78 different distorted versions of a watermarked image for each algorithm. The details of these are given in Table 4.2.

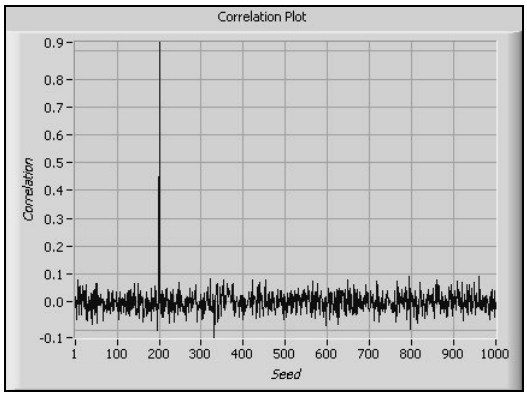
Generally, watermarks embedded in the frequency domain are observed to be more robust to many forms of attacks compared to the spatial domain watermark. In order to achieve robustness, the watermark has to be embedded in salient portions of the host signal. The frequency representations easily allow selecting the low- and mid-frequency coefficients which carry most of the signals energy. The selection of suitable transform domain coefficients is one of the most important design issues, as this choice greatly affects robustness, imperceptibility and security of the resulting watermarking scheme. The major difference, actually, between the watermarking schemes explained in Section 4.3 lies in the different coefficient (host feature) selection strategies and in this section we will see how significantly the robustness is affected resultantly.

The experiments are performed in two groups each with a different controlled parameter. In the first group (Group 1) the number of modified coefficients or the number of embedded bits is kept constant and robustnesses are compared. In the second group (Group 2), the amount of watermark energy is kept constant by tuning the embedding parameters such that the PSNR of the watermarked images are about 35dB (Table 4.1) and the comparisons are repeated.

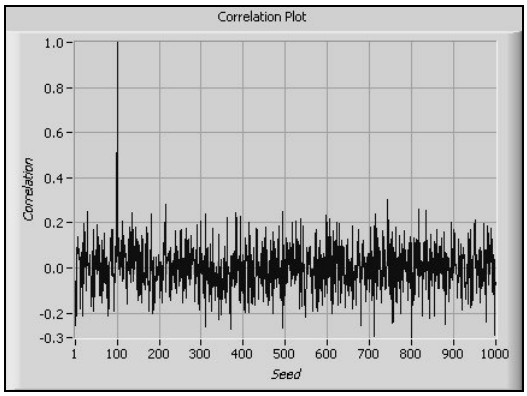
Table 4.1. PSNR and MSE values for watermarked images of the Group 2 experiments.

	BRUYN	CORVI	COX	DUGAD	KIM	KOCH	WANG	XIA	XIE
PSNR	34.98	34.94	35.04	35.04	35.01	35.00	35.03	34.99	35.06
MSE	20.65	20.81	20.35	20.36	20.49	20.53	20.44	20.59	20.30

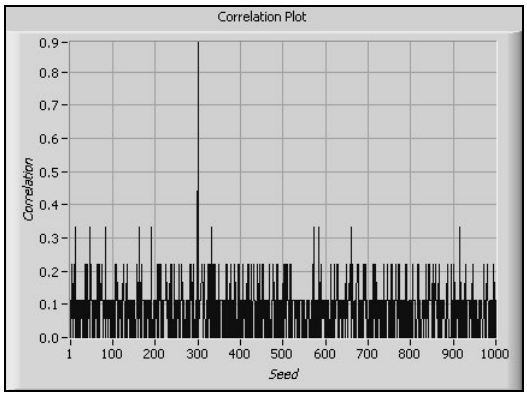
Before evaluating the outcomes of these experiments in detail we give the results of a *confidence check* on all of the 1-bit algorithms in Figure 4.7. The confidence check operation tries to find the correlation between a watermarked image that is not attacked yet and a set of possible pseudorandom watermark patterns each of which comes out of a separate seed. Ideally we see a single peak in these plots at the exact location where the embedded pattern's seed is standing. We can use these plots on an attacked image also to see if the watermark is still distinguishable from other possible watermark



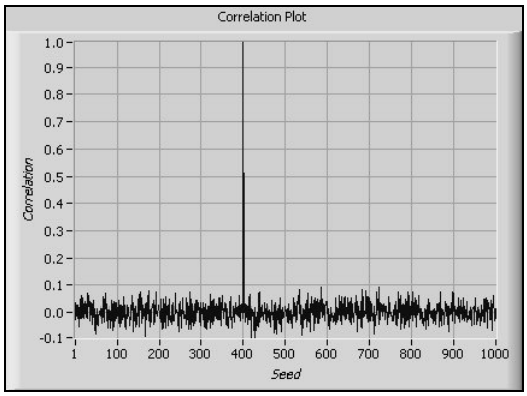
(a) Corvi



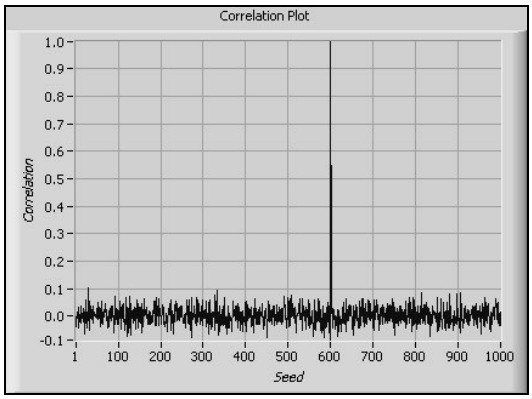
(b) Cox



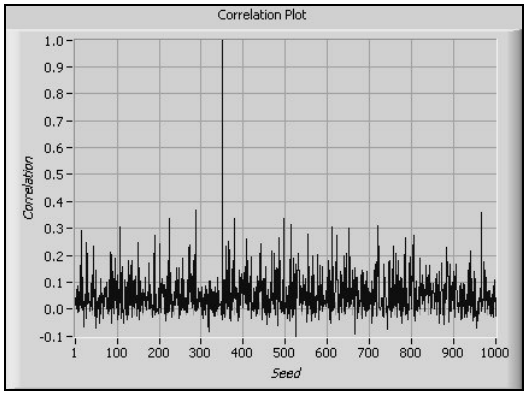
(c) Dugad



(d) Kim



(e) Wang



(f) Xia

Figure 4.7 Confidence checking on 6 single-bit algorithms. Correlation values between watermarked images and 1000 randomly generated watermarks are plotted.

Table 4.2 List of attacks performed on the test images.

Attacks	**Options**		**Implemented by**
Median Filtering	Filter sizes: 2×2 to 8×8		Stirmark + MATLAB
JPEG Compression	Quality: 10, 15, 20, 25, 30, 35, 40, 50, 60, 70, 80, 90		Stirmark
Cropping	% of original image: 1, 2, 5, 10, 15, 20, 25, 50, 75		Stirmark + MATLAB
Sharpening	Filter size 3×3		Stirmark
Gaussian Filtering	Filter size 3×3		Stirmark
Rotate + Scale	Angles: -2, -1, -0.75, -0.5, -0.25, 0.25, 0.5, 0.75, 1, 2, 5, 10, 15, 30, 45, 90		Stirmark
Random Bending	Default options		Stirmark
Row-Column Removal	*# of Rows removed*	*# of Columns removed*	Stirmark + IrfanView
	1	1	
	1	5	
	5	1	
	17	5	
	5	17	
Up-Down Scaling	Scale factors: 0.5, 0.75, 0.9, 1.1, 1.5, 2.0		Stirmark + IrfanView
Shearing	*x*	*y*	Stirmark + IrfanView
	0.0	1.0	
	0.0	5.0	
	1.0	0.0	
	1.0	1.0	
	5.0	0.0	
	5.0	5.0	
EZW Compression	Using Haar wavelet bits-per-pixel: 0.01, 0.02, 0.05, 0.10, 0.15, 0.20, 0.25, 0.50, 0.75, 1.00, 1.25, 1.50		Dedicated executable

These tests run automatically once required parameters and initial and final values for the seeds are given as a capability of the software we prepared (Figure 4.8). The user can zoom in and out of the plot and copy the displayed bitmap of the plot to Windows clipboard upon user's wish at the end of the operation. Without such a tool to execute the commands automatically each one of these plots require more than 5000 lines of DOS commands, which would be a very bulky work to do.

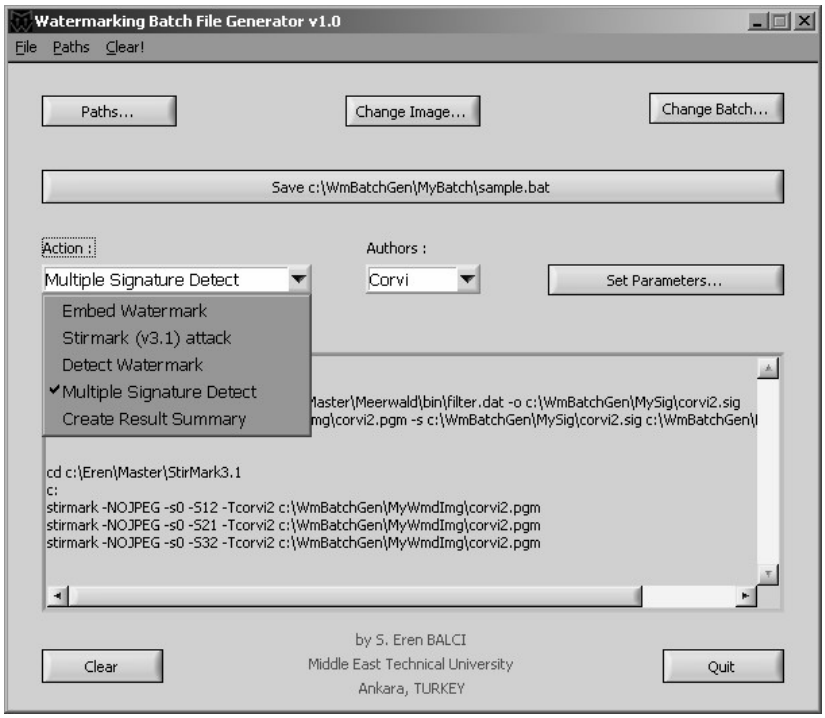


Figure 4.8 Test user interface panel for our Watermark Batch File Generator program.

4.4.1. Experiments on Removal Attacks

The complete list of attacks performed on watermarked images is given in Table 4.2. The ones that can be classified as a removal attack are median filtering, lossy JPEG and EZW compression, sharpening and Gaussian filtering. In the first controlled group of experiments all single-bit algorithms (Corvi, Cox, Dugad, Kim, Wang, Xia) the number of modified coefficients are fixed at 1000 (either DCT or DWT

coefficients) and for multiple bit algorithms (Bruyn, Koch, Xie) the watermark message length is 64 bits.

The test results for Group 1 and Group 2 set of experiments for JPEG compression, EZW compression and median filtering are provided in Figure 4.10 and 4.13, 4.11 and 4.14, and 4.12 and 4.15 respectively. The results for Gaussian filtering and sharpening operations are given in Table 4.3.

When we inspect the results we see that the DCT Algorithm 1 by Cox et al performs the best among all others against JPEG compression and low- and high-pass filtering. First of all, JPEG compression is performed in the DCT domain. Therefore, the good performance of a DCT-based technique should not be a surprise. In Chapter 3 we mentioned that when the domains which the watermark is embedded and the attack is performed match, we have the opportunity to favour the components (in this case DCT coefficients) that will possibly survive the distortion to a greater extent for watermark embedding.

Another property of this algorithm is that watermark is added to the mid-frequency coefficients, such that it will be both robust and invisible. JPEG compression, median and Gaussian filtering are all low-pass in nature and sharpening is a high-pass process. Residing on the large coefficients of the middle frequency band, the Cox watermark survives all. Just to check where the watermark will start to fade we made a few more median filtering with larger filter sizes. Even for the 15×15 case the correlation value is 0.534, a value some DWT-based and the spatial domain algorithms can hardly attain.

Table 4.3. Test results for sharpening and Gaussian filtering.

	BRUYN	CORVI	COX	DUGAD	KIM	KOCH	WANG	XIA	XIE
GROUP 1									
Sharpening 3x3	0.880	0.542	0.980	0.111	0.769	0.063	0.445	0.853	0.80
Gaussian filtering 3x3	0.880	0.786	0.998	1.000	0.293	0.469	0.768	0.705	1.00
GROUP 2									
Sharpening 3x3	0.168	0.556	0.926	0.000	0.740	0.264	0.392	0.903	0.725
Gaussian filtering 3x3	0.174	0.812	0.993	0.875	0.249	0.400	0.661	0.797	0.750

There is also another algorithm that performs in some cases equally well. It is Xie's algorithm which non-linearly modifies (quantizes) the median of a 1×3 sliding window in the wavelet domain. Except the median filtering experiments its robustness is very close to Cox's figure of merits. The decreased performance of Xie's method under median filtering is actually intuitively logical since the algorithm depends on the separation between coefficients in the sliding window. When the coefficients are smoothed with a median filter the detectors median values will get closer to the large and the small coefficients (b_3 and b_2 in Figure 4.5). Hence, the intervals are smaller so that the median value may easily swap to the next reconstruction point of the quantizer.

Cox algorithm has an advantage over Xie. It is the availability of the original image at the detector, that is, it is a non-blind scheme. However, Xie's algorithm makes a blind detection. If the original image is available at the detector, it is very easy to get a good estimate of the embedded watermark by simply subtracting it from the test image. On the other hand, blind schemes which depend on correlation values for detection must behave the image as a noise, where the signal(watermark)-to-noise ratio is very low.

What can we do to improve the correlation values? When we look at the figures about how much distortion is implied by the watermark embedding procedures of each algorithm using the values suggested we get the results in Table 4.4. Since we have seen that PSNR and MSE can be misleading, we used image quality index calculation described in Section 3.1.6.1 Eqn 3.9. We also put the PSNR and MSE correspondents as reference. The values in the table are really quite high (except Koch with 0.741 and Cox with 0.8345) which means the watermark strength can be increased more as the first and the most effective way of improving the detection.

Here, it is not surprising that Cox and Koch (both are DCT domain algorithms) result in more distortion on the original image, because these algorithms do not use special masking techniques whereas all DWT algorithms inherently involve perceptual masking. Therefore, they cause much less distortion.

Table 4.4. Some figures of fidelity for the watermarked images.

	BRUYN	CORVI	COX	DUGAD	KIM	KOCH	WANG	XIA	XIE
Image Quality Index	0.996	0.936	0.8345	0.981	0.741	0.981	0.950	0.988	0.976
PSNR (dB)	49.184	34.193	27.022	39.422	33.737	42.993	33.23	37.711	33.803
MSE	0.785	24.759	129.07	7.428	27.501	3.264	30.909	11.013	27.08

4.4.2. Experiments on Geometric Attacks

All attacks that we performed on the algorithms other than the ones mentioned in the previous section are geometric attacks. We did not apply any cryptographic or protocol attacks within the work of this thesis. Namely, we tested the algorithms against cropping, rotation and scale together (so that the resultant attacked image is of the same size as the original image), random bending (Figure 3.23), row and column removal, up-down scaling and shearing (Figure 4.9).

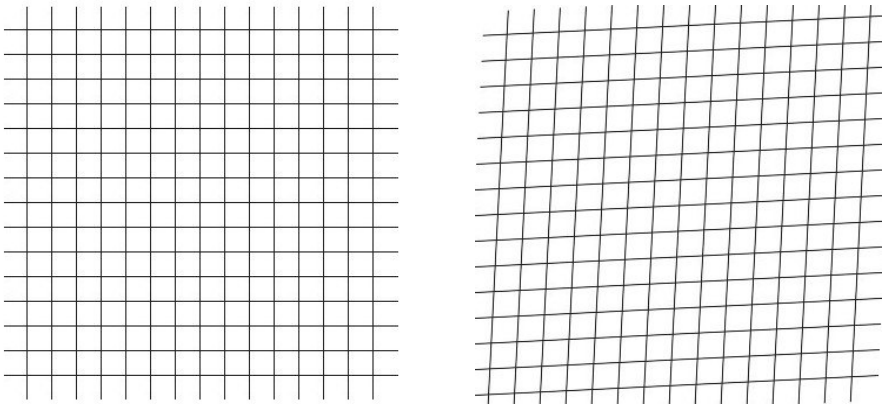


Figure 4.9 The effect of shearing attack demonstrated on a grid. The grid on the right is the result of the highest distortion applied by Stirmark, that is, 5% in each direction.

