

Digital Watermarking using Complex Wavelets

*A dissertation submitted to the University of Cambridge
for the degree of Doctor of Philosophy*

Patrick Loo (Trinity College)

March 2002



Signal Processing and Communications Laboratory
Department of Engineering
University of Cambridge

To My Parents

DECLARATION

The research described in this dissertation was carried out by the author between October 1998 and March 2002. Except as indicated in the text, the contents are entirely original and are not the results of work done in collaboration. No part of this dissertation has been submitted to any other university. This thesis contains less than 60 figures and its length does not exceed 40,000 words.

Patrick Loo

ACKNOWLEDGEMENTS

First of all, I am in debt to Dr. Nick Kingsbury for his invaluable advice and support over the last three years. I would also like to thank the members of the signal processing group for providing a friendly working environment. Thanks are also given to other researchers working in the field of digital watermarking around the world for the helpful discussions with them. In particular, I would like to thank the following people:

- Dr. Fabien Petitcolas for various discussions about his work and proofreading my thesis.
- Peter Meerwald for a lot of discussions on the weaknesses of various watermarking techniques and proofreading my thesis.
- Joachim Eggers for helpful discussions and insights on quantisation based watermarking schemes.
- Dr. Jonathan Su for providing the matlab codes to estimate the parameters of the generalised Gaussian distribution.
- Dr. Deepa Kundur for discussions of her watermarking algorithms and providing me with an office during my visit to Toronto.
- Dr. Julian Magarey and Dr. Jonathan Carr for their help with motion estimation using complex wavelets and interpolation using radial basis functions respectively.

Last but not least, I thank my parents. Without their encouragement, I would not have gotten this far.

This work was made possible through the Internal Graduate Studentship generously provided by Trinity College, Cambridge. The author would like to thank Trinity College and the Department of Engineering for their financial assistance towards the travelling expenses during the course of his research.

KEYWORDS

Digital Watermarking, Information Hiding, Complex Wavelets, Image Registration, Communication with Side Information, Watermark Detection

ACRONYMS

AWGN	additive Gaussian white noise
BER	bit error rate
BPP	bit per pel
CWT	Complex Wavelet Transform
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DSP	Digital Signal Processing
DT CWT	Dual Tree Complex Wavelet Transform
DWT	Discrete Wavelet Transform
ECC	error control code
GG	generalised Gaussian
HVS	Human Visual System
JND	just noticeable difference
JPEG	Joint Photographic Expert Group
KLT	Karhunen-Loeve Transform
LOD	locally optimal detector
ML	maximum likelihood
MPEG	Moving Picture Expert Group
PN	pseudo-random noise
PSNR	peak signal to noise ratio
QIM	Quantisation Index Modulation
RBF	radial basis function
RMS	root mean square
ROC	receiver operation characteristics
SCS	Scalar Costa Scheme
SNR	signal to noise ratio
SS	spread spectrum
ST	spread transform
UDWT	Undecimated Discrete Wavelet Transform

Abstract

Digital watermarks recently emerged as a possible solution for protecting the copyright of digital materials. The work presented in this thesis is concerned with the design of robust watermarking algorithms with complex wavelets. We choose the complex wavelet transform as our watermarking domain because it is a relatively new transform and has useful properties for image processing applications. In addition, experimental results show that watermarking systems designed using complex wavelets have good performance. We first give an overview of the watermarking problem, example applications of watermarks as well as a comprehensive survey of current watermarking techniques and attacks. The complex wavelet transform is then described and its advantages over the conventional real wavelet transform are highlighted. A human visual model in the complex wavelet domain is developed which forms the basis in the design of our watermarking algorithms.

Two different watermarking systems are considered in this thesis. The first one is based on the principles of spread spectrum communications. We discuss the precautions required when designing a watermarking algorithm in a redundant domain like the complex wavelets. An alternative matched filter is proposed for watermark decoding which is shown to have better performance than the conventional matched filter when the underlying signal is non-stationary. The issue of watermark detection is addressed and we propose a method of using the watermark itself for detection, which removes the need for a separate reference watermark. We also suggest ways in which the watermark decoder performance can be improved when the watermarked image is attacked. In particular we consider three scenarios: compression, geometric distortion and denoising. Compression and denoising are shown to have similar effects on a watermarked image and we can use the same method for decoding. An image registration algorithm based on motion estimation is developed to combat geometric distortion attacks.

Although spread spectrum based watermarks are quite robust, the host image acts as interference with respect to the watermark decoder. We consider watermarking as a communication process with side information, where the knowledge of the host image is exploited at the embedder to reduce the host interference. Our second watermarking algorithm is based on a hybrid combination between quantisation and spread spectrum techniques and we compare it with our spread spectrum based system. Finally we close with conclusions and suggest future research directions.

Contents

1	Introduction	1
1.1	What is watermarking?	1
1.1.1	Definition	1
1.1.2	Relationship of watermarking to steganography and compression	2
1.1.3	Watermarking terminology	3
1.2	Basic watermarking systems	4
1.3	Watermarking applications	5
1.3.1	Copyright protection	5
1.3.2	Copy protection	6
1.3.3	Fingerprinting for pirate tracing	6
1.3.4	Watermarking for authentication	6
1.4	Algorithm design issues	7
1.4.1	Imperceptibility, robustness and capacity	7
1.4.2	Invertibility and resolving rightful ownership	8
1.4.3	Public vs. private watermarking	8
1.5	Thesis overview	8
1.6	Notation conventions	9
2	Survey of Watermarking Techniques and Attacks	11
2.1	Introduction	11
2.2	Choice of embedding locations	12
2.3	Watermarking domain	13
2.3.1	Discrete Fourier Transform	14
2.3.2	Discrete Cosine Transform	14
2.3.3	Discrete Wavelet Transform	14
2.3.4	Translation, scale, rotation invariant domain	15
2.3.5	Key dependent domain	15
2.3.6	Other domains	15

2.3.7	On the choice of transform	16
2.4	Encoding of payload	16
2.4.1	Spread spectrum	16
2.4.2	Error control codes	17
2.5	Formation of the composite signal	20
2.6	Watermark extraction	21
2.7	Attacks and benchmarks	22
2.7.1	Removal based attacks	22
2.7.2	Geometric attacks	22
2.7.3	Cryptographic-like attacks	23
2.7.4	Protocol attacks	23
2.7.5	Benchmarking	23
2.8	Chapter summary	24
3	The Complex Wavelet Transform and the Human Visual System	25
3.1	Introduction	25
3.2	The Complex Wavelet Transform and its dual tree implementation	25
3.3	The Human Visual System and its relationship with the CWT	31
3.3.1	Contrast sensitivity	31
3.3.2	Masking	33
3.3.3	Visual model in the CWT domain	33
3.3.4	Image quality metric based on the HVS	34
3.4	Empirical results for the CWT visual model	36
3.4.1	Finding C_{T_0} and the correcting factor β	36
3.4.2	Determining k	36
3.4.3	Comparison with other domains	37
3.5	Chapter summary	40
4	Watermark Embedding and Extraction	43
4.1	Introduction	43
4.2	The watermark model and assumptions	44
4.3	Watermark embedding	44
4.3.1	Implication of CWT redundancy	44
4.3.2	Embedding algorithm	46
4.4	Watermark decoding	49
4.4.1	Simple correlator	50
4.4.2	Matched filter	52

4.4.3	Modified matched filter	53
4.4.4	Multiple channel decoding	54
4.5	Comparison with other domains	57
4.5.1	Simulation results	59
4.5.2	Discussion	65
4.6	On watermark detection	68
4.7	Chapter summary	71
5	Reliable Decoding of Watermarks after Attacks	73
5.1	Introduction	73
5.2	Watermark decoding after compression	74
5.2.1	Effects of compression on watermarks	74
5.2.2	The generalised (modified) matched filter	77
5.2.3	Simulation results and discussion	78
5.3	Watermark decoding after geometric distortion	81
5.3.1	The geometric distortion attack	81
5.3.2	Image registration based on motion estimation	81
5.3.3	Registration of rotated or scaled images	95
5.4	Robustness of watermarks against denoising attacks	96
5.4.1	The denoising attack	96
5.4.2	Simulation results and discussion	98
5.5	Chapter summary	100
6	Watermarking as Communications with Side Information	103
6.1	Introduction	103
6.2	Communication with side information and quantisation based watermarking	104
6.2.1	Definition of problem	104
6.2.2	Costa's solution	105
6.2.3	Practical implementation of Costa's solution	107
6.2.4	Extension to multiple samples quantisation	110
6.3	Spread spectrum and quantisation watermarking	111
6.3.1	Comparing spread spectrum and quantisation based watermarking	111
6.3.2	Spread transform watermarking	111
6.4	Spread transform watermarking with complex wavelets	114
6.4.1	Embedding algorithm	114

6.4.2	Decoding algorithm	116
6.4.3	Parameters selection	117
6.4.4	Performance of spread transform watermarking	119
6.5	Comparing CWT spread transform watermarks with other quantisation based schemes	121
6.6	The role of error control codes in watermarking	122
6.7	Comparing spread spectrum and spread transform watermarks	126
6.8	Chapter summary	130
7	Conclusions and Future Research	131
7.1	Thesis review	131
7.2	Future research directions	132
7.2.1	Blind image registration	133
7.2.2	Remodulation attack and second generation watermarks	133
7.2.3	Partially and iteratively image adaptive watermarking	133
7.2.4	Extension to video watermarks	134
	Appendices	135
A	Summary of visual models	135
A.1	The CWT visual model	135
A.2	The DWT visual model	136
A.3	The DCT visual model	136
B	Analysis of blind spread spectrum watermark decoding and detection using correlation	139
B.1	Decoder performance	139
B.2	Watermark detection	143
C	Test images	147
	Bibliography	149

List of Figures

1.1	Relationship between information hiding, steganography, watermarking and compression	3
1.2	A generic watermarking system	5
2.1	Watermark classification	12
2.2	Turbo encoder	18
2.3	Turbo decoder	18
2.4	Performance of the turbo code under AWGN	20
3.1	Single Tree Discrete Wavelet Transform	26
3.2	Dual Tree Complex Wavelet Transform	28
3.3	Wavelets and scaling functions of the CWT and the DWT	29
3.4	Impulse responses of the CWT and the DWT filters in 2-D	29
3.5	2 levels decomposition of the House image in the CWT and the DWT domains	30
3.6	Weber contrast sensitivity	32
3.7	HVS transducer function	33
3.8	Image quality metric	35
3.9	Change in detection threshold due to background luminance	37
3.10	Examples of heavily watermarked images in the CWT, the DWT and the DCT domains	40
3.11	Magnified watermark in two highlighted areas in fig. 3.10a	41
4.1	A simple watermark model	45
4.2	A generic spread spectrum watermark embedder	49
4.3	Correlator output statistics under mild and heavy attacks	55
4.4	Comparison of three types of correlator	56
4.5	A generic spread spectrum watermark decoder	57
4.6	Partitioning of the DCT coefficients into channels	58