Neither of the algorithms that we have in our experiment set are designed especially for geometric attacks. Therefore, we do not expect to see high correlation values as in the simple removal attacks case.

We now give the test results of Group 1 and Group 2 experiments starting with cropping (Figure 4.16 – no cropping results for Group 2), then rotation and scale (Figure 4.17 and 4.21), random bending (Table 4.5), row and column removal (Figure 4.18 and 4.22), up-down scaling (Figure 4.19 and 4.23) and finally shearing (Figure 4.20 and 4.24) respectively. For cropping, the images are padded with zeros before we try to detect the watermark, using MATLAB. As expected, as the amount of available image portion decreases in the cropping attack the correlation value decreases since there is less and less power left. However, we see a wavy figure for the Xia algorithm. This is a result of the hierarchical detection mechanism of the algorithm. In Xia's method, all detail subbands are watermarked and during detection a high correlation in only one of the subbands results in a positive decision with that correlation given as the "amount" of detection. In the cropping case, the values on the plot are sometimes the overall correlation value and sometimes it is generated by only one of the subbands that "catches" the watermark.

For the rotation and scale case (Figures 4.17, 4.21, 4.19, 4.23), the situation is similar. We have robustness only to some extent. For small rotations unto 1 degree in both directions Cox watermark manages to survive. Some other algorithms like Xie and Dugad also show some amount of resistance.

The random bending attack turns out to be successful for all algorithms. It prevents the detector to synchronize with the watermark and correlation values are decreased considerably.

Table 4.5 Test results for Stirmark random bending attack.

	BRUYN	CORVI	COX	DUGAD	KIM	KOCH	WANG	XIA	XIE
Group 1	0.400	0.218	0.516	0.000	0.000	0.281	0.060	0.313	0.200
Group 2	0.152	0.216	0.382	0.000	0.000	0.140	0.008	0.061	0.000

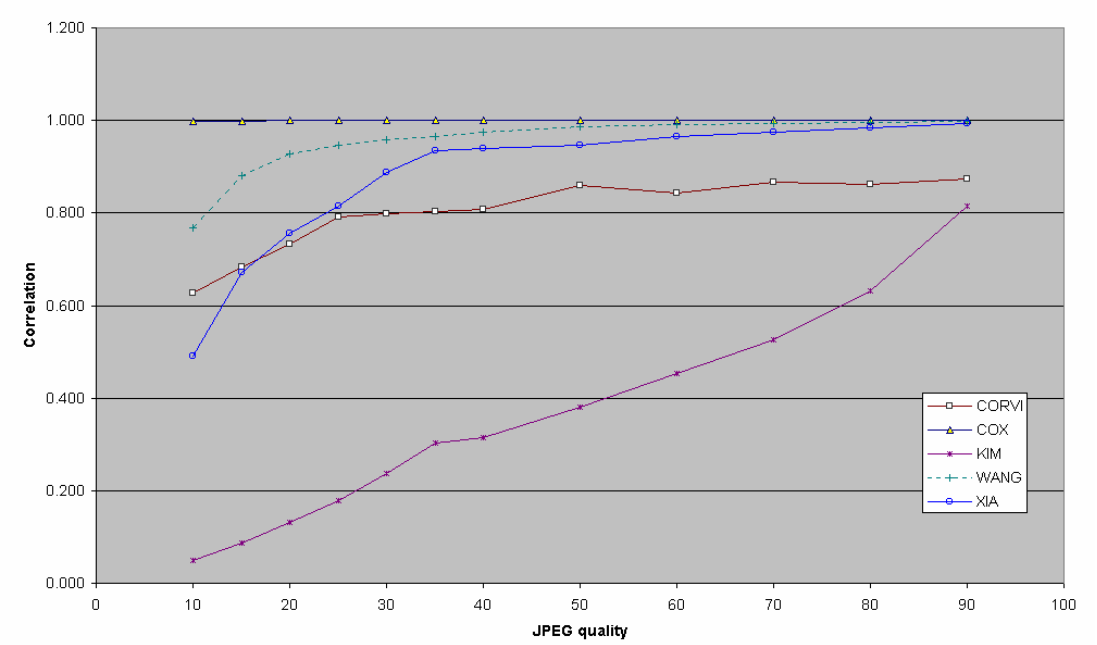
For all three attacks that will follow, the attacked images are either larger or smaller in size. We restore the image to the original size before detection using the re-sample property of IrfanView image processing program utilizing a B-spline filter. For the scaling case (Figure 4.19 and 4.23), the images generated by the scaling

attack of Stirmark are up or downsampled to their original image size before detection operation. For upscaling, where we downsample before detection, the correlation is generally higher since original watermarked image data is more conserved in this case than downscaling followed by upsampling, where lots of unwatermarked pixels are generated to bring the image back to its original size.

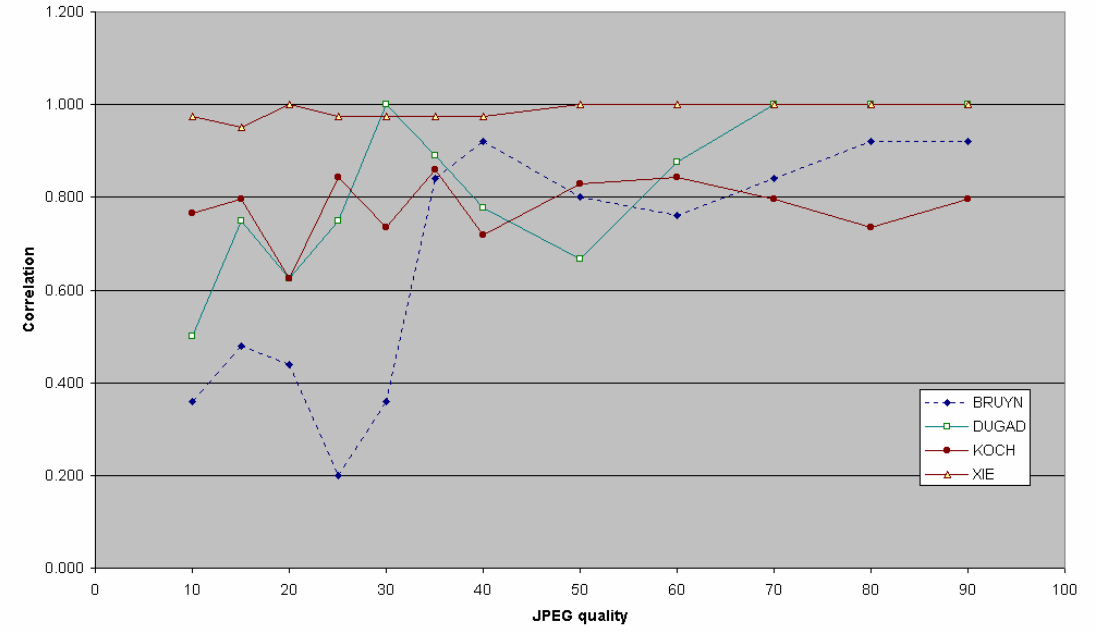
We actually predicted such results, that is, the algorithms are not that robust as in removal attacks case, since they are not designed specifically to be robust to rotation for example. However, we are not completely unable to improve this situation. Although the algorithms do not have any inherent capabilities to withstand these attacks. With the help of an image registration technique we may have chance to "undo" the distortion causes by the attacks before detection. There is such a program called CREG [27] that claims that it can partly invert the distortion when a reference image can be found. A copy of the same image watermarked with a different key, an undistorted copy of the watermarked image, or the original image itself can serve as a reference image.

As an example to improvement in detection we chose the Xia algorithm. We watermarked the same Lena image with strength 0.25 instead of previous value 0.2 and made some experiments with JPEG compression, median filtering, row and column removal, and cropping as attacks. In all cases but one (which include both removal and geometric types of attacks) the improvement in the correlation values are significant. For cropping the improvement is not as much as the first three types of attack. We also found that the visual impact of this increase is at satisfactory levels. The new image quality index is 0.922 compared to the previous value of 0.988. Here, we once again see the trade-off between robustness and fidelity (Figure 4.25).

Finally, in Table 4.6 on page 99 we present some experiment results for the spatial domain binary logo embedding technique introduced in Chapter 3. The embedded watermark (which causes a distortion that results a PSNR of 35dB) is successfully detected after most compression attacks and even after the row-column removal.

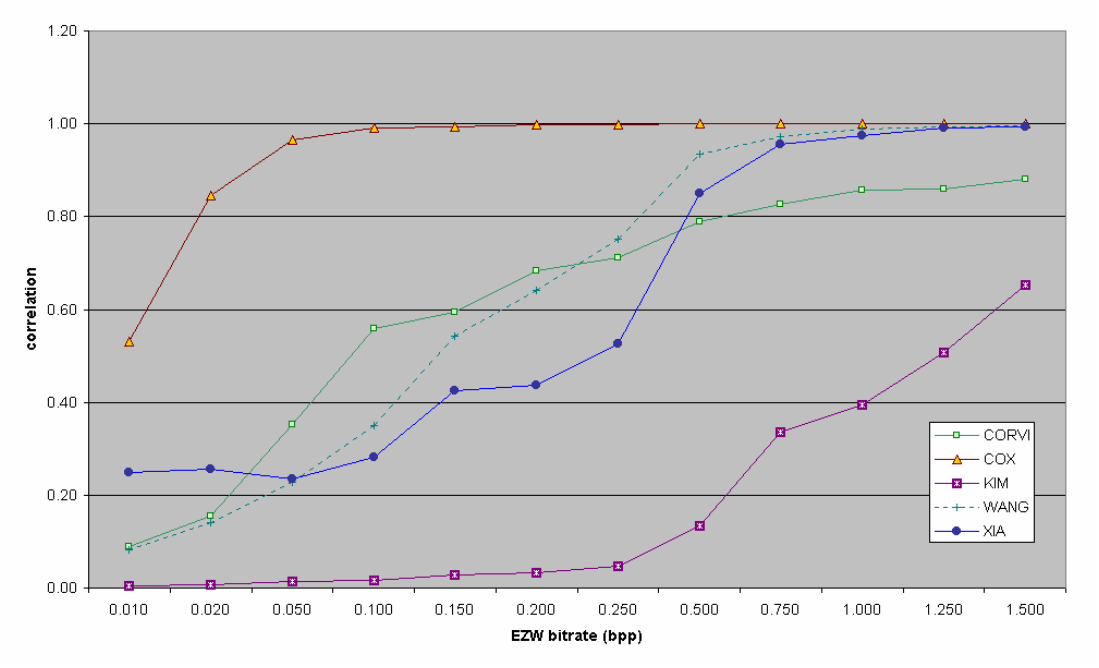


(a)

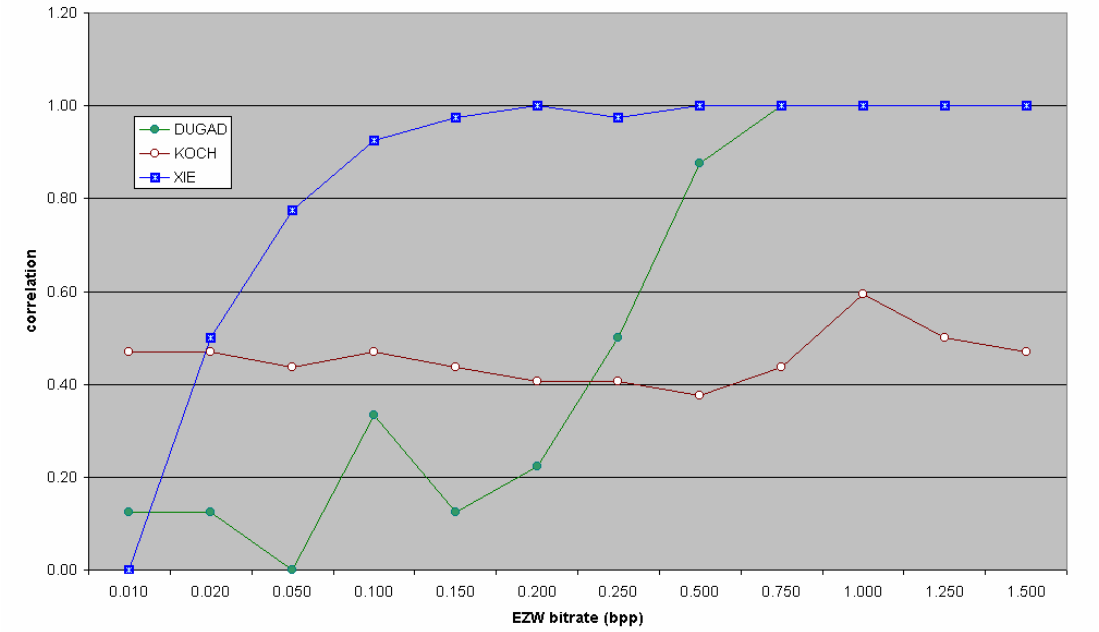


(b)

Figure 4.10 Group 1 test results for JPEG compression (a) non-blind algorithms (b) blind algorithms.

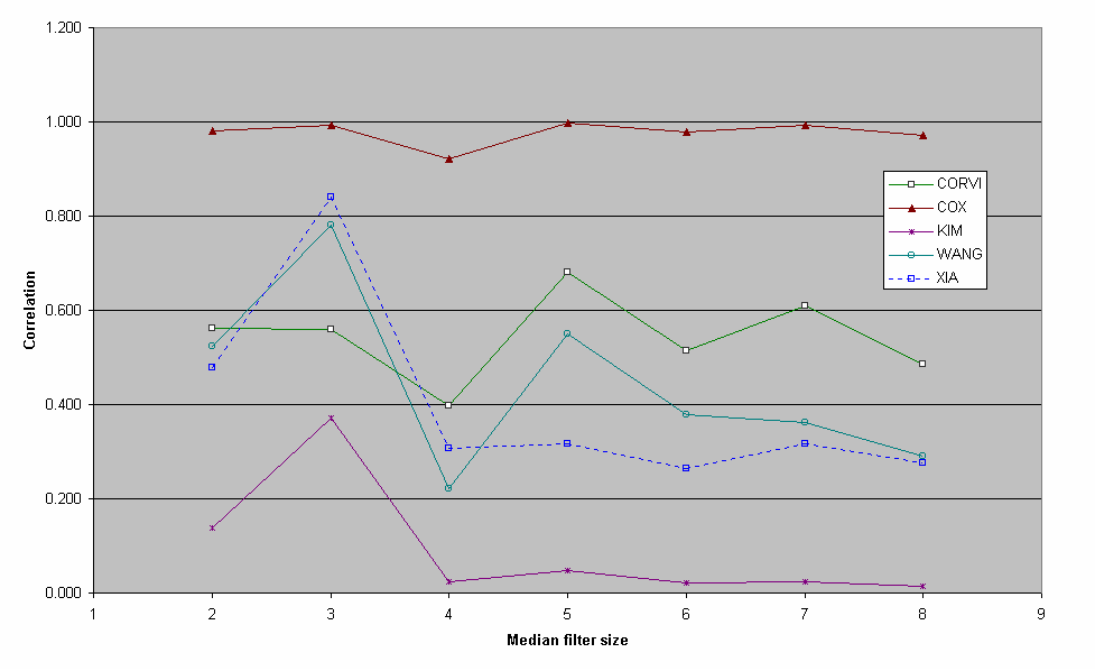


(a)

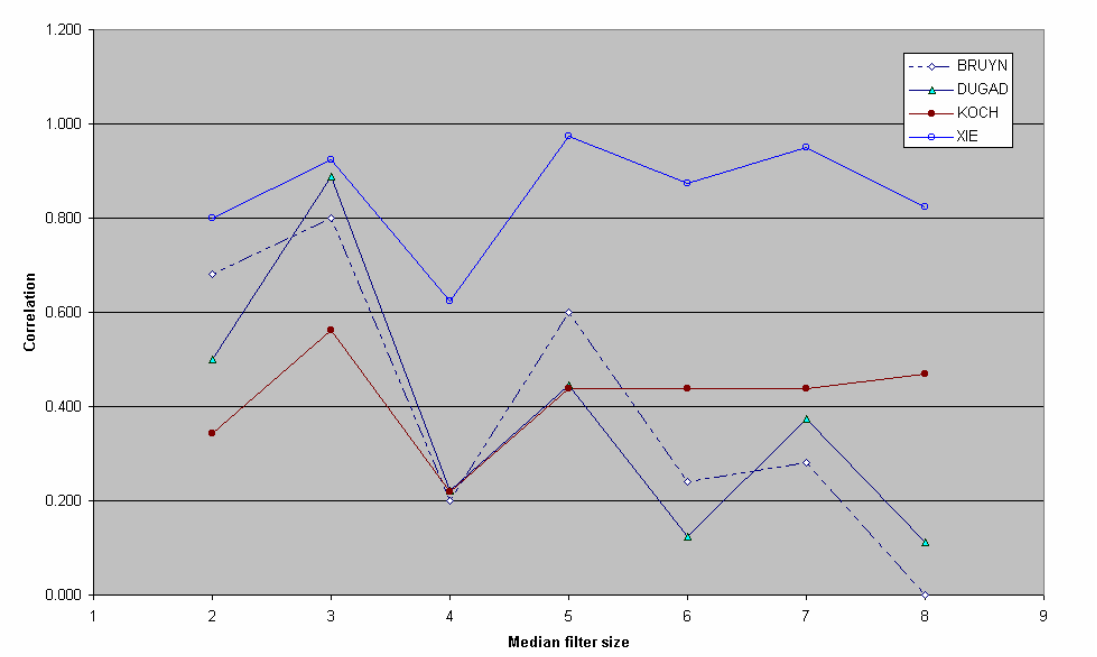


(b)

Figure 4.11 Group 1 test results for EZW compression (a) non-blind algorithms (b) blind algorithms.

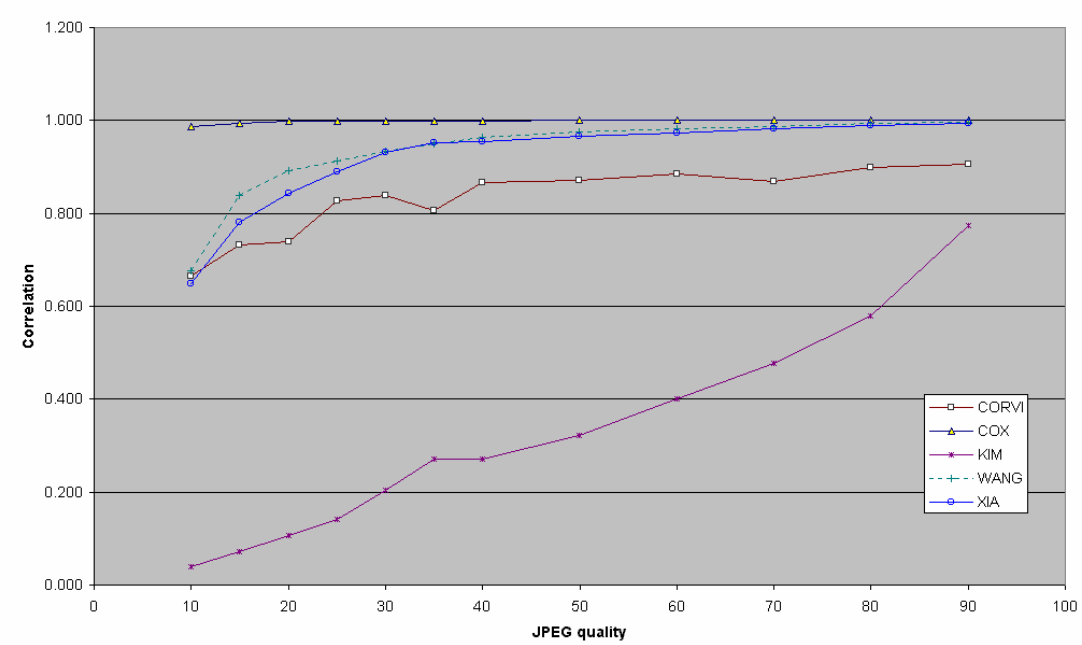


(a)

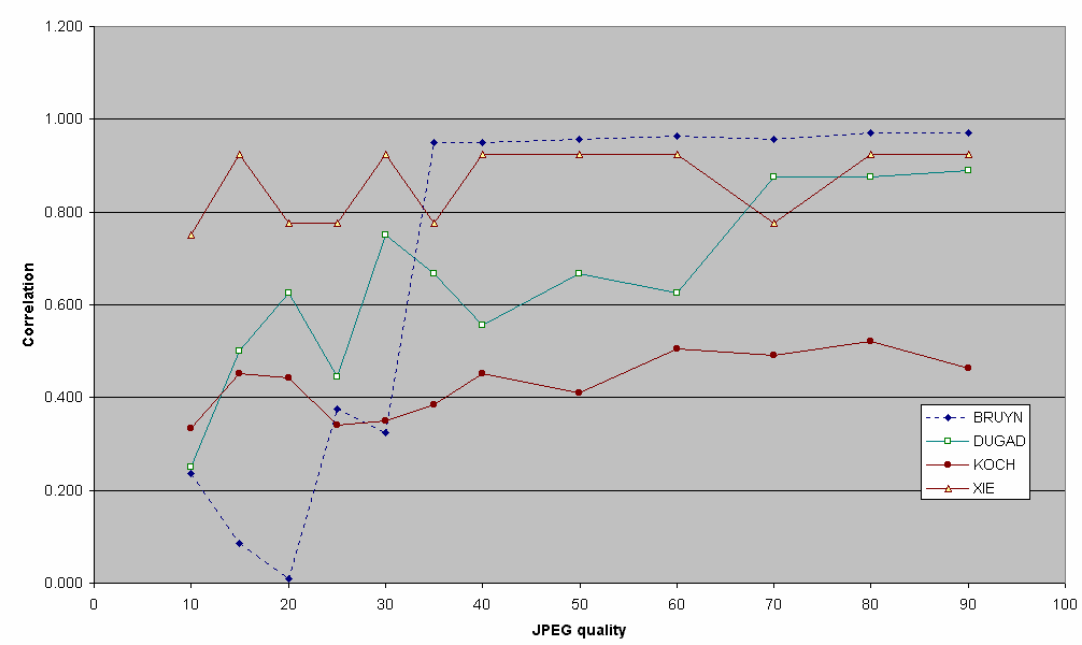


(b)

Figure 4.12 Group 1 test results for median filtering (a) non-blind algorithms (b) blind algorithms.

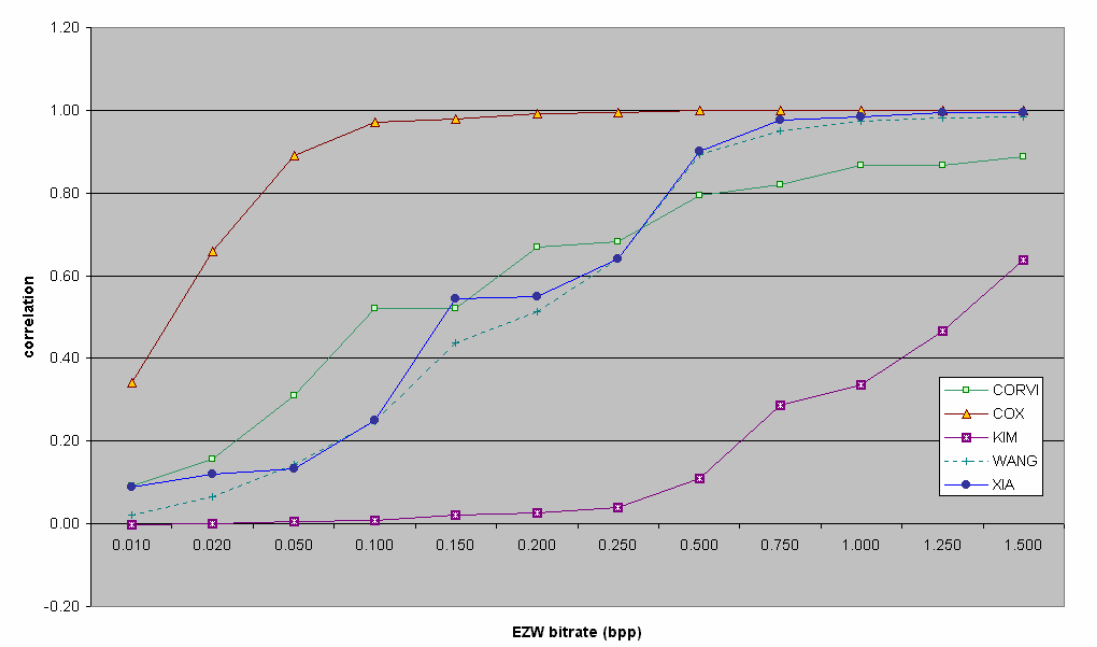


(a)

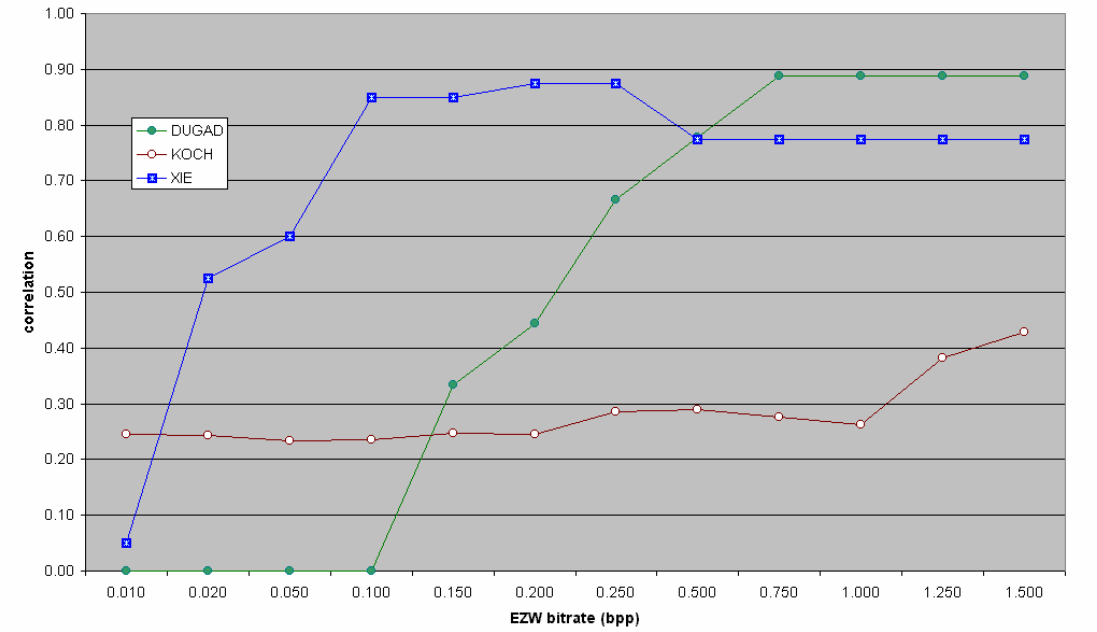


(b)

Figure 4.13 Group 2 test results for JPEG compression (a) non-blind algorithms (b) blind algorithms.

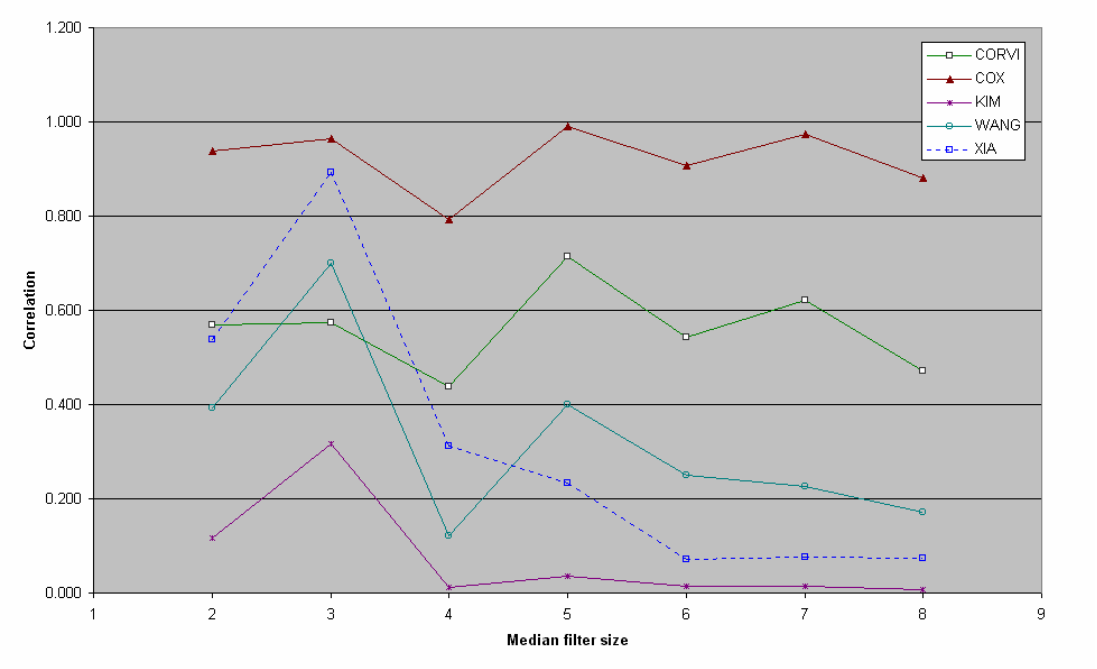


(a)

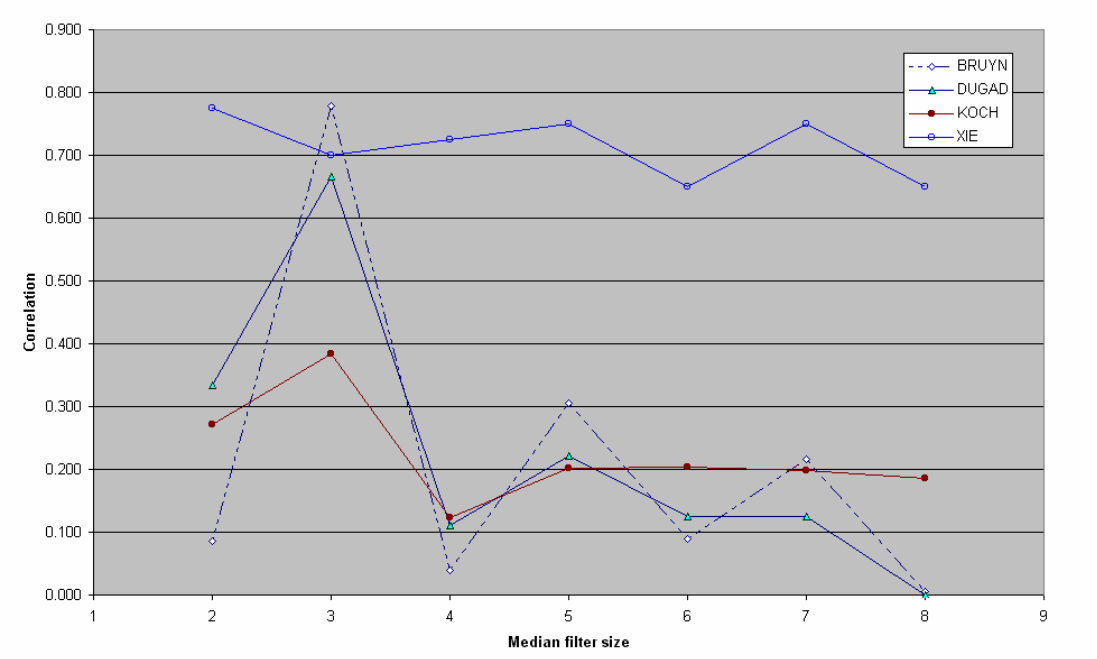


(b)

Figure 4.14 Group 2 test results for EZW compression (a) non-blind algorithms (b) blind algorithms.