4.7	Watermark examples - Lena	60
4.8	Watermark examples - Baboon	61
4.9	Watermark examples - Pills	62
4.10	Modified matched filter performance under compression	63
4.11	Same results under JPEG for watermarks with constant perceptual distortion	64
4.12	Comparing effects of JPEG vs JPEG2000 on watermark decoding	65
4.13	1-bit watermark detection performance under compression	66
4.14	Modified matched filter performance under other attacks	67
4.15	Receiver operation characteristics of two watermark detectors	71
5.1	Compression model	75
5.2	Comparison between modified correlator and generalised Gaussian decoder	79
5.3	DWT and DCT coefficients distribution after compression	80
5.4	The geometric distortion attack	82
5.5	Motion estimation algorithm in the CWT domain	84
5.6	Square difference surface	87
5.7	Example of motion vectors	90
5.8	Example of interpolation using radial basis functions	92
5.9	Motion vector correction	92
5.10	Example of registration	93
5.11	Effects of registration on watermark decoding	94
5.12	Denoising attacks	99
5.13	Examples of denoising and remodulation attacks	99
6.1	Watermarking with side information	105
6.2	Special case of communication with side information with Gaussian host and Gaussian interference	105
6.3	Non-oblivious watermarking model	105
6.4	Costa's approach to communication with a Gaussian host in an AWGN channel	107
6.5	Quantisation Index Modulation in 1-D	109
6.6	Scalar Costa Scheme	110
6.7	Spread Transform watermarking	113
6.8	Spread Transform watermarking with complex wavelets	116
6.9	Effects of α on quantisation based watermarks decoding performance	118

6.10 Comparing CWT Spread Transform watermarks with other quantisation based schemes	122
6.11 An (n, k) error control code forms a packing of k -sphere inside an n -sphere.	125
6.12 Theoretical detection and false positive probabilities of watermark detection using error control codes	126
6.13 Spread spectrum vs. spread transform with frequency partitioning under JPEG	128
6.14 Spread spectrum vs. spread transform watermarking at different capacities under JPEG	129

List of Tables

3.1	Table of empirical contrast thresholds	37
3.2	Table of k values	38
3.3	Comparison of HVS models in terms of perceptual error	39
4.1	Comparing inter symbol interference when orthogonalisation is applied before or after adaptive scaling of the CWT coefficients	47
6.1	Comparing spread spectrum and quantisation based watermarks	112
A.1	Table of k and C_{T_0} for the CWT	136
A.2	Table of k and C_{T_0} for the DWT	136

Chapter 1

Introduction

Digital watermarking has recently become a popular research area due to the proliferation of digital data in this Internet age and the need to find a solution to protect the copyright of these materials. This chapter gives a brief introduction to watermarking. We relate it to steganography and compression. The basic watermarking system is described and we discuss various applications, requirements and design issues. Finally we give an overview of the remainder of this thesis, which is concerned with designing watermarking systems for digital images in the complex wavelets domain.

1.1 What is watermarking?

1.1.1 Definition

Watermarking describes techniques which are used to convey information in a hidden manner¹ by embedding the information into some innocent-looking cover data. Typically, this information is required to be robust against intentional removal by malicious parties. In contrast to cryptography, where the *existence* but not the *meaning* of the information is known, watermarking aims to hide the existence of the information from any potential eavesdropper altogether. Watermarking has existed since approximately 15th century and in the past watermarks were mainly used on papers to identify the mill who made them. We call these *physical* watermarks because they exist in a physical media. Nowadays, physical watermarks are commonly used to authenticate important documents, for example, banknotes and passports. With the advance of the Internet and the ubiquity of digital data, it is natural to extend the

¹Watermarks are usually imperceptible, but there are also visible watermarks (see section 1.1.3).

idea of watermarking to digital data. A popular application of *digital* watermark is to give *proof of ownership* of a piece of digital data (image, audio or video) by embedding copyright information as a watermark into the data itself. The concept of using an embedded signal for the purpose of proving ownership was first mentioned in a patent in 1961 [65]². Some of the earliest publications concerned with watermarking digital images include Caronni [27], Tanaka *et al.* [142] and Tirkel *et al.* [144].

1.1.2 Relationship of watermarking to steganography and compression

Watermarking is closely related to *steganography* in that they are both concerned with covert communication and belong to a broader subject known as *information hiding*. However, the underlying philosophy of the two are different. Steganography normally relates to *point-to-point* covert communication between *two* parties and a steganographic system is typically not required to be robust against *intentional removal* of the hidden message. Watermarking, on the other hand, is usually a one-to-many communication and has the added notion that the hidden message should be robust to attempts aimed at removing it. In the case of copyright protection, obviously the copyright information should resist any modifications by pirates intending to remove it. Interested readers are referred to the survey by Petitcolas *et al.* [124] and his book [125] for more information on information hiding.

There exists a duality between watermarking (and information hiding in general) and data compression. While compression aims to identify the *perceptually insignificant* parts of the data and remove them, information hiding techniques try to insert information into them. From an information-theoretical point of view, information hiding is a game between the information hider and the attacker (for example, a compression system which removes the redundant parts of the data) [114]. Moreover, compression is one of the most common operations on images, therefore one must take into account of the effects of compression when designing a watermarking system. The most common compression standard at the moment is JPEG [119], which is based on the discrete cosine transform (DCT). JPEG2000 [34], which operates in the *real* wavelet domain, was proposed in 2000 as a future standard of compression and experiments showed it has superior performance compared with JPEG. Image compression is considered further in chapter 5. Figure 1.1 shows how everything fits together.

²To the best of the author's knowledge, this is the earliest reference to digital watermarking.

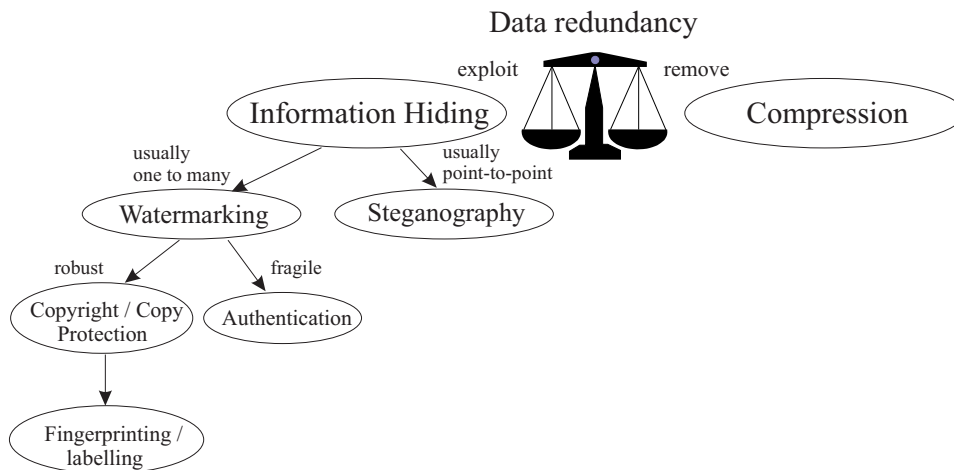


Figure 1.1: Relationship between digital watermarking, steganography, information hiding as well as compression. Information hiding aims to exploit the redundancy of data to convey information whereas compression tries to remove the redundancy. Steganography is usually a point-to-point communication process but watermarking typically involves many parties and has the additional requirement of being robust against intentional removal of the hidden message. Copyright protection, fingerprinting and authentication are different applications of digital watermarks, each with its own special requirements (see section 1.3).

1.1.3 Watermarking terminology

Over the years researchers have coined numerous terms to describe and classify watermarking techniques. We clarify these terms in this section.

The image or the piece of digital data into which we are hiding the information is called the *host* or *cover data*, and the hidden information is referred to as the *payload*. Most image watermarking systems involve making *imperceptible* alterations to the host image to convey the hidden information, but there also exist *visible* watermarks, which are visible patterns like company logos overlaid on top of an image. An example of such watermarks is the IBM Digital Library project [25, 113]³. However, in this thesis, we will concentrate on *invisible* watermarks. If the original, *unwatermarked* data is required in order to retrieve the watermark, the system is known as *non-blind* or *non-oblivious*, otherwise it is known as *blind* or *oblivious*. Watermarking systems typically require the use of a *key* (in the cryptographic sense) for retrieving the embedded payload. If the *same* key as in the watermark embedder must be used for retrieving the watermark, the scheme is known as *private*, because only the person/people who has/have the key can read the watermark. If a different key is needed to read the watermark, the scheme is known as *public*. Public watermarking is

³The IBM Digital Library project (1995). <http://www.dlib.org/dlib/july97/vatican/07gladney.html>

sometimes also known as *asymmetric* watermarking. We also distinguish between the *detection* and the *decoding* of watermarks. The former tells you whether something is likely to be there whereas the latter tells you what is there. In some watermarking schemes, only one random sequence is embedded. Thus these watermarks can only be detected but not decoded and they are referred to as ‘yes/no’ or ‘1-bit’ watermarks in this thesis. Similarly, we refer watermarks which carry actual payload as ‘multi-bit’ watermarks.

Watermarking systems can be *robust* or *fragile*. Robust watermarks are required to resist any modifications which do not decrease the commercial value of the cover image. On the contrary, fragile watermarks are *designed to fail* when the cover image is modified. Applications of fragile watermarks are discussed in section 1.3.4.

Fingerprinting and *labelling* refer to specific applications of watermarking, where the payload carries information such as the creator and the intended recipient of a piece of data. This is similar to the use of serial numbers to identify an individual copy of a product. Fingerprinting can be used to trace the origin of a piece of data if unauthorised copies of it are found. Other applications of watermarks are discussed in section 1.3.