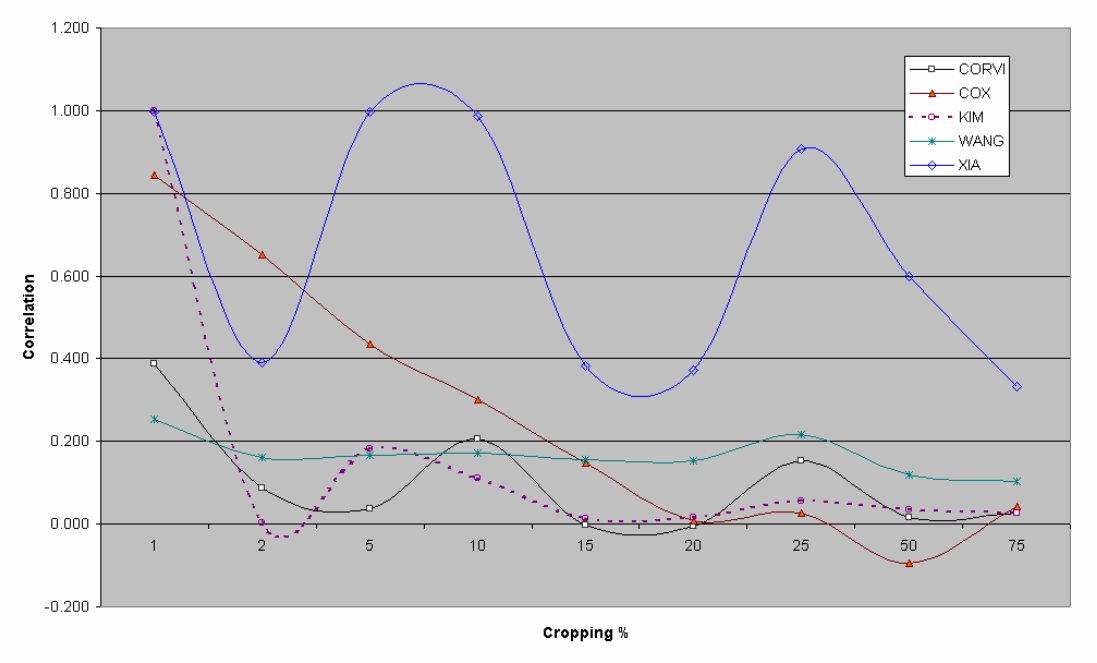


(a)

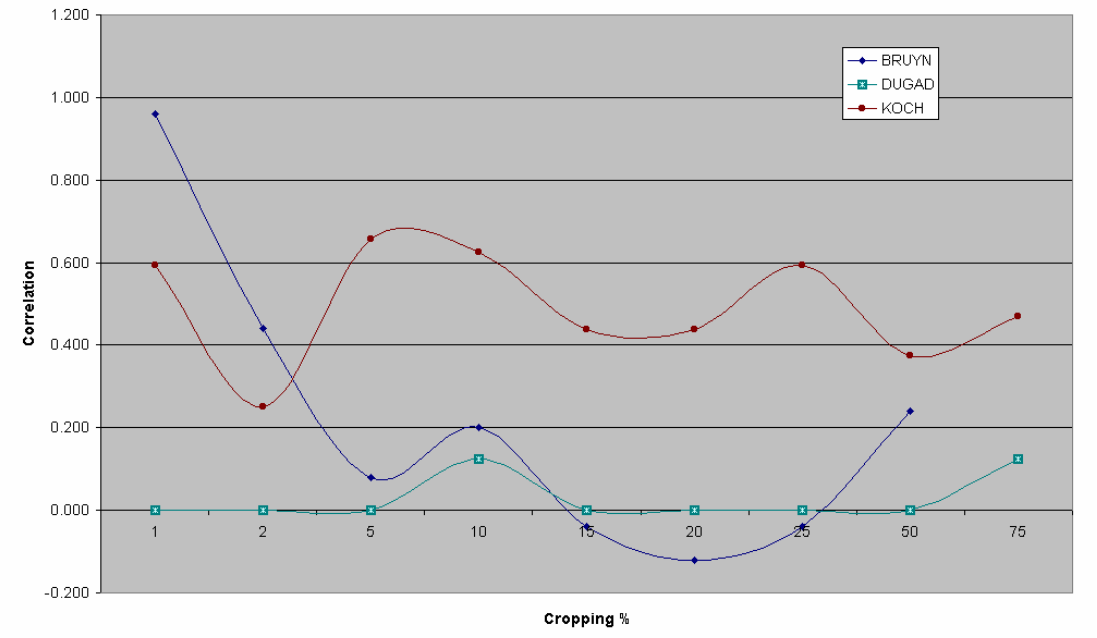


(b)

Figure 4.15 Group 2 test results for median filtering (a) non-blind algorithms (b) blind algorithms.

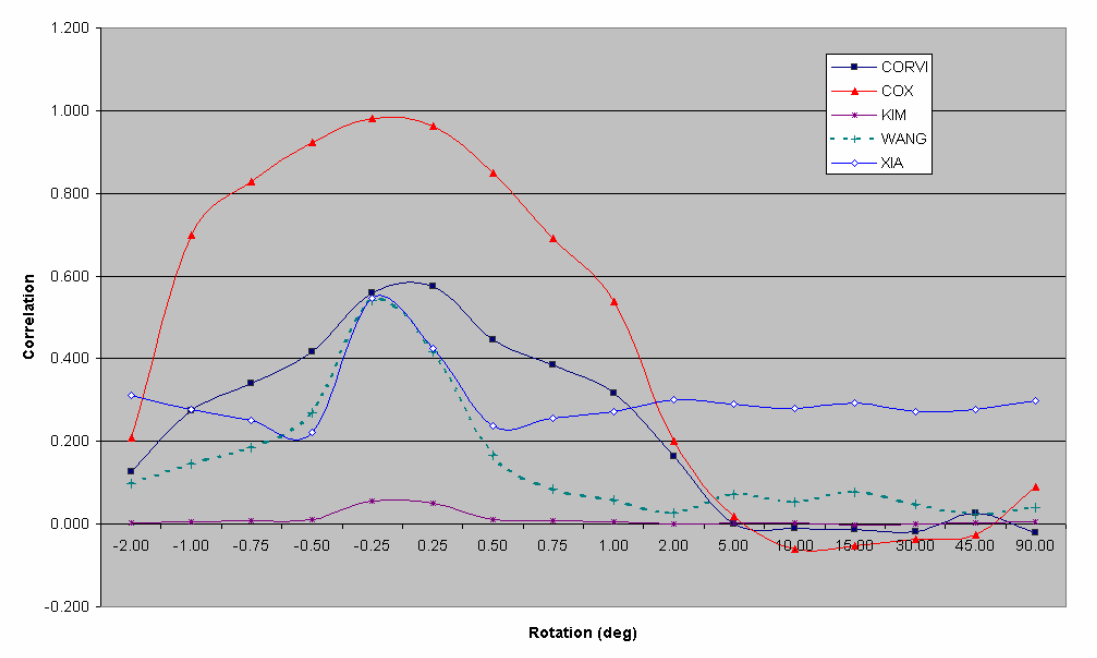


(a)

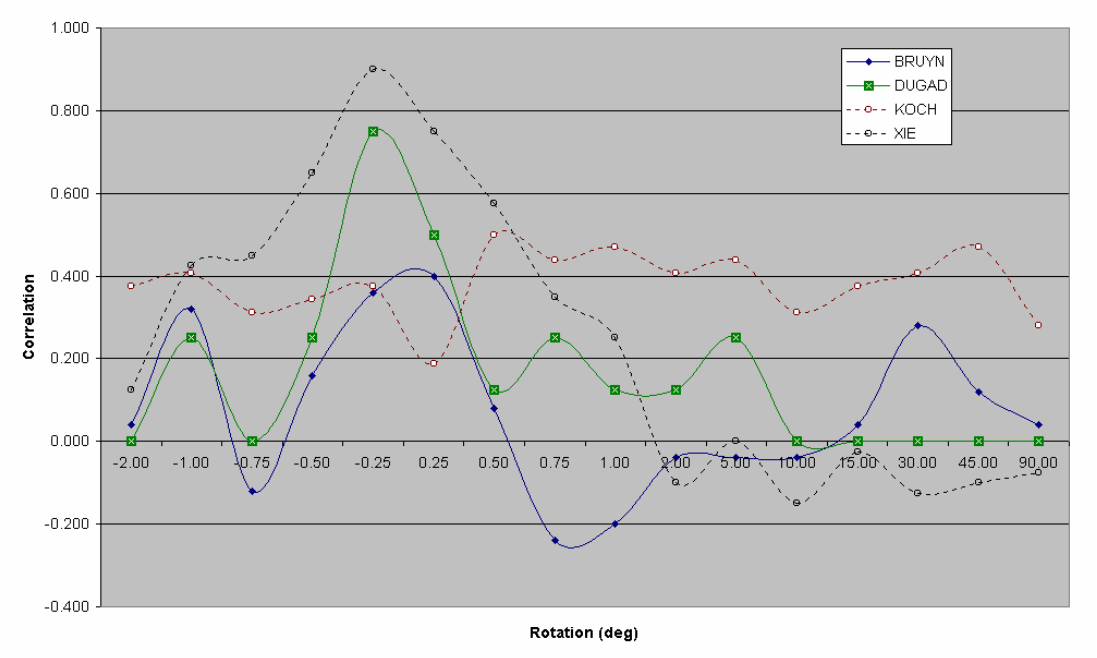


(b)

Figure 4.16 Group 1 test results for various cropping attacks. The x-axis is percentage of the original picture removed (a) non-blind algorithms (b) blind algorithms.

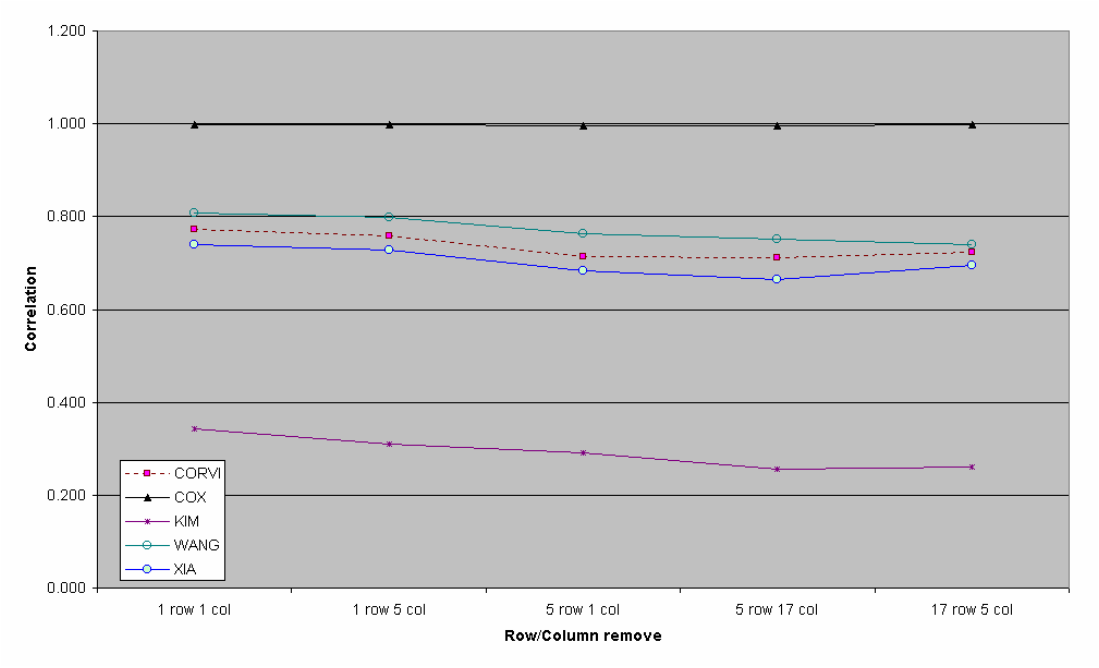


(a)

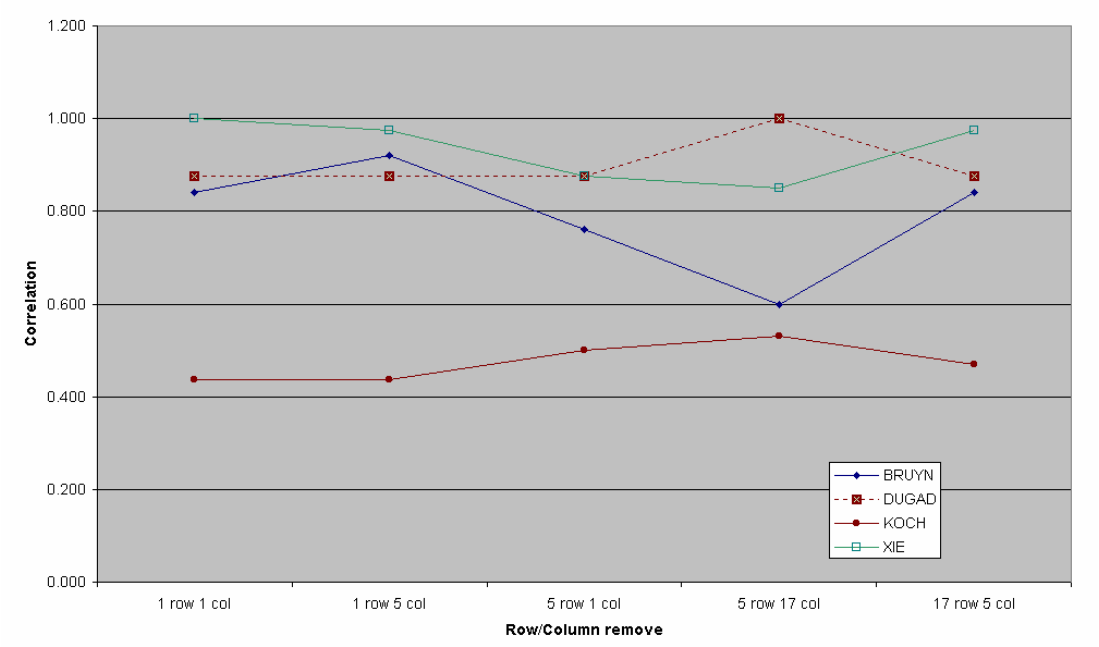


(b)

Figure 4.17 Group 1 test results for rotate and scale attacks with 16 different angles both clockwise and counter-clockwise (a) non-blind algorithms (b) blind algorithms.

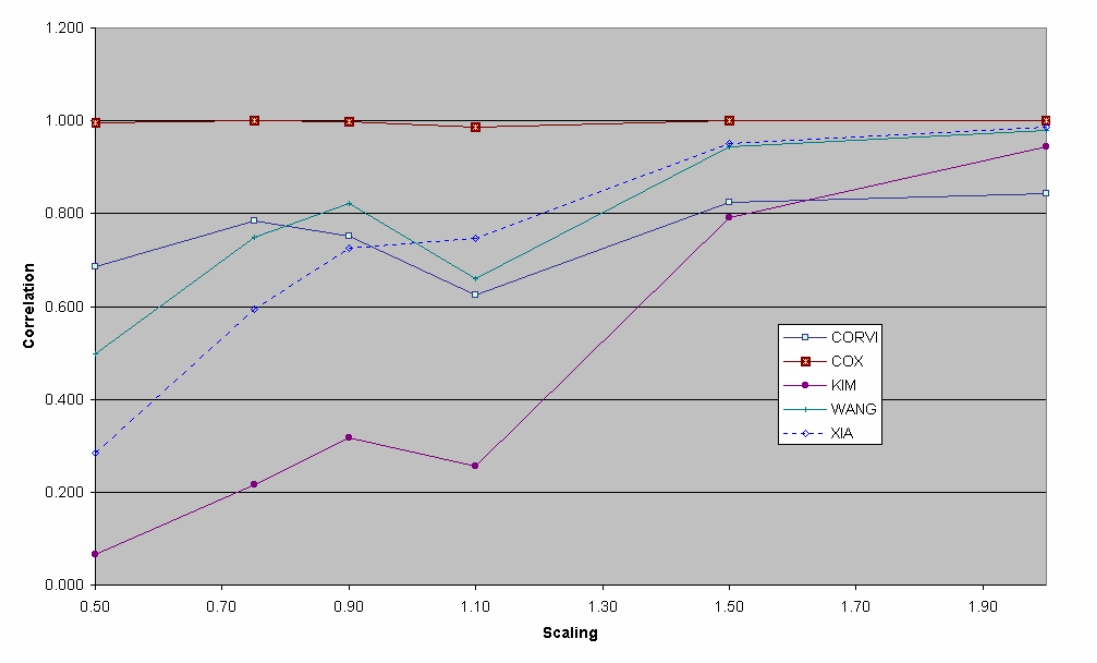


(a)

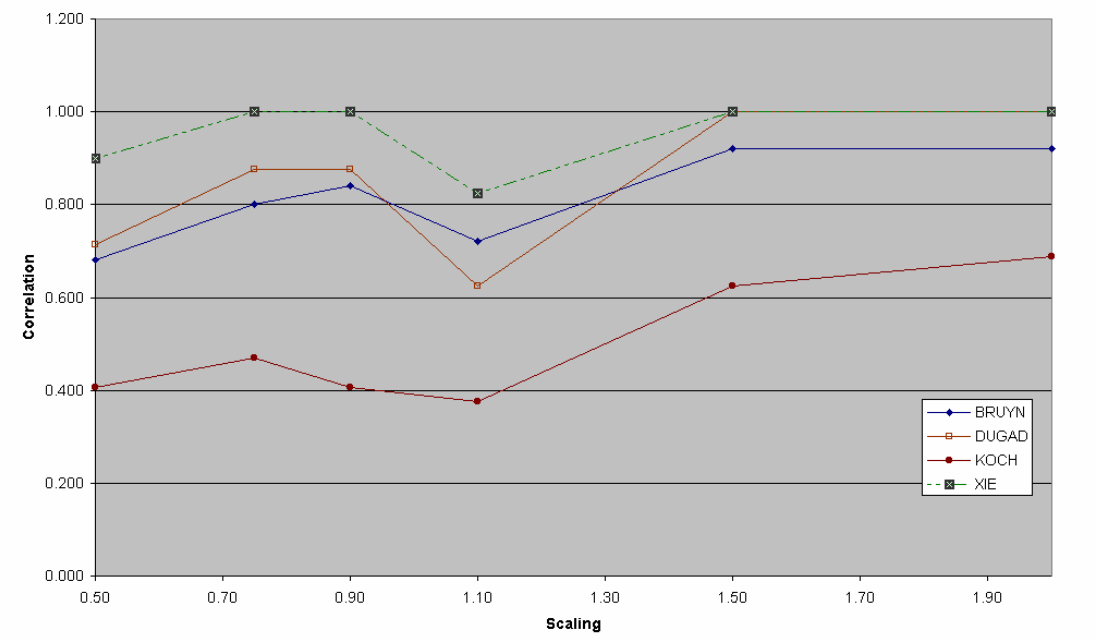


(b)

Figure 4.18 Group 1 test results for row and column removal attacks (a) non-blind algorithms (b) blind algorithms.

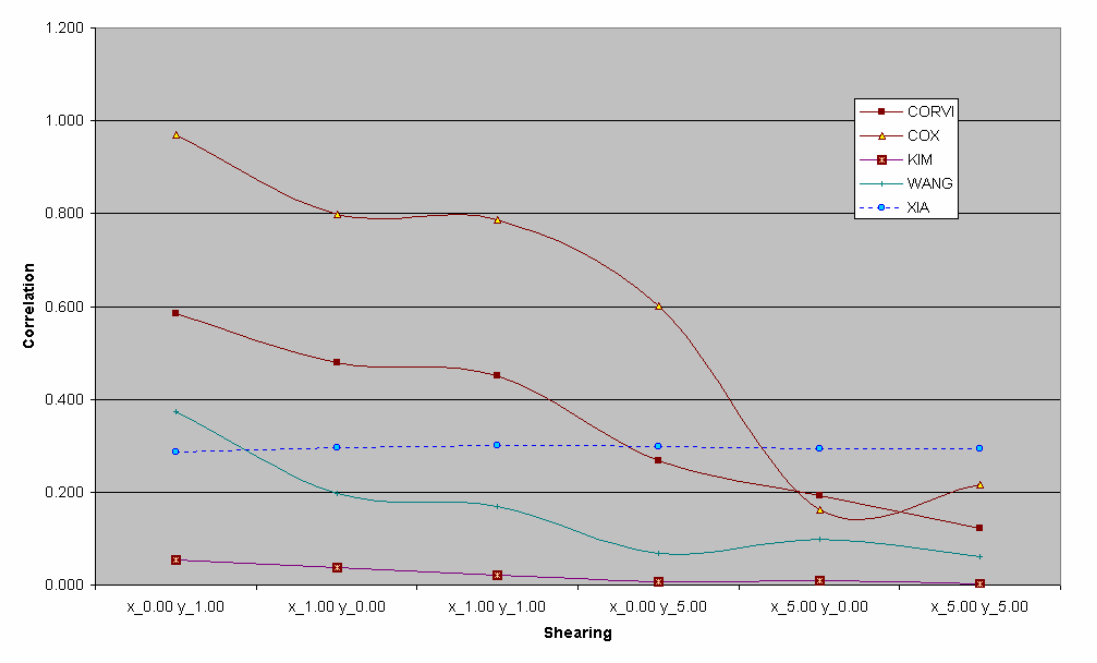


(a)

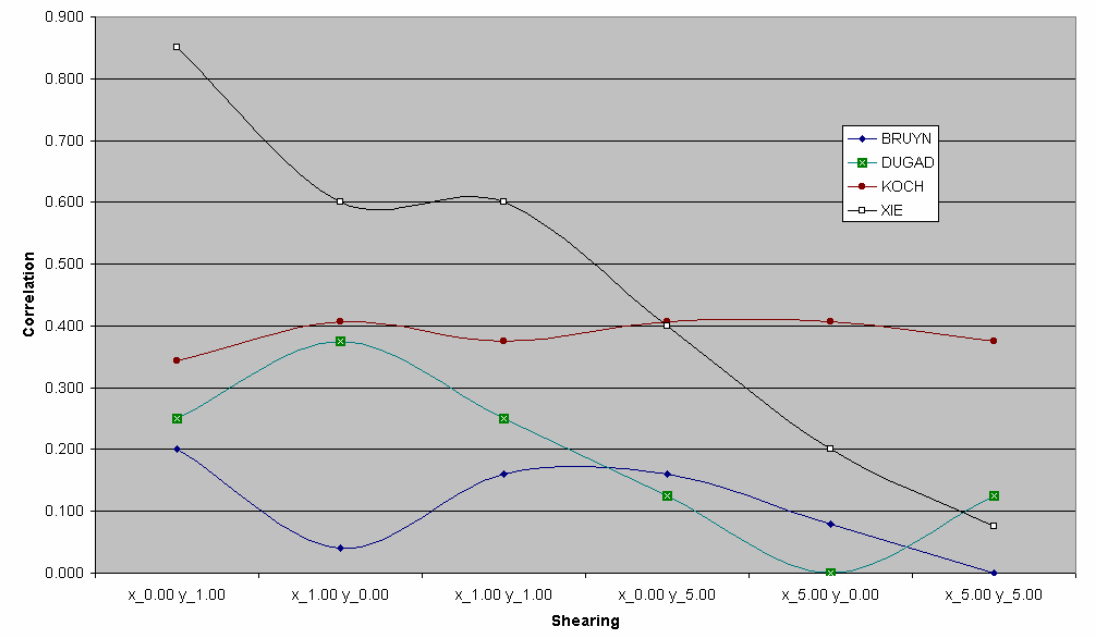


(b)

Figure 4.19 Group 1 test results for up-down scaling attacks (a) non-blind algorithms (b) blind algorithms.

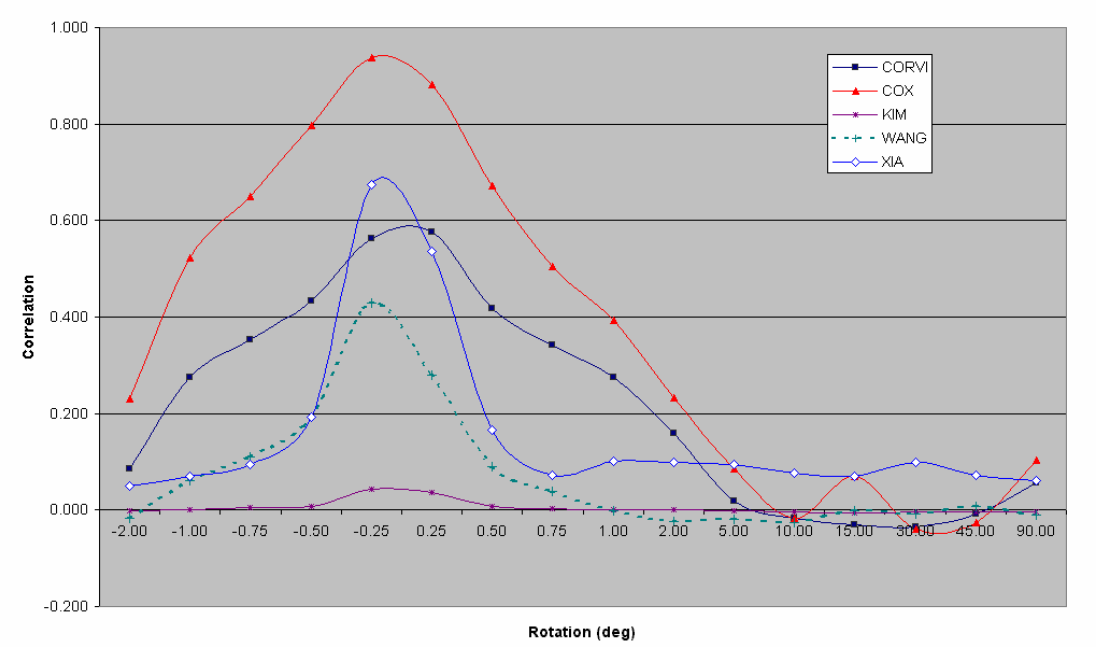


(a)

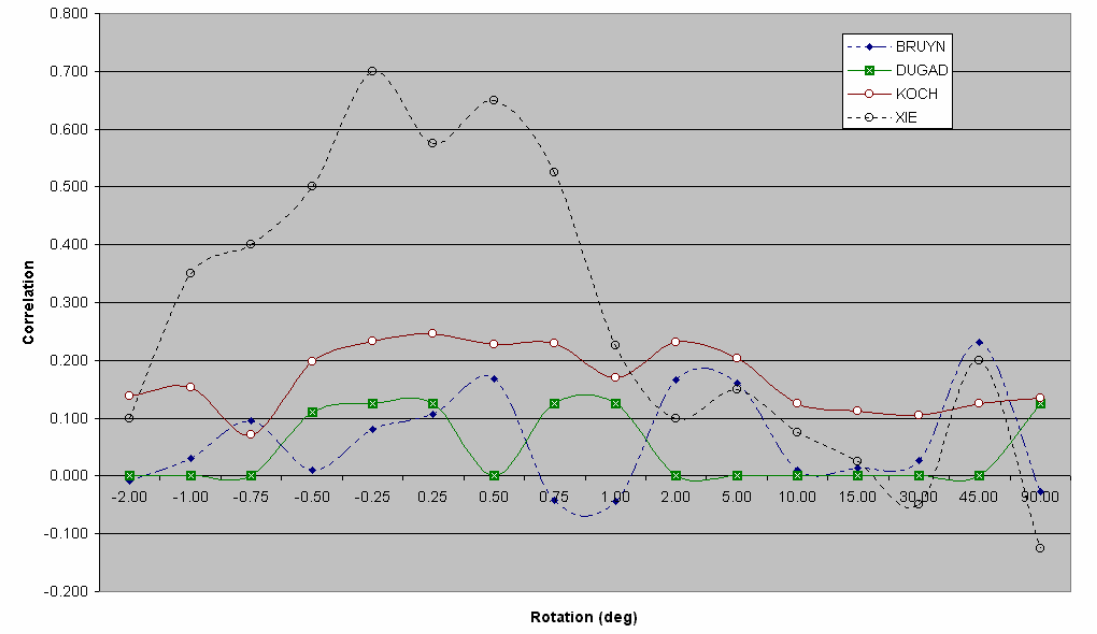


(b)

Figure 4.20 Group 1 test results for shearing attacks. The x-axis labels in the plot indicate the percentage of shearing in x and y direction. "x_0.00 y_5.00" means no shearing on x-direction and 5% shearing on y-direction (a) non-blind algorithms (b) blind algorithms.

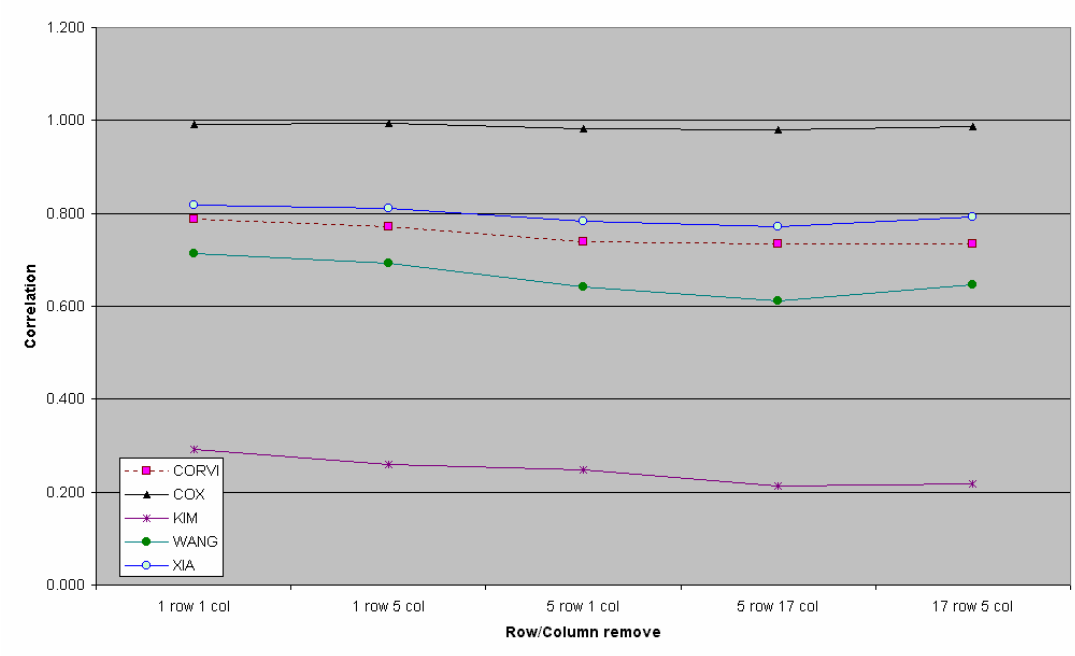


(a)

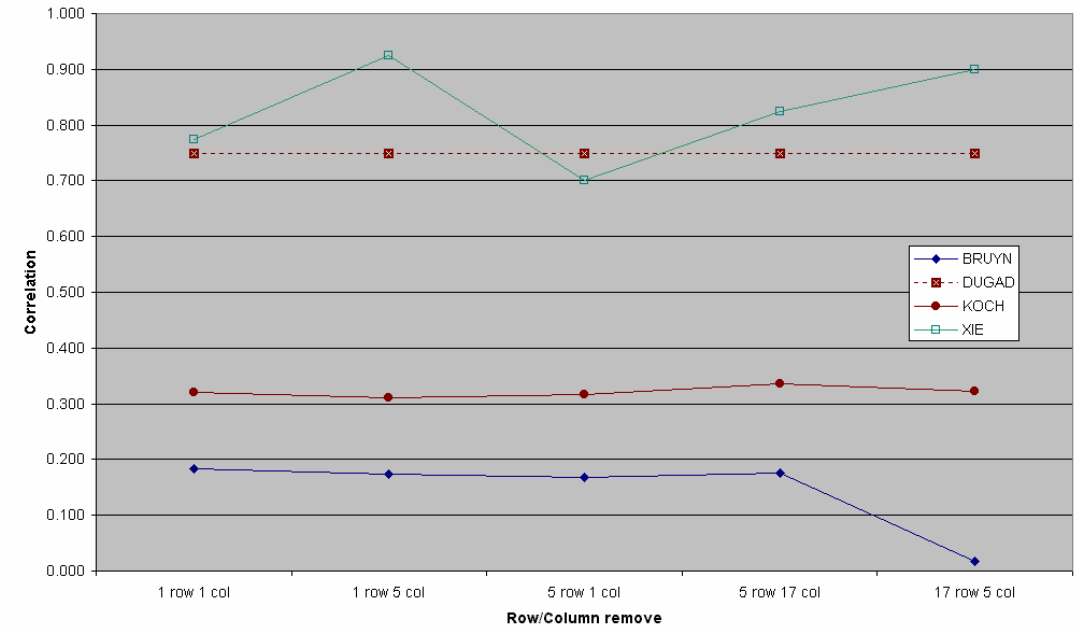


(b)

Figure 4.21 Group 2 test results for rotate and scale attacks with 16 different angles both clockwise and counter-clockwise (a) non-blind algorithms (b) blind algorithms.