1.2 Basic watermarking systems

All watermarking systems consist of an embedding part and a recovery part, which are shown in figure 1.2. The embedder takes the cover data, the payload and a (public/secret) key to produce the watermarked data. The recovery part takes the (possibly modified) watermarked data, the key and/or original unwatermarked data and returns either the payload (decoding) or a confidence measure of how likely a specific watermark is present (detection). Regardless of the medium we are watermarking in, most watermarking systems share the following properties:

- **Imperceptibility.** The modifications caused by the watermarking system should be unobtrusive (with the exception of visible watermark obviously). This means one should use some sort of perceptual model, both for embedding a watermark and for evaluating the induced distortion. The human visual system will be discussed in detail in chapter 3. Being imperceptible also implies watermarks typically have much less power than the cover data.
- **Redundancy.** Since a watermark has much less power than its host, the watermark is usually redundantly embedded in the host to achieve robustness

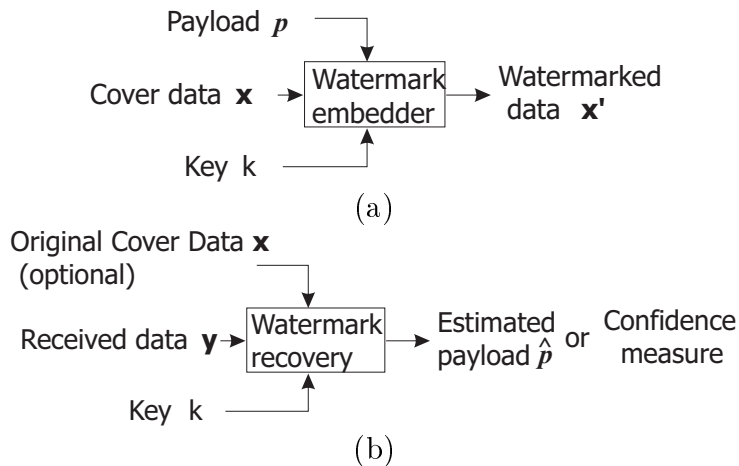


Figure 1.2: A generic watermark embedding (a) and recovery (b) system. The embedder takes the cover data, the payload and a key to produce the watermarked data. The recovery systems takes the (possibly modified) watermarked data, the key (can be public or secret) and optionally the original unwatermarked cover data and returns either the estimated payload or a confidence measure of how likely a given watermark is present.

so that the payload can be recovered from a fraction of the watermarked data.

- **The use of keys.** All watermarking systems use one or more keys to ensure security against intentional removal of the hidden information. This is in accordance with the *Kerckhoffs Principle* in cryptography, in which the attacker is assumed to have full knowledge of the system and so security must lie only in the choice of key [77]. If the watermark can be read, it may also be removed by the same person. This is why *public* watermarking still remains as a major challenge to researchers.

1.3 Watermarking applications

Depending on the specific application of a watermarking system, the actual requirements will vary. We review a few of the major applications of digital watermarks here. In the next section, we will discuss the design issues associated with some of these applications.

1.3.1 Copyright protection

Copyright protection is one of the major forces which drive the research in watermarking. Data can now be distributed in digital format with ease due to the existence of the Internet. The objective here is to embed copyright information into the data so

that the rightful owner of a piece of data can at least prove his/her ownership in the case of a dispute. The watermarks in this scenario obviously require a high level of robustness and should resist attempts in removing them. Note that watermarks for copyright protection do not prevent people from copying the digital data, they simply exist as a means for owners to assert ownership over some digital data. Typically these watermarks are used in conjunction with a *buyer-seller protocol* [112] in online distribution of digital data.

1.3.2 Copy protection

In contrast to copyright protection, a copy protection mechanism actually prevents users from making unauthorised copies of the digital data. This is difficult in open systems like the Internet but it is possible to enforce copy protection in a controlled system like the DVD player. For example, the watermark which exists on a DVD [21, 101] tells a compliant DVD player whether a user is allowed to copy the video.

1.3.3 Fingerprinting for pirate tracing

Watermarks are used in fingerprinting applications to identify the legal recipient of the digital data and are typically used together with copyright protection watermarks in a transaction. The existence of multiple differently watermarked copies of the same data allow the collusion attack (section 2.7) and so fingerprints must be designed to be collusion-secure [49]. Watermarks for fingerprinting otherwise have identical requirements to that of copyright watermarks.

1.3.4 Watermarking for authentication

Fragile watermarks are used to authenticate digital data. For example, if a *digital* photograph is to be used as evidence in a court, we have to prove that the photo has not been manipulated. A fragile watermark can be inserted into the image as soon as it is taken. If the image is modified maliciously, the watermark will be destroyed. If the watermark can be retrieved by the recipient, the image is deemed authentic, otherwise, it should be discarded as fake. A low level of compression is usually permitted but not content alteration of the image. Therefore a fragile watermark will have some robustness rather than like a checksum which fails even if only 1 bit of the data has been changed. In addition, it should be difficult for a malicious user to simultaneously modify both the cover data and the fragile watermark. There are several types of

fragile watermarks. Some only allow us to detect if an image has been modified (e.g. Fridrich [56] and Kundur [90]); some allow us to calculate an approximate version of the original image in the modified regions (e.g. Rey *et al.* [136]); while some allow us to invert the watermarking process and recover the original unwatermarked image if it is successfully authenticated (e.g. Fridrich [58, 59]). The last type is of particular interest to the medical community. Medical images cannot afford to be modified since this might cause misdiagnosis. Invertible fragile watermarks allow us to both authenticate and recover the original digital medical image. This thesis, however, concentrates on the design of robust watermarking system and fragile watermarks will not be discussed further.

1.4 Algorithm design issues

1.4.1 Imperceptibility, robustness and capacity

All watermarking algorithm designs involve determining a tradeoff between three conflicting requirements: namely imperceptibility, robustness and capacity. The higher is the embedding strength of the watermark, the more robust it is, but it will also be more visible. The more data we embed, the less is the energy allocated to each bit of the payload and the less robust is the watermark. Using a good perceptual model will allow us to maximise the energy of the watermark while keeping its visibility to a minimum.

It is hard to define what robustness means. Ideally a robust watermarking scheme should resist any form of malicious distortion which does not render the image useless. It is perhaps impossible to design such a *perfect* watermark. Depending on a particular application and the medium we are working in (image, audio or video, etc.), some attacks will be more important than others and a practical system may only be required to be robust against these attacks. We will discuss various types of attacks in more detail in the next chapter (section 2.7).

Capacity is normally not as important an issue in watermarking as in steganography⁴. In steganographic applications, a high capacity may be required for conveying a hidden message. The knowledge of the host signal in the watermark embedder should be exploited in this case to maximise the capacity of the cover data. This is discussed in more detail in chapter 6.

⁴There are, however, applications in which a protocol is to be sent via a watermark [67, 132], in which case a high capacity may be necessary.

1.4.2 Invertibility and resolving rightful ownership

Craver *et al.* [45] argue that, in order to be able to resolve rightful ownership, a watermarking scheme must be non-invertible. At the very least, the algorithm must be oblivious, otherwise a pirate can just create his/her own watermark, subtract it from the watermarked image to make a *counterfeit* image and claim it as the original image. Using a non-blind algorithm will detect the pirate's watermark using the counterfeit image as the original. Being blind is not the only requirement, additional techniques such as time-stamping will also be needed to make a watermarking scheme non-invertible.

1.4.3 Public vs. private watermarking

Most of the watermarking schemes in current literature are private because a secret key is required in order to retrieve the watermark. It is not possible for a user to find out whether a piece of data is watermarked unless he/she has this key. Therefore some commercial systems embed both a private and a public watermark into the data. However, as mentioned earlier, as soon as a watermark can be read, it may be removed by inverting the embedding process. Hence embedding two watermarks is not a complete solution. A public watermark should allow watermarking *detection* without the need of a key or revealing the actual watermark. The use of special mathematical properties of some linear transform, for example, the eigenvectors, has been suggested by Eggers *et al.* [53] for designing public watermarking systems. Alternatively one can use zero knowledge protocols in cryptography [44]. The remainder of this thesis will however concentrate on private watermarks.

1.5 Thesis overview

Chapter 2 surveys current watermarking techniques and attacks on watermarking systems. In particular watermarking techniques are classified based on the domain they operate in, how the watermark payload is encoded, how the embedding locations are chosen, how the watermark is combined with the cover signal and how the watermark is extracted, whereas attacks on watermarks are classified into four categories: estimation-based attacks, geometric attacks, cryptographic-like attacks and protocol attacks. A brief review of the turbo code as an example of error control codes is also given due to their extensive use in watermarking systems.

In this thesis, we concentrate on designing watermarking algorithms with complex wavelets because they are relatively new and possess properties which have been shown to be very useful in image processing applications. Chapter 3 introduces the Complex Wavelet Transform (CWT), and explains why it is better than the traditional real Discrete Wavelet Transform (DWT). A commonly used human visual model is described and its close relationship with the CWT is demonstrated. Finally, a visual model in the CWT domain is developed which forms the basis of our watermarking systems in the CWT domain.

Chapter 4 describes our proposed spread spectrum based watermarking algorithm in the complex wavelet domain. Unfortunately, the CWT is a redundant domain and this can cause problems unless precautions are taken. We discuss how one can design the watermark algorithm to get round the problems caused by this redundancy. We also derive a new decoder for decoding an image-adaptive watermark and address the issue of watermark detection.

Chapter 5 discusses ways to improve the performance of the watermark decoder when the watermarked image is attacked. We consider three scenarios: compression, geometric distortion and denoising. In the case of compression, an alternative decoder is derived and compared with the decoder proposed in chapter 4. A registration algorithm is developed to combat geometric distortion attacks. It is also demonstrated that denoising has similar effects compared to compression and we can use the same strategy for decoding watermarks.

Chapter 6 introduces a new class of watermarking techniques based on quantisation. Quantisation based watermarking has recently been proposed as a means to achieve high capacity watermarking due to its ability to suppress interference from the host image. Quantisation and spread spectrum based watermarking techniques are contrasted and compared. A hybrid watermarking algorithm based on both quantisation and spread spectrum is developed in the CWT domain, which is compared with the spread spectrum based method described in chapter 4.

Chapter 7 closes with conclusions and gives possible future research directions.

1.6 Notation conventions

Throughout this thesis, scalar quantities are represented by normal type font, e.g. x ; vector and matrix quantities are denoted by bold type font in lower and upper case respectively, e.g. \mathbf{f} , \mathbf{T} ; a field of scalar/vector quantities is denoted as $\{\}$ surrounding the respective scalar/vector element, e.g. $\{x\}$, $\{\mathbf{f}\}$, etc.

Chapter 2

Survey of Watermarking Techniques and Attacks

2.1 Introduction

As explained in chapter 1, one of the reasons for the rapid development in research in digital watermarking is the need to find a solution for protecting intellectual properties of digital material. Although there exist techniques for watermarking all kinds of digital data, most of the literature addresses the watermarking of still images for copyright protection. In this chapter, we give an overview of existing robust image watermarking algorithms, and briefly review the turbo code as an example of error control codes used in watermarking. Most of the watermarking techniques described apply to greyscale images, but they can be easily extended to colour images by watermarking their luminance component. Finally we give a classification of attacks on watermarking systems and describe examples of each category.

Most of the existing watermarking algorithms can be classified according to the following criteria:

- The selection of locations where the watermark is embedded. The use of human visual models in watermarking is addressed.
- The domain in which the algorithm operates. For example, an algorithm can modify the image in the spatial domain directly to embed the watermark, or it can first transform the image into some domain (for example, Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT), Wavelet), insert the watermark and inverse transform the result to obtain the watermarked image. The pros and cons of working in different domains are discussed.

- The encoding of the payload. How the payload is represented by a watermark is described. The use of error control codes is also reviewed.
- The formation of the composite signal. How the watermark is embedded into the cover data is addressed. The watermark can simply be added to the cover data or the cover data may be changed in a way to represent the presence of a watermark.
- How the watermark is extracted. Since the watermark has low power compared with the cover data, the watermark decoder operates in a low signal-to-noise ratio environment. There exist numerous ways in which the performance of the decoder can be improved, including the case where the watermarked image has been modified by a malicious user.

Each of these features of a watermarking algorithm will be detailed in the following sections. The classification is summarised in figure 2.1.

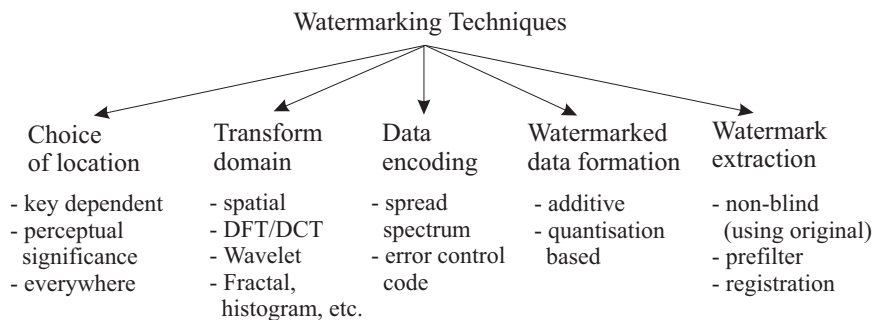


Figure 2.1: Watermarking techniques can be classified according to the choice of embedding location, the operation domain, the encoding format, formation of the watermarked data as well as the optimisation of the detector/decoder. These criteria are discussed in sections 2.2 to 2.6.

2.2 Choice of embedding locations in the host

Human eyes are less sensitive to noise in regions with textures than in smooth areas of an image. Compression systems also tend to preserve textured areas and edges while coarsely quantising smooth areas. Based on these reasons Cox *et al.* [41, 42] argue that one should embed watermarks in perceptually significant parts of an image. An attacker cannot remove the watermark easily without causing significant distortion. Numerous approaches have been proposed which choose embedding locations based on this principle. Koch *et al.* [83, 84] suggest changing the ranking order of pairs

of DCT coefficients in mid-frequency to embed a watermark, because mid-frequency coefficients tend to survive JPEG compression and cause less visual distortion than the lower frequency coefficients. Bors and Pitas propose a similar scheme but use a different constraint [23]. Wang and Kuo select significant wavelet coefficients for watermarking [156] based on a multi-threshold wavelet codec [155], but their approach requires the original image for decoding. In [46], Darmstaedter *et al.* describe a spatial watermarking scheme where the image is divided into blocks, with each block classified depending on its contrast and being watermarked accordingly. Watermarking in different domains will be discussed in more detail in section 2.3.

In addition to choosing locations based on their visual significance, one can also use a key, usually a random number seed, to select the coefficients to be marked. An example is the Patchwork algorithm [18], which selects n pairs of pixels using a key k . The luminance of half of the selected pixels are incremented by 1 while the other half are decremented by 1. The decoder, using the same key k , selects the same pairs of pixels and compares the difference between the means of the two halves. Based on the assumption that n is sufficiently large and that the chosen pixels are roughly independent and identically distributed, the difference between the means will be roughly $2n$. This method, however, only allows us to verify the existence of a watermark, and cannot carry extra payload. Kutter's algorithm [91] uses a key to partition the image pixels and then select a subset of them for watermarking. The watermark is also perceptually adapted to the image, unlike the Patchwork algorithm, which only changes the statistics of the image. The importance of perceptually adaptive watermarking will be pursued further in section 5.4. Finally, one can also repeatedly watermark the whole image and weight the different extracted copies at the decoder according to their reliability [89].

2.3 Watermarking domain

The Patchwork algorithm described in the previous section operates in the spatial domain. Other examples of spatial watermarking schemes include [91, 126, 154]. Many researchers propose watermarking in some frequency domain due to the better resilience against intentional attacks they provide. Some example domains are given next.

2.3.1 Discrete Fourier Transform

The DFT enables the watermark embedder to shape the watermark spectrum according to human visual sensitivity to each frequency band. De Rosa *et al.* [47] propose a scheme which embeds a payload (or a random sequence) directly into the magnitude of the DFT coefficients in mid-frequency bands, while ÓRuanaidh *et al.* [116] propose modulating the phase instead. Other variations of embedding in the DFT domain proposed include [135, 137]. The scheme described in [135] embeds the watermark in the Karhunen-Loeve Transform (KLT) of the magnitude of the DFT coefficients. There is no clear optimal way of how the DFT coefficients should be modified. However, no matter how the coefficients are modified to encode a watermark, one must ensure the coefficients in two of the quadrants are conjugate symmetrical with the other two, because images are real signals.

2.3.2 Discrete Cosine Transform

The DCT domain has been commonly used for watermarking because it is the domain used by JPEG and MPEG. By incorporating the JPEG compression mechanism into the watermark embedder, the watermark can gain resilience against compression. Another reason for employing of the DCT domain is due to extensive study of visual models in the domain [158] in the context of compression, which can be used for adapting the watermark to the cover image.

Methods for embedding watermarks into the DCT coefficients are similar to those in the DFT domain. One can either add a pseudo-random noise sequence directly to the host DCT coefficients [69, 127, 141], or impose constraints on some coefficients [23, 83, 84].

2.3.3 Discrete Wavelet Transform

Wavelets have recently become an important tool in image processing due to the good energy compaction properties they possess as well as the existence of efficient algorithms for computing the wavelet transform. They also form the basis of the new compression standard JPEG2000. Wavelets are discussed in more detail in chapter 3. In a nutshell, the wavelet transform splits the image into multiple scales in a spatial-frequency manner. Kundur *et al.* [87] describe a scheme based on fusing the wavelet coefficients of the cover image with those of the watermark. Prior to fusion, the watermark coefficients are scaled according to a saliency model [159]. Wang and

Kuo [156] suggest using significant wavelet coefficients for watermarking because these coefficients cannot be modified significantly without destroying the cover image. Xia *et al.* [165] propose a hierarchical watermark extraction algorithm which can reduce the computational load if the watermarked image is not too severely distorted. The idea is to try to detect the watermark in the finest scale first, and use the coarser scales only if detection fails. Kundur *et al.* [88] later propose a scheme based on quantising the median of a triplet of coefficients. The wavelet packet transform, which is closely related to the DWT but differs in the way the frequency plane is partitioned, has also been suggested for watermarking [98].

2.3.4 Translation, scale, rotation invariant domain

When the watermarked image undergoes an affine transform, many watermarking systems fail. ÓRuanaidh *et al.* [117] introduce the Fourier-Mellin Transform to overcome this problem. A log-polar map of the DFT coefficients is computed and the watermarked is embedded in this domain. Scaling and rotation are mapped to translation in this domain and the watermark detector can perform a search around the expected embedding locations to extract the watermark. Alternatively, a geometrically transformed image can be registered prior to watermark detection. Various image registration techniques are discussed in section 5.3.

2.3.5 Key dependent domain

Fridrich *et al.* in [57] propose using a secret key to generate the basis functions of the transform domain used for watermarking. As long as the attacker does not know these basis functions, he has to induce severe distortion in order to defeat the watermark. This technique is in fact a special case of spread transform watermarking described in chapter 6 (section 6.3.2), except that the knowledge of the cover signal is not exploited at the embedder. Meerwald *et al.* [111] propose a similar approach using key dependent wavelet filters.

2.3.6 Other domains

Although most watermarking techniques operate in either the spatial or a frequency domain. A few techniques have been proposed in less common domains. Puate *et al.* [131] propose using fractals to encode a watermark. Fractals describe the relationship between blocks within an image. By varying the mapping between blocks,

one can encode a message. Coltuc *et al.* [36] suggest modifying the histogram of an image to hide a watermark, but since most image processing operations will change the histogram, the watermark is fragile. Podilchuk *et al.* [129] propose a method for watermarking the JPEG bitstream directly. Unfortunately, any algorithm which relies on a particular aspect of file format (for example, colour palette, compressed bitstream) will fail if the image format is changed.

2.3.7 On the choice of transform

Watermarking in the spatial domain is fast and hence is suitable for real time applications like watermarking video as it is being broadcasted [73]. Watermarking in frequency domains requires more processing power but purposely built DSP chips can solve this problem. On the other hand, depending on the properties of the particular transform we use, the watermark can be made to be resilient to certain attacks. For example, the magnitude of DFT coefficients is affected less by spatial shifts than the value of the pixels and so watermarks embedded in the DFT domain are more robust than spatial watermarks against translation and cropping¹. Similarly, DCT based watermarks can be made to resist JPEG compression. Although techniques exist for augmenting spatial watermarks so they can cope with the aforementioned attacks, the most important advantage that frequency domains offer is the ability to *independently* process the components of an image in different frequency bands, which in turn allows us to weight the detector response from different bands according to their reliability. This is discussed in more detail in section 4.4.4.

2.4 Encoding of payload

2.4.1 Spread spectrum

Since watermarks are required to have low power (so they will be imperceptible), watermarking can be seen as a communication process through a very noisy channel. Spread spectrum techniques are known to facilitate communications in noisy environments and are therefore widely used in watermarking. Almost all watermarking schemes represent the payload in the form of a pseudo-random noise (PN) sequence,

¹The amount by which the magnitude of the DFT coefficients changes depends on the extent of cropping/translation relative to the image size. If the shifts are circular, the magnitude will be invariant, but attackers hardly use circular shifts as an attack because the resulting image will certainly not be visually similar to the original image.

which is the so-called direct sequence form of spread spectrum. The random number seed used to generate the sequence becomes the key of the watermark. Because the decoder needs to know the key to decode the watermark, these schemes are essentially private. Hartung *et al.* [63] propose making part of the PN sequence public to allow public decoding of the watermark. The detector/decoder also needs to synchronise with the PN sequence before the watermark can be detected/decoded. Unfortunately this becomes a major weakness of many existing schemes, as we will see later. A variation on the basic spread spectrum principle is bandpass/lowpass filtering the sequence prior to watermark embedding [24] so that the watermark will have little energy in the high frequency components, which tend to be discarded by compression systems.