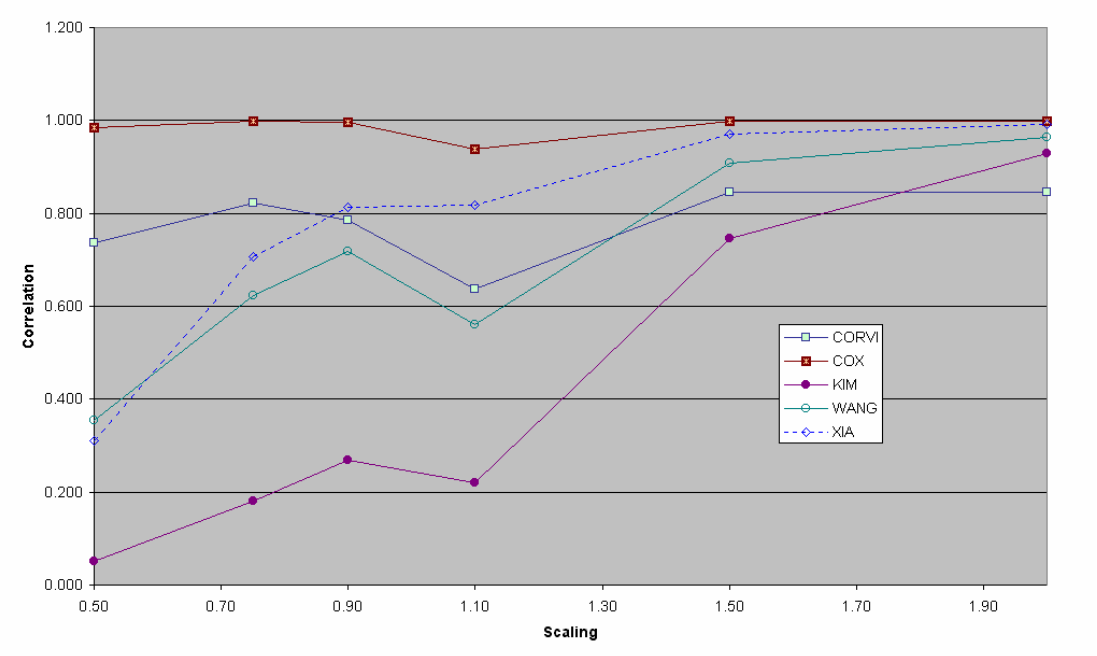


(a)

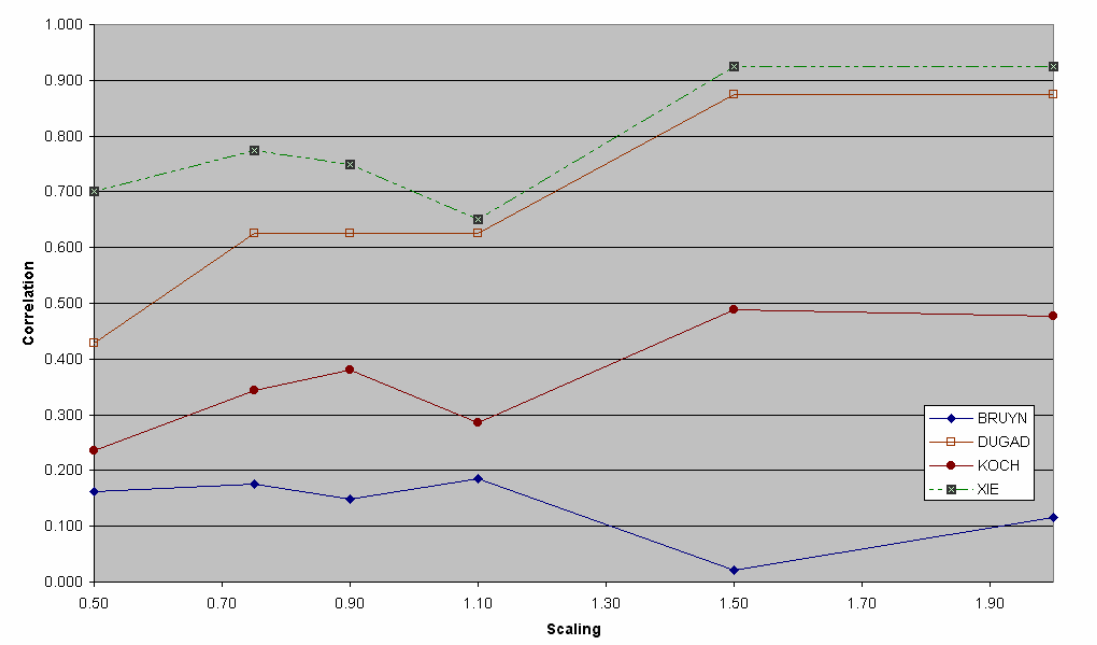


(b)

Figure 4.22 Group 2 test results for row and column removal attacks (a) non-blind algorithms (b) blind algorithms.

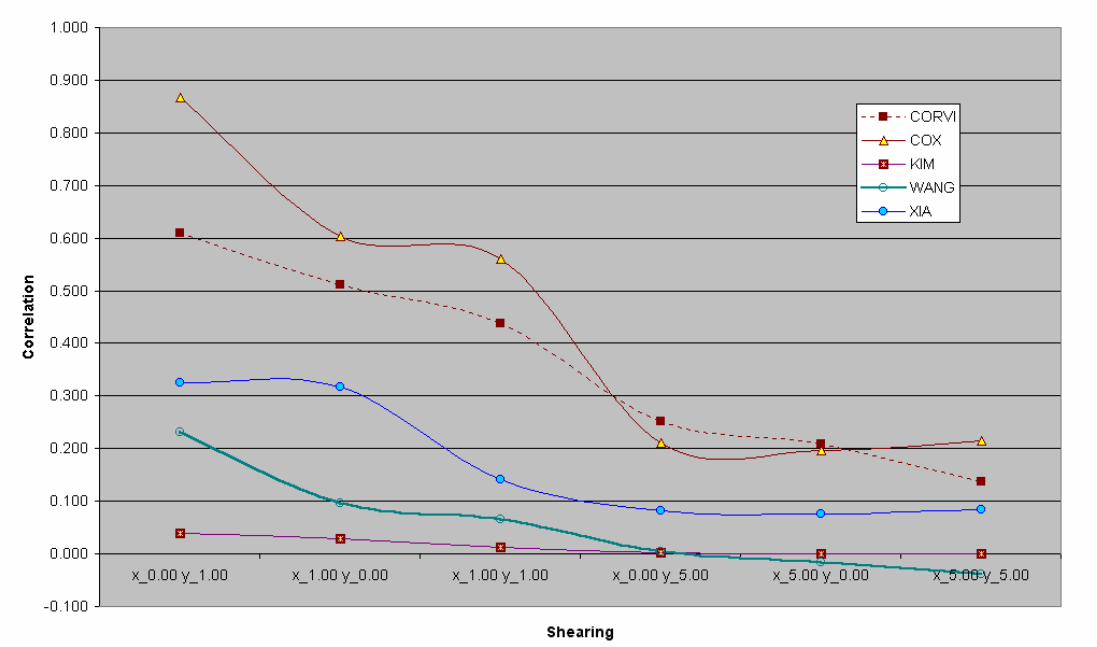


(a)

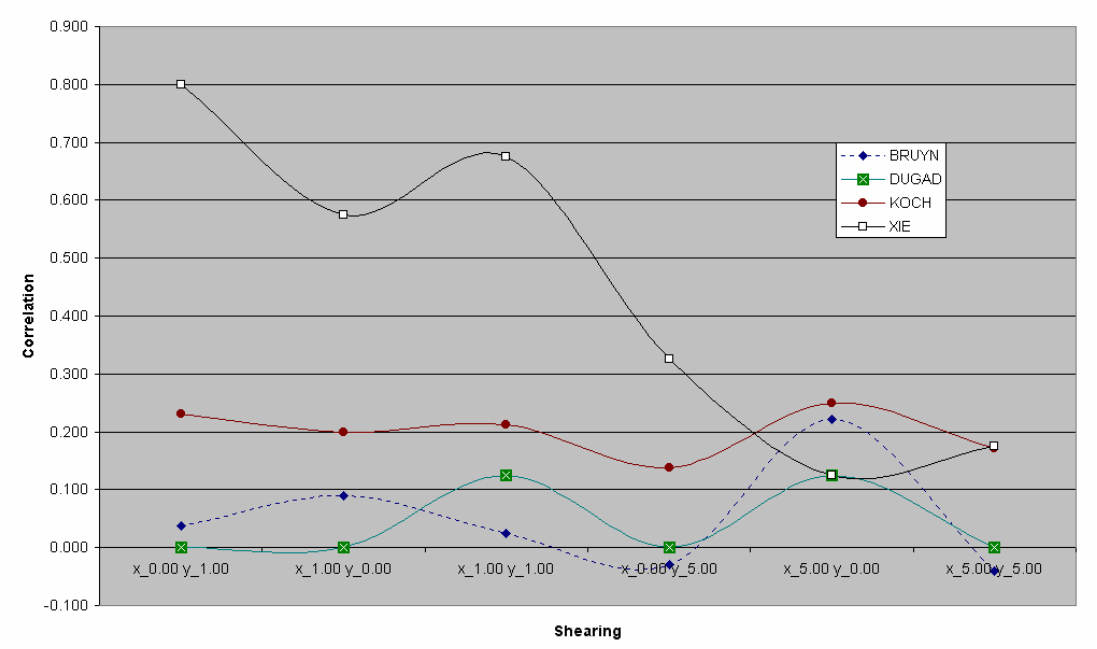


(b)

Figure 4.23 Group 2 test results for up-down scaling attacks (a) non-blind algorithms (b) blind algorithms.

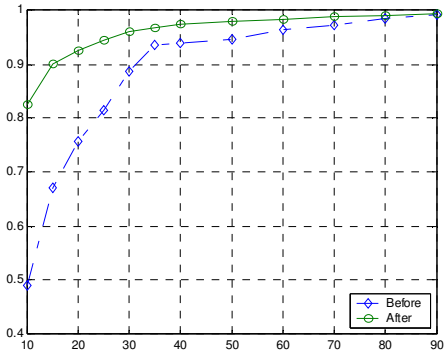


(a)

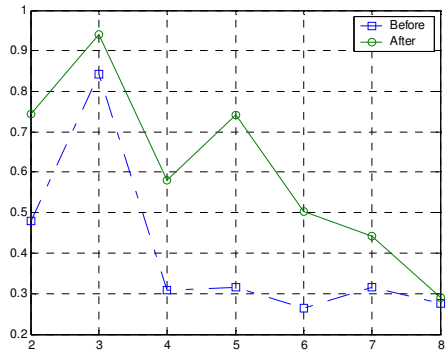


(b)

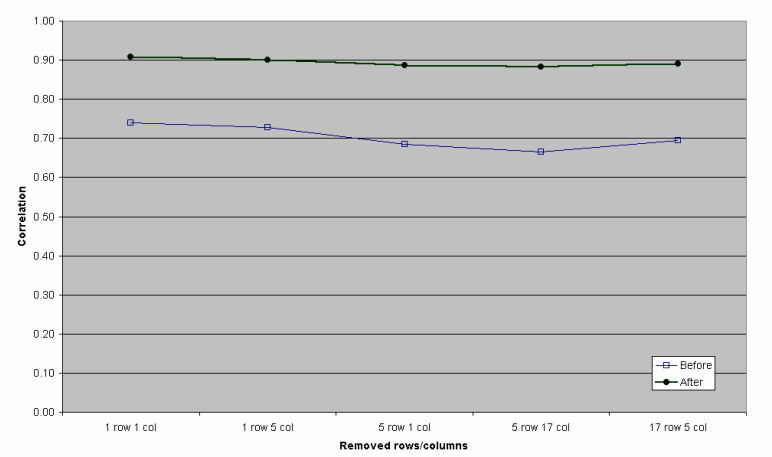
Figure 4.24 Group 2 test results for shearing attacks. The x-axis labels in the plot indicate the percentage of shearing in x and y direction. "x_0.00 y_5.00" means no shearing on x-direction and 5% shearing on y-direction (a) non-blind algorithms (b) blind algorithms.



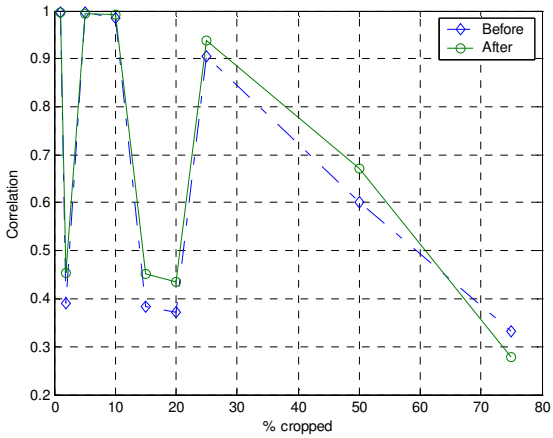
(a) JPEG quality vs. correlation



(b) Median filter size vs. correlation



(c)



(d)

Figure 4.25 The improvement on detection by increasing watermark power against (a) JPEG compression (b) median filtering (c) row and column removal (d) cropping.

Table 4.6 Test results for the spatial domain binary logo embedding technique.

Attack	% recovered pixels	Logo detectable?
Median Filtering		
2x2	88.6475	Yes
3x3	69.7754	Yes
4x4	58.7891	No
5x5	60.7668	Hardly detectable
6x6	55.6641	No
7x7	56.7139	No
8x8	53.9795	No
EZW Compression (bpp)		
1.50	98.7549	Yes
1.25	96.5332	Yes
1.00	91.3330	Yes
0.75	81.3232	Yes
0.50	70.5811	Yes
0.25	57.0801	No
0.20	54.2725	No
0.15	53.5645	No
JPEG Compression		
90	98.3887	Yes
80	93.4814	Yes
70	89.2090	Yes
60	84.1309	Yes
50	83.3984	Yes
40	86.5479	Yes
30	83.1543	Yes
20	69.0674	Yes
10	55.3955	No
Row-Column Removal		
1 row 1 col	94.9463	Yes
1 row 5 col	94.1406	Yes
5 row 1 col	93.9209	Yes
17 row 5 col	92.3096	Yes
5 row 17 col	92.7002	Yes
Gaussian Filter 3x3	92.4805	Yes
Sharpening Filter 3x3	98.9512	Yes

4.5. Conclusions

Among several watermarking algorithms available in the literature we picked 9 algorithms which are readily implemented and prepared a graphical environment for them that helps a user to perform various operations like watermark embedding, detection and Stirmark attacks easily. We have made extensive use of this program during the algorithm testing phase, since it automatically generates thousands of lines of DOS-type commands, executes them and sorts the outputs to dedicated folders and generates a result summary sheet viewable in MS-Excel, thereby creating a neat testing environment.

We used our program to make various experiments on the selected algorithms. First we watermarked a standard Lena image using these 9 algorithms. Each watermarked image is distorted (attacked) by the popular benchmarking software Stirmark that is capable of applying various attacks (linear, non-linear and geometrical) to create 78 attacked versions of each watermarked image.

The attacked images are fed to the watermark detector of each algorithm and we got the detection results for each attacked image to make a comparison with the other selected algorithm's results and evaluate their performance in terms of robustness.

For simple removal attacks like JPEG compression, median filtering, sharpening and Gaussian filtering most algorithms are successful to a great extent. However for geometrical attacks they can provide protection for a limited amount of distortion, since they do not have inherent mechanisms to withstand such attacks. Among all algorithms the method by Cox et al seems to be the most robust one. The availability of the original image at the detector and utilization of spread spectrum communication ideas together with significant coefficient selection are the major factor that makes that algorithm successful.

For removal attacks, the possibilities of improvement are basically in forms of increasing the watermark power. We have seen that most algorithms have a "room" for increased watermark power since with the default parameters proposed in

the papers, the fidelity measures are quite high. So that, some fidelity can be traded off for the sake of increased detection efficiency.