2.4.2 Error control codes

In order to improve the robustness of a watermark further, error control codes are often used to encode the payload prior to embedding. In this section, we review the turbo code, which was discovered relatively recently and was shown to achieve asymptotically close to the Shannon's bound [19]. The two main features of turbo codes are concatenated coding and iterative decoding. Concatenated coding refers to the fact that each codeword is the concatenation of the results of two or more encoders. Figure 2.2 shows an example of the simplest possible turbo encoder which has two constituent encoders. In the case of turbo codes, the encoders are identical and each is a recursive convolution coder. The second encoder is preceded by an interleaver π so that the input is scrambled before being fed into the encoder. The final output is the concatenation of the payload, followed by the outputs from the first then the second encoder. The code is therefore systematic (the payload appears directly in the coded data). Additional encoders, each preceded by a *unique* interleaver, can be added and their outputs concatenated to construct more complex turbo codes. Given an l -bit payload, an encoder with constraint length m and c constituent encoders, the rate of the code is given by $\frac{l}{(c+1)(l+m-1)}$,² which is approximately $\frac{1}{c+1}$ for large l . The interleaver should be *random* so that on average the Hamming distance between any two codewords is large. The constraint length of the encoder does not affect the rate significantly but it affects the decoding complexity exponentially. There are 2^{m-1} states for a constraint length m encoder. Therefore increasing the constraint length

²The factor $l + m - 1$ appears because $m - 1$ bits are needed at the end to terminate the trellis during each encoding process.

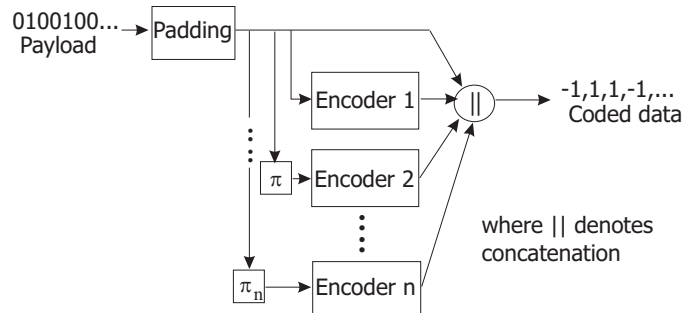


Figure 2.2: A generalised turbo encoder. All the encoders use recursive convolution code and are identical. The first encoder takes the padded payload as direct input and returns the parity bitstream. All the other encoders use a differently interleaved (denoted by π) version of the padded payload as the input. The final output is formed by the padded payload concatenated with the parity bitstreams from each of the encoders. The simplest turbo codes have 2 encoders, but additional encoders can be used to construct lower rate codes. Higher rate codes can be constructed by puncturing. The payload is padded to ensure all the encoders return to *all-zero* state at the end of the coding stage.

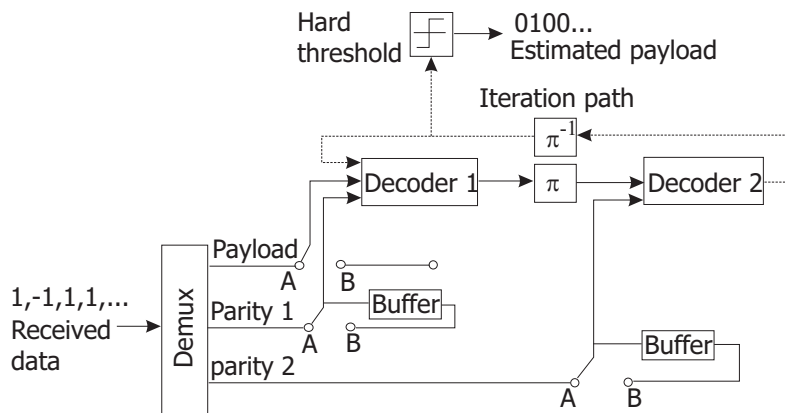


Figure 2.3: An iterative turbo decoder. The switches are in position A at the beginning. The received data is first demultiplexed into different parity bit streams and the systematic payload. In the first iteration, the first decoder returns a soft output using the received payload and the first parity bitstream. This soft output is interleaved and passed together with the second parity bitstream to the second decoder, which in turns returns another soft output. The switches are switched to position B for the iteration stage. The output from decoder 2 is deinterleaved and fed back to decoder 1, replacing the received payload. Hence each decoder uses the (interleaved/deinterleaved) output from the other decoder and its parity bitstream to produce a soft output. This is repeated a number of times before the deinterleaved output from decoder 2 is hard thresholded to give the estimated payload.

by 1 will double the decoding complexity. In practical applications, the constraint length is limited to around 9. The longer is the constraint length, the better is the performance of the code. In our simulations, we use encoders with constraint length 5 as a compromise between decoding complexity and performance. The simplest possible turbo codes have two encoders and therefore a rate of $\frac{1}{3}$. Puncturing can be used to increase the code rate at the expense of lowering the performance of the code. We do not use punctured codes in this thesis as the aim of spread spectrum coding is to add plenty of redundancy to the payload.

Figure 2.3 shows the corresponding example decoder for the encoder in figure 2.2. In the beginning all switches in the figure are in position A. The received bitstream is first partitioned into the parity bitstreams from each encoder and the data stream. The first decoder takes the data stream and the first parity bitstream and produces a soft output (using for example the soft output Viterbi algorithm [62]) of the estimated payload, which is then *interleaved* and passed to the second decoder together with the second parity bitstream. The second decoder produces an improved soft estimate of the payload. The switches are switched to position B after this and remain there. The soft output from the second decoder is *deinterleaved* and fed back to the first decoder. This is the iterative step of turbo decoding. After a few iterations, the output from the second decoder will converge and we can just hard threshold the *deinterleaved* output to obtain the decoded payload. The improvement in performance decreases with increasing number of iterations and it is found that after 18 iterations the improvement is negligible [26]. In our implementation, we use 10 iterations, which is a good compromise between decoding complexity and performance. Valenti [146] analyses the turbo code as a linear code and obtains an approximate upper bound of word error rate under AWGN:

$$P_{err,turbo} \approx \leq \sum_{w=d_{min}}^{d_{max}} A_w Q(\sqrt{SNR \cdot R_c w}), \quad (2.1)$$

where d_{min} and d_{max} are the minimum and maximum Hamming distance of the code; A_w is the number of codewords with weight w ; R_c is the rate of the code and SNR is the signal to noise ratio per bit.

Figure 2.4 shows the performance of the turbo code under AWGN with the following parameters: constraint length 5 encoder, block length 128 with 2 encoders, which results in a (384,124) code. The output (bit error rate) BER is truncated at 10^{-5} . We can see that the output BER curve has a steep gradient. If the input BER is less than a certain threshold under Gaussian noise (or the input SNR is higher than

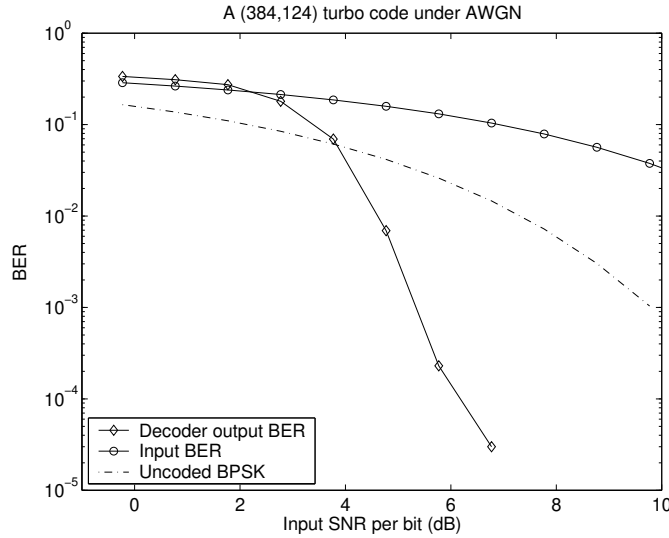


Figure 2.4: The performance of a (384,124) turbo code under AWGN with 10 decoding iterations and a random interleaver. The gradient of the output BER is very steep and we can achieve very small decoding error if the input SNR is higher than a certain threshold.

a certain threshold), we can achieve very small decoding error by employing turbo codes³. This has important consequences in watermarking, and we will discuss this property of error control codes further in chapter 4 and chapter 6. The use of turbo codes in watermarking is investigated in Balado *et al.* [16] and Kesal *et al.* [78]. The Hadamard code, sometimes also referred to as M -ary or multilevel coding, has also been considered for used in watermarking [93], but it offers worse tradeoff between performance and complexity than the turbo code.

2.5 Formation of the composite signal

The most common way of forming the watermarked signal is the direct addition of the watermark, usually in the form of a PN sequence, to the amplitude of the cover data. This can happen in the spatial, the DFT, the DCT, or the wavelet domain as discussed before. Alternatively, the phase of the cover data can be modified instead, as in [116]. No matter what aspect of the cover data we are modifying, since we are not taking into account the cover data in the embedding process, it will act as strong interference at the watermark decoder, which limits the decoder's performance even in the absence of attacks.

Recently, a new class of embedding methods have been proposed which are based

³If we use a longer blocklength for the turbo code, the BER curve will have a steeper gradient.

on dither quantisation [32]. The cover samples are quantised to represent the embedded data. The decoder quantises the received samples and looks at which bin each sample falls into to recover the embedded data. The cover signal no longer acts as interference. Spread spectrum and quantisation based watermarking systems will be compared and discussed in section 6.3.1.

2.6 Watermark extraction

Most spread spectrum based schemes use correlation as the basis for watermark detection and decoding. This implicitly assumes the underlying interference is Gaussian because only then will correlation be optimal. Typical images in the spatial domain, on the other hand, do not follow the Gaussian distribution and many authors (for example, [74, 91]) suggest prefiltering (by a highpass filter) the received image prior to detection to improve the performance. This is because much of the image energy resides in the low frequency components and will be removed and highpass filtering also makes the signal more Gaussian like. Subtracting the original image from the watermarked image prior to detection, as in non-blind algorithms, can also be seen as a special case of preprocessing. If some prior knowledge of the distribution of the image coefficients is available, one can use a maximum likelihood decoder instead of the correlator. For example, de Rosa *et al.* [47] model the DFT coefficients using a Weibull distribution and derive the corresponding optimal decoder. If watermarking occurs in a frequency domain, one can combine the outputs from different subbands in an optimal way to maximise the overall signal to noise ratio. This is discussed in section 4.4.4. Another aspect of watermark detection which can be optimised is the selection of the threshold. Piva *et al.* [128] argue that choosing the threshold based on the Neyman-Pearson Principle [130] is better than minimising the error probability in the presence of attacks.

Spread spectrum systems rely on the perfect synchronisation between the transmitter and the receiver for successful communication. If the image is cropped, scaled or otherwise geometrically transformed, this synchronisation is lost and the watermark must be resynchronised prior to detection. Several authors have proposed using a separate pattern [120] or constructing the watermark sequence in a special way [149] to allow synchronization. Geometric attacks will be discussed in sections 2.7.2 and 5.3 and image registration techniques will be discussed in section 5.3.

2.7 Attacks and benchmarks

In this section we review some common as well as state-of-the-art attacks on robust watermarking systems. In general, attacks can be classified into four classes: removal based attacks, geometric attacks, cryptographic-like attacks and protocol attacks [153]. There are also specific attacks which target only particular algorithms (for example, [105]).

2.7.1 Removal based attacks

These attacks aim to estimate the watermark from the image and *subtract* it from the image. This assumes we have some prior knowledge about the distribution of the watermark. In fact, compression can be considered as a special case of removal attack. An improved version of estimation based attack was proposed recently by Voloshynovskiy *et al.* [152], which is based on the idea from Langelaar *et al.* [97] and involves adding back part of the estimated watermark to the image. They called it the *remodulation* attack and this is discussed in section 5.4. Another important class of removal attack is the *collusion* attack, which does not actually remove the watermark, but rather reduces its power by averaging many *different* watermarked copies of the *same* image. Collusion attack is generally not a problem for image watermarks but it is easy to apply collusion attack to video by averaging similar frames. Collusion attacks can be circumvented simply by using collusion secure watermark sequences, for example, by making part of the sequence identical for all watermarks.

2.7.2 Geometric attacks

Geometric attacks alter the geometry of the image and *desynchronise* the PN sequence rather than remove it. This class of attacks include affine transformation (for example, scaling, rotation, etc.), cropping, geometric distortion, and jitter. Some watermarking schemes include a registration pattern to combat global transformation, but this is not very effective for geometric distortion because the transform is localised and random. In section 5.3 we propose a registration technique to cope with geometric distortion. Jittering is a special geometric attack where samples are removed at random places and duplicated elsewhere so the total number of samples remains constant. The jitter attack is generally not very effective in images (where a row or a column is jittered at a time) because the resulting artifacts quickly become visible. The jitter attack is more effective in audio watermarking, because a single audio sample is much less

significant with respect to the entire piece of audio compared to a line with respect to an image.

2.7.3 Cryptographic-like attacks

An example of cryptographic-like attacks is the brute force search of the existence of all possible watermarks and remove them if any is found. However, unless a system is badly designed such that the number of possible keys is small, brute force attacks are typically ineffective. If a watermark detector device and a copy of the watermarked signal are available, Kalker *et al.* [75] show that an unwatermarked copy of the signal can be constructed in time linear to the number of data samples, provided that the watermark detection algorithm is linear. They argue that a watermark detector should have non-linear components.

2.7.4 Protocol attacks

This kind of attack targets the application of a watermarking system. For example, if the algorithm is invertible, a pirate can construct a counterfeit original by subtracting his own watermark from the watermarked image (as discussed in the previous chapter (section 1.4)). Another attack of this type is the copy attack [96], where the watermark is estimated from a watermarked image and copied onto another *unwatermarked* image so as to confuse the buyer-seller protocol.

2.7.5 Benchmarking

Petitcolas *et al.* [95, 123] propose the *StirMark* benchmark for the evaluation of watermarking systems. The attacks included are mainly geometric attacks (geometric distortion, scaling, cropping, etc.), noise addition and common signal processing operations such as compression, mean/median filtering. Due to the lack of removal based attacks in the benchmark, Pereira *et al.* [122] propose another benchmark which includes removal based attacks as well as separating attacks into multiple benchmarks for different watermarking applications. In chapter 5, we will address three attacks: compression, geometric distortion and estimation based attacks and discuss how to improve the watermark detector/decoder after attacks.

2.8 Chapter summary

In this chapter, we reviewed current watermarking techniques and attacks. Watermarking techniques were classified according to the following five criteria:

- The choice of embedding location.
- The domain in which the watermark is inserted.
- The way in which the watermark is encoded.
- How the watermark is combined with the cover data to form the composite signal.
- How the watermark is extracted.

The turbo code was reviewed as an example of error control codes for watermarking. The output BER of the turbo decoder has a steep gradient with respect to the input BER, which means if the input BER is below a certain level, one can achieve arbitrarily small decoded error. Attacks on watermarking systems were classified into four classes: removal based, geometric, cryptographic-like and protocol based attacks. Example attacks were given in each category.

In the next chapter we will introduce the complex wavelet transform and relate it to the human visual system. We will also derive an empirical human visual model in the complex wavelets domain.

Chapter 3

The Complex Wavelet Transform and the Human Visual System

3.1 Introduction

The use of wavelets in image coding has increased significantly over the years, mainly due to the superior energy compaction property of wavelets compared with the traditional transforms like the DCT, and that there exists an efficient algorithm [108] to compute the wavelet transform. The new compression standard, JPEG2000 [34], is based on the *real* Discrete Wavelet Transform, for example. However, the conventional real wavelet transform has two drawbacks: lack of shift invariance and lack of directional selectivity. These hinder the use of wavelets in other areas of image processing. In this chapter, the Complex Wavelet Transform, an alternative to the real wavelets, will be described. How the CWT overcomes the problems with real wavelets is discussed. We then review some existing models of the human visual system (HVS) and outline the relationship of the CWT with these models. A visual model in the complex wavelet domain is proposed and we show that the CWT allows a watermark to adapt to the content of the host image better than the DWT and the DCT, two commonly used frequency domains for image watermarking.

3.2 The Complex Wavelet Transform and its dual tree implementation

Figure 3.1 shows the analysis and the reconstruction trees of the conventional real wavelet transform for 1-D signals. Wavelets offer compact support in both spatial

and frequency domains and good energy compaction properties. Thus they are more suited for image compression than some traditional frequency transforms like DFT (which has wide spatial support) and DCT (which produces blocking artifacts, if a block based transform like that in JPEG is used). However, the DWT suffers from the following two problems:

1. Lack of shift invariance. This results from the downsampling operation at each level (see figure 3.1). When the input signal is shifted slightly, the amplitude of the wavelet coefficients at different levels varies dramatically. This can be problematic for operations which require shift invariance such as edge detection using wavelet maxima.
2. Lack of directional selectivity. As the DWT filters are real and separable, the frequency response is symmetric about zero in four quadrants of the 2-D frequency space and therefore the DWT cannot distinguish between the two opposing diagonal directions.

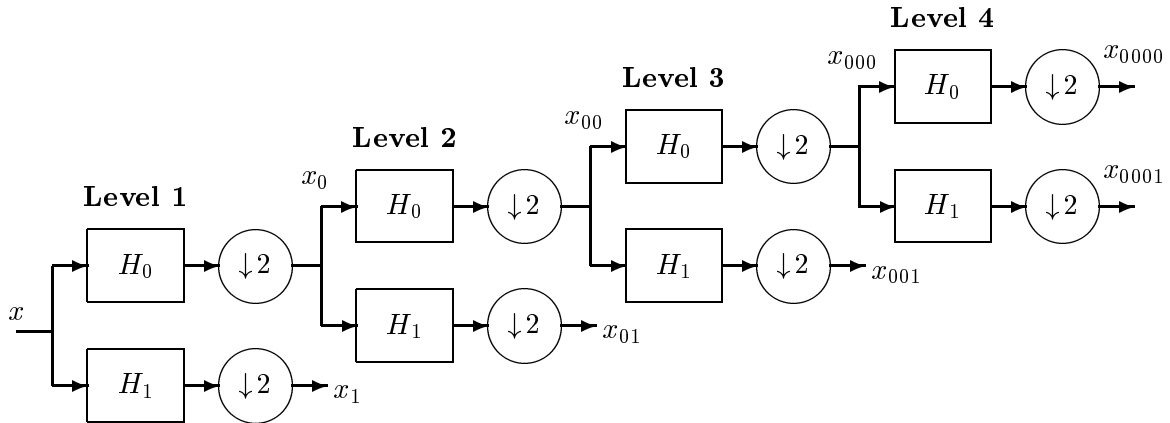


Figure 3.1: A filter tree showing the conventional real wavelet decomposition of 1-D signals. The wavelet filter H_1 and the scaling function H_0 are identical for all resolutions and have zero phase. Without loss of generality, odd length filters² were used for experiments involving the DWT in this thesis.

The first problem can be avoided if the filter outputs from each level are not downsampled, but this increases the computational costs significantly and the resulting undecimated wavelet transform (UDWT) still cannot distinguish between opposing

²We use a pair of near symmetric filters for the DWT, which are designed using the transformations of variables method [143].

diagonals because the transform is still separable. In order to distinguish opposing diagonals with separable filters, the filter frequency responses are required to be asymmetric for positive and negative frequencies. One way to achieve this is to use complex wavelet filters which can be made to suppress negative frequency components. As we will see later, a dual tree implementation of the Complex Wavelets can also suppress the aliased components of a signal, which result from the downsampling and upsampling operations, and this leads to the approximate shift invariance of the CWT.

Complex wavelets have not been used widely in image processing due to the difficulty in designing *complex* filters which satisfy the perfect reconstruction (PR) property. To overcome this, Kingsbury [79, 80] recently proposed a *dual tree* implementation of the CWT (DT CWT) which uses two trees of *real* filters to generate the real and imaginary parts of the wavelet coefficients *separately*. One can think of the two trees as being the real and imaginary parts of an effectively *complex* filter, which can be approximated as a *Gabor* filter of the form:

$$\phi(x) = g(x) \cdot \exp(-i\omega x), \quad (3.1)$$

where $g(\cdot)$ is a Gaussian envelope and $\exp(-i\omega x)$ is a modulating sinusoidal signal of frequency ω , which is the centre frequency of the wavelet or scaling function. The two trees are illustrated in figure 3.2 for 1-D signals. The top level filters in the two trees operate on the odd and even samples of the input respectively, in other words, this is equivalent to the UDWT. Even though the outputs of each tree are downsampled, by summing the outputs of the two trees during reconstruction, we are able to suppress the aliased components of the signal and achieve approximate shift invariance [82]. Below level 1, the filters in the two trees are designed such that they have identical frequency responses³. To compute the 2-D CWT of images, these two trees are applied to the rows and then the columns of the image, just like the conventional DWT. This operation results in 6 subbands per resolution instead of 3 as in the DWT. Figures 3.3 and 3.4 show the wavelet and scaling functions as well as the 2-D impulse responses for the CWT and the DWT. Figure 3.5 shows 2 levels of decomposition of the House image in the CWT and the DWT domains. We can

³In the original implementation of the DT CWT, the filters in the two trees can only have *similar* frequency responses as they must be of different lengths in order to make the sampling interval below the top level uniform. Kingsbury recently proposed a new design methodology of the filters [81], which results in better symmetry properties of the transform, and *identical* frequency responses for the analysis and reconstruction filter banks.

approximate the 2-D filter using a separable 2-D Gabor filter as in (3.1):

$$\phi(x, y) = g_x(x) \exp(-i\omega_x x) \cdot g_y(y) \exp(-i\omega_y y), \quad (3.2)$$

where $g_x(\cdot)$ and $g_y(\cdot)$ are now the corresponding envelopes for the row and column filter. The 2-D sinusoid orients the filter in the direction $-\omega_y/\omega_x$. Since the frequency response is not symmetric about zero for Gabor filters, the two diagonal directions will have *separate* filters associated with them. This is apparent in figures 3.4 and 3.5. The CWT can distinguish between the two opposing diagonal directions whereas the DWT cannot as its filters are real. The significance of the Gabor filter like nature of the CWT will become apparent later. The filters of the two CWT trees are listed in [81].