As the fidelity measure, we did not use the classical PSNR or MSE criteria since we have shown that they can be quite misleading. Instead, an image quality index technique is utilized, which credits the image fidelity in terms of perceptual figures like means or variances. Image quality index is actually an approximation that can be used instead of more complicated human visual system models like the Watson model. We have seen that DCT-based algorithms are causing much more distortion (both in terms of image quality index and PSNR/MSE criteria) on the original image than the DWT-based algorithms provided that we fix the number of coefficients modified. This result is a result of the fact that DWT domain watermarking algorithms have intrinsic perceptual masking characteristics whereas in DCT domain one has to apply a perceptual mask explicitly to take the perceptual properties of the image into account.

For geometrical attacks, we have seen that it is not possible to increase the detection capabilities significantly without finding a way of inverting the distortion before the detector operates on the attacked image.

REFERENCES

[1] T. Kalker, J-P. Linnartz, G. Depovere, and M. Maes, "On reliability of detecting electronic watermarks in digital images," in *Proc. IX European Signal Processing Conf.*, vol 1, Island of Rhodes, Greece, Sept. 8-11, 1998, pp. 13-16.

[2] E. Koch and J. Zhao. "Toward robust and hidden image copyright labeling," in *Proc. 1995 IEEE Workshop Nonlinear Signal and Image Processing,* North Marmaras, Greece, June 20-22, 1995, pp. 452-455.

[3] J. J. Quisquater, O. Bruyndonckx, and B. Macq, "Spatial method for copyright labeling of digital images," in *Proc. 1995 IEEE Workshop Nonlinear Signal and Image Processing,* North Marmaras, Greece, June 20-22, 1995, pp. 456-459.

[4] Pitas and T. H. Kaskalis, "Applying signatures on digital images," in *Proc. 1995 IEEE Workshop Nonlinear Signal and Image Processing,* North Marmaras, Greece, June 20-22, 1995, pp. 460-463.

[5] S. Craver, N. Memnon, B. L. Yeo, and M. Yeung, "Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks and implications," *IEEE Trans. On Selected Areas of Communications*, 16(4):573-586, 1998.

[6] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," *Proc. SPIE,* vol. 2420, pp. 40, 1995.

[7] G. L. Friedman, "The trustworthy digital camera: Restoring credibility to the photographic images," *IEEE Trans. Consumer Electronics,* vol. 39, no. 4, pp. 905-91, Nov. 1993.

[8] E. F. Hembrooke, "Identification of sound and like signals," *United States Patent,* (3,004,104), 1961.

[9] C. R. Abbey and H. H. Pursel, "Data channel monitor," *United States Patent,* (3,415,947), 1968.

[10] T. Ohsawa and M. Karita, "Automatic telecasting or radio broadcasting monitoring system," *United States Patent,* (3,760,275), 1973.

[11] M. G. Crosby, “Communications including submerged identification signal,” *United States Patent,* (3,845,391), 1974.

[12] D. E. H. and C. M. Solar, “Automatic monitor for programs broadcast,” *United States Patent,* (4,025,851), 1977.

[13] N. Komatsu and H. Tominaga, “Authentication system using concealed images in telematics,” *Memoirs of the School of Science and Engineering,* Waseda University, 52:45-60, 1998.

[14] R. Anderson, editor, “Information Hiding,” volume 1174 of *Lecture Notes in Computer Science,* pp. 39-48. Berlin; New York: Springer-Verlag, 1996.

[15] P. W. Wong and E. J. Delp, editors, *Security and Watermarking of Multimedia Contents, Proceedings of the Society of Photo-optical Instrumentation Engineers,* volume 3657, 2000.

[16] “SDMI Portable Device Specification,” Part 1, version 1.0, document number pdwg99070802, July 1999.

[17] G. Depovere, T. Kalker, J. Haitsima, M. Maes, L. de Strycker, P. Termont, J. Vandwege, A. Langell, C. Alm, P. Norman, B. O’Reilley, G. Howes, H. Vaanholt, R. Hintzen, P. Donnelly, and A. Hudson, “The VIVA project: Digital watermarking for broadcast monitoring,” *IEEE International Conference on Image Processing,* volume 1, pp. 430-434, 1998.

[18] G. W. Braudaway, K. A. Magerlein, and F. Mintzer, “Protecting publicly available images with a visible image watermark,” in *SPIE Conference on Optical Security and Counterfeit Deterrence Techniques*, volume 2659, pp. 126-133, 1996.

[19] S. Craver, N. Memon, B. L. Yeo, ans M. M. Yeung, “Can Invisible Watermarks Solve Rightful Ownerships?” *IBM Technical Report RC 20509,* IBM Research, July 1996. IBM Cyberjournal: *http://www.research.ibm.*

[20] Ingemar J. Cox, Matt L. Miller, and Jeffrey A. Bloom, “Digital Watermarking,” Morgan Kaufmann Publishers, 2002, ISBN 1-55860-714-5.

[21] Marco Corvi and Gianluca Nicchiotti, “Wavelet-based image watermarking for copyright protection,” in *Scandinavian Conference on Image Analysis SCIA '97*, Lappeenranta, Finland, June 1997.

[22] D. R. Stinson, “Cryptography: Theory and Practice,” Boca Raton, FL: CRC Press, 1995.

[23] D. J. Fleet and D. J. Heeger, “Embedding invisible information in colour images,” in *Proc. ICIP’97, IEEE Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997.

[24] S. Pereira, T. Pun, “Fast robust template matching for affine resistant watermarks,” in *Third International Information Hiding Workshop*, Dreseden, Germany, September 1999.

[25] F. Deguillaume, G. Csurka, T. Pun, “Countermeasures for unintentional and intentional video watermarking attacks,” in *IS&T/SPIE Electronic Imaging 2000*, San Jose, CA, USA, January 2000.

[26] M. Kutter, S. Voloshynovskiy, A. Herrigel, “Watermark copy attack,” in P.W. Wong, E.J. Delp (Eds.), *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II, SPIE Proceedings*, vol. 3971, San Jose, CA, USA, 23}28, January 2000.

[28] http://www.sigproc.eng.cam.ac.uk/~pl201/watermarking/index.html

[29] J. M. Barton, “Method and apparatus for embedding authentication information within digital data,” *US Patent* 5,646,997, 1997.

[30] D. Coppersmith, F. Mintzer, C. Tresser, C. W. Wu, and M. M. Yeung, “Fragile imperceptible digital watermark with privacy control,” in *Security and Watermarking of Multimedia Contents,* SPIE-3657, pp. 79-84, 1999.

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

[32] Rakesh Dugad, Krishna Ratakonda, and Narendra Ahuja, “A new wavelet-based scheme for watermarking images,” in *Proceedings of the IEEE International Conference on Image Processing, ICIP '98*, Chicago, IL, USA, October 1998.

[33] W. D. Moon, R. J. Weiner, R. A. Hansen, and R. N. Linde, “Broadcast signal identification system,” *United States Patent* 3,919,479, 1975.

[34] Jong Ryul Kim and Young Shik Moon, “A robust wavelet-based digital watermark using level-adaptive thresholding,” in *Proceedings of the 6th IEEE International Conference on Image Processing ICIP '99*, page 202, Kobe, Japan, October 1999.

[35] Borodin, R. Ostrovsky, and Y. Rabani, “Lower Bounds for High-Dimensional Nearest Neighbor Search and Related Problems,” in *Proceedings of the 31st ACM STOC,* pp. 473-480, 1999.

[36] Unzign watermark removal software, http://altern.org/watermark/, July 1997.

[37] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, “A video watermarking system for broadcast monitoring,” in *Proceedings of SPIE Electronic Imaging ’99, Security and Watermarking of Multimedia Contents,* San Jose, CA, Jan. 1999, pp. 103-112.

[38] C. I. Podilchuk and W. Zeng, “Image-adaptive watermarking using visual modals,” *IEEE Journal of Selected Areas in Communication*, 16(4):525-539, 1998.

[39] Competitive Media Reporting, http://www.cmr.com

[40] G. Depovere, T. Kalker, and J. P. Linnartz, “Improved watermark detection

using filtering before correlation," *Proc. 5th IEEE Int. Conf. Image Processing ICIP'98*, vol. I, Chicago, IL, Oct. 4-7 1998, pp. 430-434.

[41] http://www.confirmedia.com

[42] Juan R. Hernandez Martin and Martin Kutter, "Information retrieval in digital watermarking," *IEEE Communications Magazine*, August 2001, pp.110-1167.

[43] Liehua Xie and Gonzalo R. Arce, "Joint wavelet compression and authentication watermarking," in *Proceedings of the IEEE International Conference on Image Processing, ICIP '98*, Chicago, IL, USA, 1998.

[44] J. Ryan, "Method and Apparatus for Preventing the Copying of a Video Program," *United States Patent*, 4,907,093, 1990.

[45] Xiang-Gen Xia, Charles G. Boncelet, and Gonzalo R. Arce, "Wavelet transform based watermark for digital images," *Optics Express*, 3 pp. 497, December 1998.

[46] http://www.markany.com/eng/pro02.htm

[47] Houng-Jyh Wang, Po-Chyi Su, and C. C. Jay Kuo, "Wavelet-based digital image watermarking," *Optics Express*, 3 pp. 497, December 1998.

[48] A. B. Watson, "DCT quantization matrices optimized for individual images," *Human Visual, Visual processing, and Digital Display IV*, SPIE-1913:202-216, 1993.

[49] Emil Frank Hembrooke, "Identification of sound and like signals," *United States Patent*, 3,004,104, 1961.

[50] J. Cox, Matt L. Miller, "Electronic Watermarking: The First 50 Years," *Proceedings of the IEEE 2001 Int. Workshop on MultiMedia Signal Processing*, 2001.

[51] William M. Tomberlin, Louis G. MacKenzie, and Paul K. Bennett, "System for transmitting and receiving coded entertainment programs," *United States Patent*, 2,630,525, 1953.

[52] R. H. Baer, "Digital video modulation and demodulation system," *United States Patent*, 3,993,861, 1976.

[53] R. S. Broughton andW. C. Laumeister, "Interactive videomethod and apparatus," *United States Patent*, 4,807,031, 1989.

[54] Ray Dolby, "Apparatus and method for the identification of specially encoded FM stereophonic broadcasts," *United States Patent*, 4,281,217, 1981.

[55] http://www.digimarc.com/mediabridge/index.htm

[56] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. IEEE Int. Conference Image Processing*, vol. 2, Austin, TX, Nov. 1994, pp. 86-90.

[57] W. Bender, D. Gruhl, and N. Morimoto, "Techniques for data hiding," in *Proc.*

SPIE, Storage and Retrieval for Image and Video Databases III, vol. 2420, San Jose, CA, Feb. 9-10 1995, pp.165-173.

[58] Pitas and T. H. Kaskalis, "Applying signatures on digital images," in *Proc. IEEE Workshop on Nonlinear Signal and Image Processing*, Neos Marmaras, Thessaloniki, Greece, June 20-22, 1995, pp.460-463.

[59] G. Carroni, "Assuring ownership rights for digital images," in *Proc. Reliable IT Systems*, *VIC'95*, Germany, 1995, pp. 251-263.

[60] F. Hartung and Bern Girod, "Digital watermarking of raw and compressed video," in *Proc. SPIE 2952: Digital Compression Technologies and Systems for Video Communication*, Oct. 1996, pp. 205-213.

[61] Hanjalic, G. C. Langelaar, P. M. B. van Roosmalen, J. Biemond, and R. J. Lagendijk, *Image and Video Databases: Restoration, Watermarking and Retrieval* (Advances in Image Communications, vol. 8). New York: Elsevier Science, 2000.

[62] Pitas and T. H. Haskalis, "A method for signature casting on digital images," in *Proc. ICIP-96, IEEE Int. Conf. Image Processing*, vol. III, Lausanne, Switzerland, Sept. 15-17, 1996, pp.215-218.

[63] J. R. Smith and B. O. Comiskey, "Modulation and information hiding in images," in *Preproc. Information Hiding*, University of Cambridge, UK, May 1996.

[64] R. B. Wolfgang and E. J. Delp, "A watermark for digital images," in *Proc. ICIP-96, IEEE Int. Conf. Image Processing*, vol. III, Lausanne, Switzerland, Sept. 15-17, 1996, pp.219-222.

[65] R. B. Wolfgang and E. J. Delp, "A watermarking technique for digital imagery: Further studies," in *Proc. Int. Conf. Imaging Science, Systems and Technology*, Las Vegas, NV, June 30-July 3, 1997.

[66] G. C. Langelaar, J. C. A van der Lubbe, and R. L. Lagendijk, "Robust labeling methods for copy protection of images," in *Proc. SPIE Electronic Imaging'97, Storage and Retrieval for Image and Video Databases V*, San Jose, CA, Feb. 1997, pp. 298-309.

[67] W. Zeng and B. Liu, "On resolving rightful ownerships of digital images by invisible watermarks," in *Proc. ICIP-97, IEEE Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997, pp.552-555.

[68] J. Fridrich, "Robust bit extraction from images," in *Proc. IEEE ICMCS'99 Conf.*, Florence, Italy, June 7-11, 1999.

[69] R. B. Wolfgang and E. J. Delp, "Overview of image security techniques with applications in multimedia systems," in *Proc. SPIE Conf. Multimedia Networks: Security, Displays, Terminals, and Gateways*, vol. 3228, Dallas, TX, Nov. 2-5, pp. 297-308.

[70] R. B. Wolfgang and E. J. Delp, "Fragile watermarking using the VW2D

watermark," in *Proc. Electronic Imaging'99*, vol. 3657, San Jose, CA, Jan. 25-27, 1999, pp. 204-213.

[71] K. S. Ng and L. M. Cheng, "Selective block assignment approach for robust digital image watermarking," in *Proc. SPIE/IS&T Int. Conf. Security and Watermarking of Digital Multimedia Contents*, vol. 3657, San Jose, CA, Jan. 25-27, 1999, pp. 14-17.

[72] G.C. Langelaar, I. Setyawan, R.L. Lagendijk, "Watermarking Digital Images and Video Data: A state-of-the-art overview", *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20-46, 2000.

[73] J. Proakis, "Digital Communications," McGraw-Hill, 1983.

[74] Henry Beker and Fred Piper, "Cipher Systems: The Protection of Communications," Northwood Publications, 1982.

[75] M. Barni, C. I. Podilchuk, F. Bartolini, and E. J. Delp, "Watermark Embedding: Hiding a Signal Within a Cover Image," *IEEE Communications Magazine*, August 2001, pp.102-108.

[76] Joseph J. K. Ruanaidh and T. Pun, "Rotation, scale and translation invariant digital image watermarking," *Proc. ICIP'97, IEEE Int. Conf. Image Processing*, Santa Barbara, CA, Oct. 1997, pp. 536-539.

[77] N. Jayant, J. Johnston, and R. Safranek, "Signal compression based models of human perception," *Proceedings of the IEEE*, vol. 81, pp. 1385-1422, October 1993.

[78] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into a dithered multilevel image," *In IEEE Military Commun. Conf.*, pages 216-220, September 1990.

[79] M. Kobayashi, "Digital watermarking: Historical roots," Technical report, IBM Research, Tokyo Res. Lab, 1997.

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

[81] Seok Kang and Yoshinao Aoki, "Image Data Embedding System for Watermarking Using Fresnel Transform," *IEEE Int. Conf. on Multimedia Computing and Systems Volume-1*, June 7-11, 1999, Florence, Italy, p.9885.

[82] Patrick Bas, Jean-Marc Chassery and Franck Davoine, "Using the fractal code to watermark images," in *Proceedings of ICIP'98*, Vol. I, Chicago, Illinois, USA, 1998, pp. 469-473.

[83] A. Z. Tirkel, G. A. Rankin, R. G. Van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne, "Electronic watermark," in *Dicta-93*, pp. 666-672, Macquarie University, Sydney, December 1993

[84] A. Z. Tirkel, R. G. Van Schyndel, and C. F. Osborne, "A two-dimensional digital watermark," in *ACCV'95*, pp. 378-383, University of Queensland,

Brisbane, December 6-8, 1995.

[85] R. G. Van Schyndel, A. Z. Tirkel, and C. F. Osborne, "Towards a robust digital watermark," in *Dicta-95*, pp. 504-508, Nanyang Technological University, Singapore, December 5-8, 1995.

[86] I. J. Cox and M. L. Miller, "A review of watermarking and the importance of perceptual modeling," *Proceedings of SPIE, Human Vision & Electronic Imaging II,* vol. 3016, pp.92-99, 1997.

[87] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp, "Perceptual watermarks for digital images and video," *Proceedings of the IEEE,* vol. 87, no 7, pp. 1108-1126, 1999.