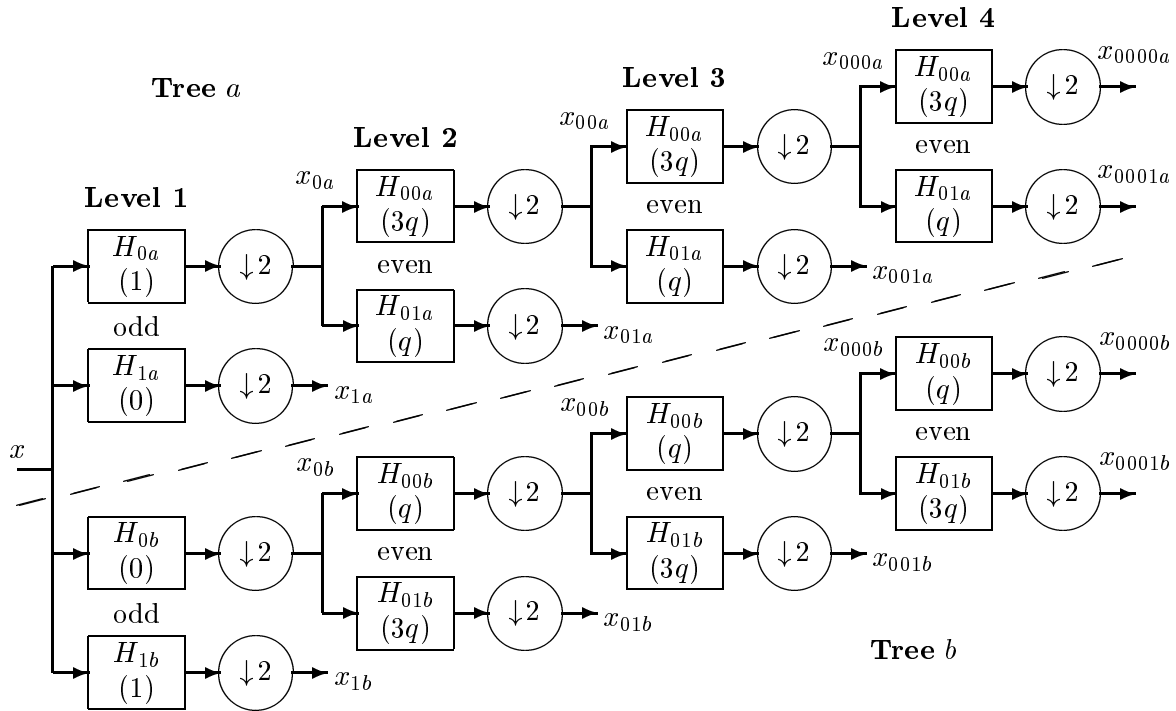


Figure 3.2: The Dual Tree implementation of the complex wavelet transform of 1-D signals. Tree a and b give the real and imaginary parts of the wavelet coefficients respectively. At the top level, identical odd filters are used for both trees and the two trees operate on the odd and even position (indicated by (1) and (0)) samples respectively. Below the top level, the filters of the two trees are of even length, orthogonal and time reverse of each other. They have $\frac{1}{4}$ and $\frac{3}{4}$ samples delay (indicated by q and $3q$) respectively to satisfy perfect reconstruction. The overall frequency response at any given level is therefore identical for the two trees, and each parent wavelet coefficient below the finest level samples lies exactly halfway between its two children. When the transform is extended to 2D, we filter the rows of the image with the two trees, then filter the columns with the two trees.

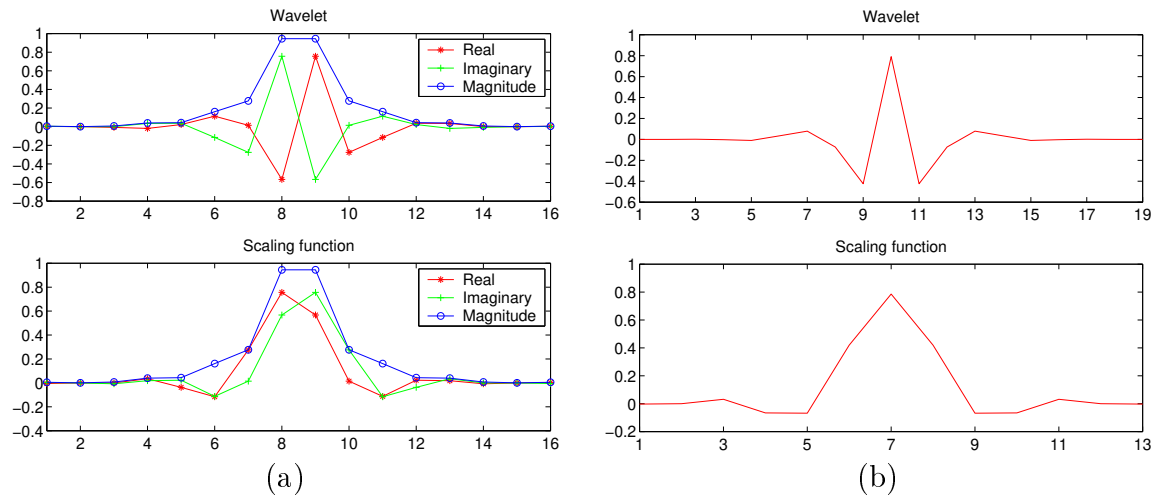
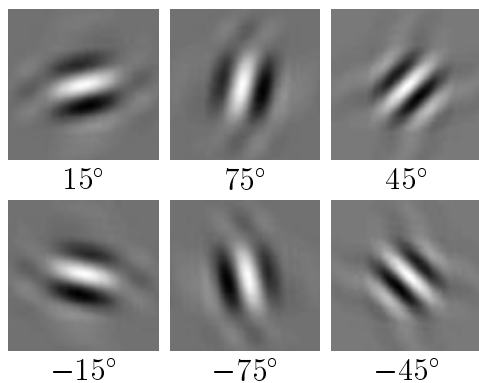


Figure 3.3: Wavelets and scaling functions for the CWT (a) and the DWT (b). The CWT filters are orthogonal and the DWT ones are zero phase, which means the transform can only be biorthogonal. Therefore the frequency responses for the DWT analysis and reconstruction filter bank will be different, unlike the CWT case, where they are identical.

(a) CWT 2D Filter Impulse Responses



(b) DWT 2D Filter Impulse Responses

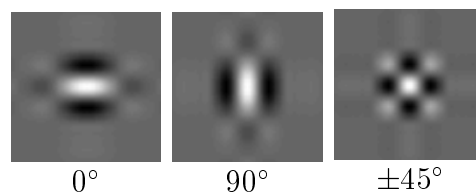


Figure 3.4: 2-D filter impulse responses of the CWT (a) and the DWT (b). Note that the CWT can distinguish between the two diagonal directions while the DWT cannot because all the filters are real and separable.

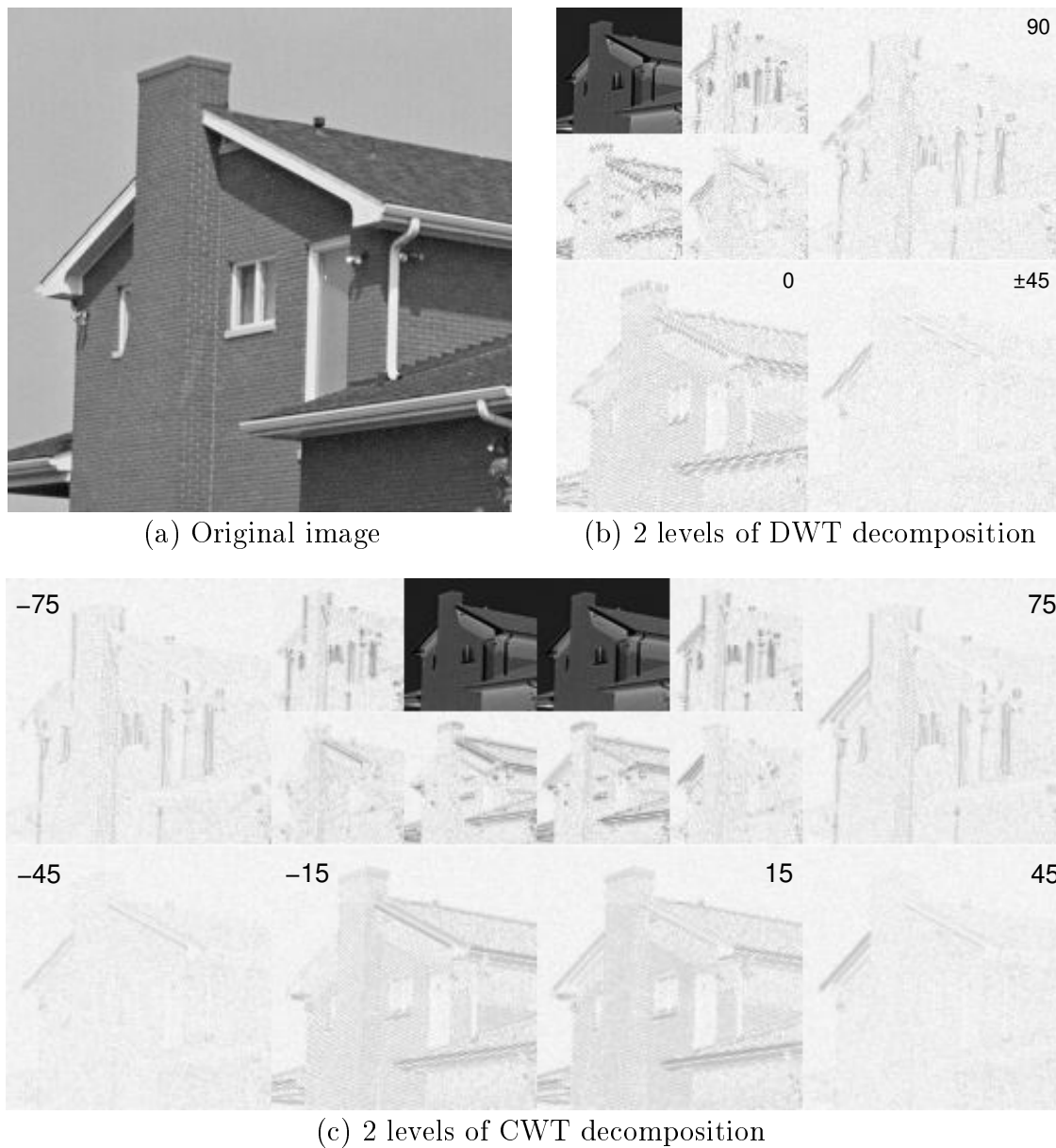


Figure 3.5: Examples of the DWT (b) and the CWT (c) decomposition of the House image (a). Only two levels of decomposition are shown in each case. The orientation of the corresponding filter is shown in the corner of each subband. The darkness of each subimage represents the magnitude of the wavelet coefficients. The dark images correspond to the lowpass wavelet coefficients. The contrast of the images has been enhanced for illustration purpose. The ability of the CWT (but not the DWT) to separate the two diagonal directions is evident in the figures.