[88] P. W. Wong and E. J. Delp, eds, *Security and Watermarking of Multimedia Contents,* vol. 3657, San Jose, California, USA, 25-27 Jan. 1999, The Society for Imaging Science and Technology (IS&T) and the International Society for Optical Engineering (SPIE), SPIE, ISBN 0-8194-3128-1.

[89] M. Kutter and F. A. P. Petitcolas, "A fair benchmark for image watermarking systems," in Wong and Delp [87], pp.226-239, ISBN 0-8194-3128-1.

[90] J. Fridrich anf M. Goljan, "Comparing robustness of watermarking techniques," in Wong and Delp [87], pp. 214-225, ISBN 0-8194-3128-1.

[91] John Wiley, L.A. Olzak, and J.P. Thomas, "Handbook of perception and human performance," *Volume 1: Sensory Processes and Perception. Chapter 7: Seeing Spatial Patterns,* University of California, Los Angeles, California, 1986.

[92] Sviatoslav Voloshynovskiy, Shelby Pereira, and Thierry Pun, "Attacks on digital watermarks: Classification, Estimation-based attacks, and benchmarks," *IEEE Communcations Magazine,* August 2001.

[93] Sviatoslav Voloshynovskiy, Shelby Pereira, Alexander Herrigel, Nazanin Baumgärtner, and Thierry Pun, "Generalized watermark attack based on watermark estimation and perceptual remodulation," In Ping Wah Wong and Edward J. Delp eds., *IS&T/SPIE's 12th Annual Symposium, Electronic Imaging 2000: Security and Watermarking of Multimedia Content II*, Vol. 3971 of SPIE Proceedings, San Jose, California USA, 23-28 January 2000. (Paper EI 3971-34).

[94] J. J. K. Ruanaidh and Shelby Pereira, "A Secure robust digital image watermark," in *Electronic Imaging: Processing, Printing and Publishing in Colour*, SPIE Proceedings, Zürich, Switzerland, May 1998. (SPIE/IST/Europto Symposium on Advanced Imaging and Network Technologies)

[95] V. Darmstaedter, J. F. Delaigle. J. J. Quisquater, B. Macq, "Low cost spatial watermarking,"

[96] Fabien A. P. Petitcolas, Ross J. Anderson, Markus G. Kuhn, "Attacks on copyright marking systems," in David Aucsmith (Ed), *Information Hiding,*

Second International Workshop, IH'98, Portland, Oregon, USA, April 15-17, 1998, Proceedings, LNCS 1525, Springer-Verlag, ISBN 3-540-65386-4, pp. 219--239.

[97] Martin Kutter and Fabien A. P. Petitcolas, "A fair benchmark for image watermarking systems," To in E. Delp et al. (Eds), in vol. 3657, *Proceedings of Electronic Imaging '99, Security and Watermarking of Multimedia Contents*, San Jose, CA, USA, 25-27 January 1999. The International Society for Optical Engineering.

[98] http://anchovy.ece.utexas.edu/~zwang/research/quality index/demo.html

[99] G. R. Arce and L. Xie, "A blind content based digital image signature," *Proceedings of the 2nd Annual Fedlab Symposium on ATIRP*, Feb. 1998.

[100] G. R. Arce and L. Xie, "A blind wavelet based digital signature for image authentication," *Proceedings of the EUSIPCO-98*, Sept. 1998.

[101] T. Ebrahimi, S. Winkler and E. Drelie Gelasca, "Perceptual Quality Assessment for Video Watermarking," in Watermarking Quality Evaluation Special Session at *ITCC, International Conference on Information Technology: Coding and Computing*, Las Vegas, USA, April 8-10, 2002

[102] A. Z. Tirkel, "Image and watermark registration," submitted to *Signal Processing*, January 1997.

[103] I. J. Cox, F. T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," in *Proceedings of the IEEE ICIP '97*, vol. 6, pp.1673-1687, Santa Barbara, California, USA, 1997.

[104] A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "DCT-based watermark recovering without restoring to the uncorrupted original image," in *International Conference in Image Processing*, vol. III, pp. 520-523, 1997.

[105] Houng-Jyh Wang, Yi-Liang Bao, C. C. Jay Kuo, and Homer Chen, "Multi-threshold wavelet codec (MTWC)," *Technical Report*, Dept of Electrical Engineering, University of Southern California, Los Angeles, CA, USA, March 1998.

[106] Houng-Jyh Wang and C. C. Jay Kuo, "High fidelity image compression with multi-threshold wavelet coding (MTWC)," in *SPIE's Annual Meeting – Application of Digital Image Processing*, San Diego, CA, USA, August 1997.

[107] F. Bartolini, M. Barni, V. Cappellini, and A. Piva, "Mask building for perceptually hiding frequency embedded watermarks," in *Proc. 5th IEEE Intl. Conf. Image Processing ICIP'98*, vol I, Chicago, IL, Oct. 4-7, 1998, pp. 450-454.

[108] Jian Zhao and Eckard Koch, "Embedding robust labels into images for copyright protection," in *Proceedings of the International Congress on Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, pp. 242-251, Vienna, Austria, August 1995.

[109] G. Langelaar, R. Lagendijk, and J. Biemond, "Removing spatial spread spectrum watermarks by non-linear filtering", *in Proc. EUSIPCO98*, **4**, pp. 2281-2284, 1998.

[110] R. L. Pickholtz, D. L. Schilling, and L. B. Millstein, "Theory of spread spectrum communications – a tutorial," *IEEE Transactions on Communications* vol. COM-30, no-5, pp. 855-884, May 1982.

[111] S. Mallat, "Multiresolution approximations and wavelet orthonormal bases of $L^2(R)$," *Trans. Amer. Math. Soc.*, 315, 69-87 (1989).

[112] I. Daubechies, "Orthonormal bases of compactly supported wavelets," *Comm. on Pure and Appl. Math.*, 41, 909-996 (1988).

[113] O. Rioul and M. Vetterli, "Wavelets and signal processing," *IEEE Signal Processing Magazine*, 14-38, (1991).

[114] I. Daubechies, "Ten Lectures on Wavelets," *SIAM*, Philadelphia, 1992.

[115] M. Vetterli and J. Kovacevic, "Wavelets and Subband Coding," Prentice Hall, Englewood Cliffs, NJ, 1995.

[116] J. Shapiro, "Embedded image coding using zerotrees of wavelet coeffcients," *IEEE Trans. on Signal Processing*, 41, 3445-3462 (1993).

[117] http://www.lnt.de/~su/research2-e.html

[118] Xiang-Gen Xia, Charles G. Boncelet and Gonzalo R. Arce, "Wavelet transform based watermark for digital images," *Optics Express* Vol. 3, No. 12.

[119] J. J. K. O Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," in *IEEE International Conference on Image Processing*, Vol. 3, pp. 239-242, Lausanne, Switzerland, September 1996.

[120] Peter Meerwald, "Digital Image Watermarking in the Wavelet Transform Domain," MS Thesis, University of Salzburg, 2001.

[121] J. Fridrich and M. Goljan, "Protection of digital images using self embedding," in *Proc. Symp. Content Security and Data Hiding in Digital Media*, New Jersey Institute of Technology, Mar. 16, 1999.

[122] A. B. Watson, editor, "Digital images and human vision," Cambridge, MA:MIT Press, 1993.

[123] B. Girod, "What is wrong with mean-squared error?" in A. B. Watson, editor, *Digital images and human vision*, Chapter 15, pp. 207-220, Cambridge, MA:MIT Press, 1993.

[124] A. R. Prasad, R. Esmailzadeh, S. Winkler, T. Ihara, B. Rohani, B. Pinguet and M. Capel, "Perceptual Quality Measurement and Control: Definition, Application and Performance,"

[125] W. B. Pennebaker and J. L. Mitchell, *The JPEG Still Image Compression Standard*, New York: Van Nostrand, 1993.

[126] Zhou Wang and Alan C. Bovik, "A Universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, March 2002.

[127] Zhou Wang, Alan C. Bovik, and Ligang Lu, "Why is image quality assessment so difficult?,"

[128] C. J. van den Branden Lambrecht and J. E. Farrell, "Perceptual quality metric for digitally coded colour images," *Proceedings of EUSIPCO*, pp. 1175-1178, 1996.

[129] Stefan Winkler, "A Perceptual distortion metric for digital colour images," *Proceedings of the 5th IEEE International Conference on Image Processing (ICIP'98)*, vol. 3, pp. 399-403, Chicago, USA, October 4-7, 1998.

APPENDIX A

BACKGROUND MATERIAL

A.1. Spread Spectrum Watermarking [116]

Spread spectrum was originally developed for military use (radar and communications) in several countries. Since its declassification, it has found civilian application, particularly in code-division multiple-access (CDMA) communications, the system used for cellular telephony.

In spread spectrum, a narrowband signal (the message to be transmitted) is modulated by a broadband carrier signal, which broadens (spreads) the original, narrowband spectrum; hence the term "spread spectrum." The following properties of spread spectrum are particularly well-suited for watermarking:

- **Antijamming (AJ)**

The *antijamming* (AJ) property results from the fact that an attaker does not know the privileged information that the sender and an authorized receiver possess. As a result, the attacker must jam the entire spectrum of the broadband signal. The jammer has limited power, however, so it can only jam each frequency with low power. Hence, the sender and receiver have an effective signal-to-jammer advantage (called the *processing gain*).

With application to watermarking, the AJ property means that, in order to jam a watermark, an attacker must distort the marked media severely – so severely that the attacked media is no longer of acceptable quality or has no commercial value.

- **Low probability of intercept (LPI)**

The *low probability of intercept* (LPI) property is a consequence of spreading: a large signal power is distributed over the entire frequency spectrum, so only a small amount of power is added at each frequency. Often, the increase is below the noise floor, so an attacker may not even detect the transmission of a spread-spectrum signal.

In watermarking, the LPI property allows a watermark to be embedded unobtrusively. The power of a watermark at any frequency can be made very small if sufficient spreading is possible. In this way, the marked media can be made imperceptible, which is a desired feature for media such as audio, images, and video.

- **Pseudo-noise (PN)**

For security, the carrier is often a *pseudo-noise* (PN) signal, meaning that it has statistical properties similar to those of a truly random signal, but it can be exactly regenerated with knowledge of privileged information. For example, the carrier could be the output of a random-number generator that has been initialized with a particular seed, and the seed is known only to the owner.

The PN property is useful for watermarking, because it makes it difficult for an attacker to estimate the watermark from marked media. In addition, with properly chosen PN signals, even if the attacker can perfectly estimate some small segments of the watermark, it is not possible to determine the rest of the mark.

A.2. Discrete Wavelet Transform [117]

The wavelet transform has been extensively studied in the last decade, see for example [110-115]. Many applications of wavelet transforms have been found such as compression, detection, and communications. There are many excellent tutorial books and papers on these topics. Here, we introduce the necessary concepts of the DWT for the purposes of this text.

The basic idea in the DWT for a one dimensional signal is the following. A signal is split into two parts, usually high frequencies and low frequencies. The edge

components of the signal are largely confined to the high frequency part. The low frequency part is split again into two parts of high and low frequencies. This process is continued an arbitrary number of times, which is usually determined by the application at hand. Furthermore, from these DWT coefficients, the original signal can be reconstructed.

This reconstruction process is called the inverse DWT (IDWT). The DWT and IDWT can be mathematically stated as follows. Let

$$H(w)=\sum_k h_k e^{-jk\omega} \qquad , \qquad G(w)=\sum_k g_k e^{-jk\omega}$$

be a lowpass and a highpass filter, respectively, which satisfy a certain condition for reconstruction to be stated later. A signal, $x[n]$ can be decomposed recursively as

$$c_{j-1,k}=\sum_n h_{n-2k}c_{j,n}$$
$$d_{j-1,k}=\sum_n g_{n-2k}c_{j,n}$$

for $j=J+1,J,\cdots J_0$ where $c_{J+1,k}=x[k],k\in Z,J+1$ is the high resolution level index, and J_0 is the low resolution level index. The coeffcients $c_{J_0,k},d_{J_0,k},d_{J_0+1,k},\cdots,d_{J,k}$ are called the DWT of signal $x[n]$, where $c_{J_0,k}$ is the lowest resolution part of $x[n]$ and $d_{j,k}$ are the details of $x[n]$ at various bands of frequencies. Furthermore, the signal $x[n]$ can be reconstructed from its DWT coeffcients recursively:

$$c_{j,n}=\sum_k h_{n-2k}c_{j-1,k}+\sum_k g_{n-2k}d_{j-1,k}$$

The reconstruction formula above is called the IDWT of $x[n]$. To ensure the above IDWT and DWT relationship, the following orthogonality condition of the

filters $H(\omega)$ and $G(\omega)$ is needed:

$$|H(\omega)|^2 + |G(\omega)|^2 = 1$$

An example of such a pair is the Haar wavelet filters given by:

$$H(\omega) = \frac{1}{2} + \frac{1}{2}e^{-j\omega} \qquad G(\omega) = \frac{1}{2} - \frac{1}{2}e^{-j\omega}$$

The DWT and IDWT for two dimensional signals $x[m, n]$ can be similarly defined by implementing the one dimensional DWT and IDWT for each dimension m and n separately: $DWT_n[DWT_m[x[m, n]]]$.

A.3. Different Forms of Correlation

A.3.1 Linear Correlation

The linear correlation between two vectors **v** and **w** is the average product of their elements formulated as follows:

$$Z_{LC}(\mathbf{v}, \mathbf{w}) = \frac{1}{N}\sum_i \mathbf{v}[i]\,\mathbf{w}[i]$$

Watermarking-wise we can think of **v** as the received signal (suspected to be watermarked) and **w** as the test (reference) pattern whose presence is sought in **v**.

For the linear correlation to be high, the two vectors must be similar to each other and the maximum value is obtained if they are exactly the same. In communications, this idea of detecting a pattern in a signal by convolving the signal with that search pattern itself is called *matched filtering* and it is the optimum way of detecting signals in the presence of additive, white Gaussian noise.

A.3.2. Normalized Correlation

Normalized correlation is a more "intelligent" version of linear correlation. Linear correlation values are highly dependant on the magnitudes of the elements of the vectors involved in the operation. In the watermark case, this means that the watermark will not be robust against brightness changes in an image or volume difference in a watermarked audio playback.

This problem is eliminated by normalizing the vectors before the correlation. This means we use $\tilde{\mathbf{v}}$ and $\tilde{\mathbf{w}}$ instead of **v** and **w**, where;

$$\tilde{\mathbf{v}} = \frac{\mathbf{v}}{|\mathbf{v}|} \text{ and } \tilde{\mathbf{w}} = \frac{\mathbf{w}}{|\mathbf{w}|}.$$

Then, normalized correlation is calculated as follows:

$$Z_{NC}(\mathbf{v}, \mathbf{w}) = \sum_i \tilde{\mathbf{v}}[i]\tilde{\mathbf{w}}[i]$$

Since we also know that the inner product of two vectors is equal to the product of their Euclidian lengths and the cosine of the angle between them, we can find that the normalized correlation between two vectors is just the cosine of the angle between them.

It is sometimes suggested that linear correlation can be used as a detection measure, but that the detector's threshold should be scaled by the magnitude of the extracted mark. This is equivalent to the use of a normalized correlation detection measure.

A.3.3. Correlation Coefficient

Correlation coefficient is obtained by subtracting out the means of the two vectors before computing the normalized correlation between them. That is:

$$\left.\begin{aligned}\hat{\mathbf{v}} &= \mathbf{v} - \overline{\mathbf{v}} \\ \hat{\mathbf{w}} &= \mathbf{w} - \overline{\mathbf{w}}\end{aligned}\right\} \quad Z_{CC}(\mathbf{v},\mathbf{w}) = Z_{NC}(\hat{\mathbf{v}},\hat{\mathbf{w}}).$$

This provides robustness against changes in the DC term of the work, such as the addition of a constant intensity to all pixels of an image.

Geometrically, correlation coefficient between two vectors in *N*-space is just normalized correlation between the two after projection into an (*N*-1)-space. This is because the mean vector of **v** describes the point on the diagonal of the coordinate system that lies closest to **v**. Thus, the result of the subtraction is a vector that is orthogonal to the diagonal. This means that the resulting vector lies in an (*N*-1)-space orthogonal to the diagonal of the *N*-dimensional coordinate system. Because of this relationship the two measures can be used interchangeably.