3.3 The Human Visual System (HVS) and its relationship with the CWT

Recall from chapter 1 that two of the main requirements of digital watermarks are that the embedded signal should be imperceptible and robust at the same time (except for authentication watermarks). This requires the watermark energy to be adapted to the image content. Therefore it is important to have an understanding of the human visual system (HVS) in order to develop a visual model for watermark embedding.

Human visual perception is non-linear and is strongly dependent on the frequency as well as the orientation of the stimuli [157]. Our eyes are more sensitive to the difference in the intensity of the visual stimuli (i.e. contrast) than the absolute intensity of the stimuli themselves. Furthermore, our visual sensitivity decreases in the presence of another stimulus of similar frequency and orientation, a process known as *masking*. *Contrast sensitivity* and *masking* are the two main components of the human visual model.

3.3.1 Contrast sensitivity

A commonly used contrast in image processing is the *Weber contrast*, followed from Weber's law [70], which is defined as follows:

$$C_w = \frac{\Delta I}{I}, \quad (3.3)$$

where I is the intensity of the stimulus. Figure 3.6 (from [100]) shows the normalized detection threshold contrast for different intensities. We see that the normalized detection threshold is relatively constant for a wide range of intensities but increases for very bright or very dark background. However, if the data acquisition device performs gamma correction on the captured data, the detection threshold⁴ should be roughly independent of the background signal intensity. This is because gamma correction is close to logarithmic (for $\gamma < 1$) for a large range of intensities. Since we are concentrating on watermarking of monochrome images in this thesis, intensity of the stimuli corresponds to the luminance of the image. The chrominance contrast can be similarly defined for colour images.

The simple Weber contrast defined above only works well for simple signals and

⁴The unnormalized detection threshold is the minimum value of ΔI , regardless of the value of the background intensity, above which the stimulus will be perceived.

a better contrast sensitivity model is required for real images. We can see that the Weber contrast depends on two factors: the local relative difference (ΔI) and the local mean (I). Peli [118] extended the definition of Weber contrast to natural images and defined the contrast as the ratio of a local bandpass filter to a local lowpass filter:

$$C_p(x, y) = \frac{bp_f(x, y)}{lp_f(x, y)}, \quad (3.4)$$

where $bp_f(x, y)$ and $lp_f(x, y)$ are outputs of the bandpass and lowpass filters of frequency band f at location (x, y) . Using the same argument regarding gamma corrected inputs as before, we can deduce that the approximate contrast of gamma corrected images can be computed using the output of the bandpass filters only. However, using bandpass filters alone is not enough for computing local contrast, as the *masking* process also depends on the orientation of masker and the stimuli. Watson proposes the *Cortex Transform* [157], which simulates the response of human visual neurons. The cortex transform decomposes an image into multiple scales as well as multiple orientation ($0^\circ, 90^\circ, 45^\circ$ and -45°) subbands. The filters used in the transform are *Gabor Filters* (3.2). We can infer that the CWT is a good approximation of the human visual model⁵, and that the wavelet filter outputs are closely related to visual neuron response. Our visual sensitivity also varies with different frequencies and orientations [37]⁶, thus we should adjust the watermark energy accordingly in different frequency subbands in order to maximize the embedding energy while maintaining imperceptibility.

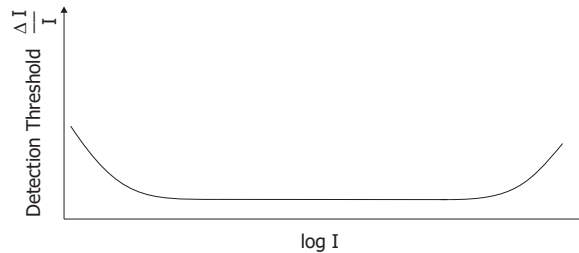


Figure 3.6: Typical human contrast sensitivity in the absence of masking. The normalized detection threshold contrast (Weber contrast) is constant for a large range of intensities and increases for very strong or very weak stimuli.

⁵The fact that the CWT has 6 subbands per resolution rather than 4 as in the cortex transform is not important. The important point is that the opposing diagonal directions are separated.

⁶A typical visual sensitivity curve can be found in Comes [37]. In general, our eyes are most sensitive at mid frequencies and relatively insensitive to high frequencies. We are also better at identifying vertical and horizontal features than diagonal features.

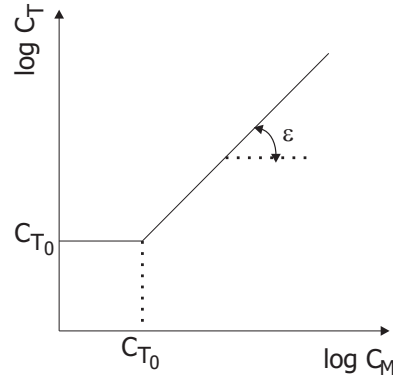


Figure 3.7: A commonly used transducer function. C_T and C_M are the detection threshold contrast and the masker's contrast respectively.

3.3.2 Masking

The masking phenomenon is usually modeled by relating the contrast of the masker to the contrast of the stimuli. For a given masker contrast, the detection threshold contrast C_T is given by the so called *transducer function*. A commonly used transducer function [37] is shown in figure 3.7, which can be written as follows:

$$C_T = \begin{cases} C_{T_0} & \text{if } C_M \leq C_{T_0}, \\ C_{T_0} \left(\frac{C_M}{C_{T_0}}\right)^\epsilon & \text{if } C_M > C_{T_0}. \end{cases} \quad (3.5)$$

When the masker's contrast C_M is below the critical contrast C_{T_0} , the detection threshold stays constant at C_{T_0} . As soon as the masker's contrast exceeds this critical value, the detection threshold increases with the masker's contrast. The exponent ϵ typically has value between 0.6 and 1. In reality, the detection threshold actually decreases slightly when the masker's contrast is close to C_{T_0} , a phenomenon known as *facilitation*. However, its effects are small and can normally be ignored. It is also measured [37] that the angular bandwidth of masking is around 30 degrees, which means almost no masking occurs if the stimuli occur in a different subband with respect to the masker (angular bandwidth of a CWT subband is about 30 degrees).

3.3.3 Visual model in the CWT domain

Using the definitions of contrast and masking in the previous sections, we can derive a simple yet effective visual model for watermark embedding in the CWT domain.

We use a modified transducer function of the form:

$$C_T = \sqrt{C_M^2 + C_{T_0}^2}. \quad (3.6)$$

Effectively we are taking ϵ in (3.5) as 1. Since we assume the input device performs gamma correction on the input data, we compute the contrast C_M as a function of the wavelet filter outputs. As mentioned in the previous section, the masking effect is localised in one orientation subband as long as the angular bandwidth is not much smaller than 30 degrees, we can therefore compute masking for each wavelet subband independently. Putting everything together we obtain the overall weighting factor as follows:

$$g_{l,\theta}(u, v) = \beta \sqrt{k^2 |x_{l,\theta}|^2 + C_{T_0}^2}, \quad (3.7)$$

where $g_{l,\theta}(u, v)$ is the allowable gain for the watermark at transform level l and orientation θ at location (u, v) ⁷. $|x_{l,\theta}|^2$ is the lowpass filtered (a 3×3 Gaussian window of standard deviation 0.5 is used) version of the squared amplitude of the CWT coefficients of subband (l, θ) in a 3×3 neighbourhood centered at (u, v) . k is a *subband dependent* constant for calculating the masker's contrast. C_{T_0} is also subband dependent. β is a function of the lowpass CWT coefficients and allows for correction of the detection threshold due to the variation in background luminance.

Let us see how (3.7) exploits the HVS. In the absence of masking (for example, flat regions of an image), the watermark weight is approximately βC_{T_0} . Around an edge or textured areas where the wavelet coefficients are large, the term $k^2 |x|^2$ dominates the watermark weight. Performing watermark weighting separately in different subbands allows us to align the watermark signal along any oriented structures of the host image to ensure imperceptibility. There is an additional factor (see next chapter) α , which is constant for the whole image, and scales the whole watermark so it satisfies some energy constraint.

3.3.4 Image quality metric based on the HVS

Root mean squared (RMS) error and peak signal to noise ratio (PSNR) are the two commonly used image distortion measures in most existing watermarking algorithms. However, neither of them takes account of the HVS. In [147], van den Branden Lambrecht proposes a quality metric for colour images which involves decomposing the

⁷The subscripts l, θ will be dropped from now on whenever the notation is clear.

error image into multiple frequency bands and multiple orientations, just like the cortex transform, and the transducer function in figure 3.7 is used for computing masking. The error components are then divided by the masking factor, independently in each subband, and the final results are pooled together to give the perceived error. A generic image quality metric is shown in figure 3.8. Karunasekera *et al.* [76] use a similar approach to compute the perceived artifacts due to image coding, but the authors only used multiple orientation filters, *without* multiple scales. In this thesis, we consider the latter quality metric⁸ combined with the error pooling technique introduced in [147], which is defined as:

$$Error = \left[\frac{1}{N_c} \sum_i \left(\frac{1}{N_i} \cdot \sum |err_i|^n \right) \right]^{1/n}, \quad (3.8)$$

where err_i is the error from channel i ; N_i is the number of elements in channel i ; N_c is the number of channels. n is recommended to have a value of 4 to allow regions with larger unmasked errors to dominate the perceived error, because human observers tend to look for artifacts in images. If n is 2, error pooling reduces to RMS error. An alternative image quality metric is also considered, which employs a similar approach to that suggested by [147], except that the cortex transform and the transducer function are substituted by the CWT and our CWT visual model.

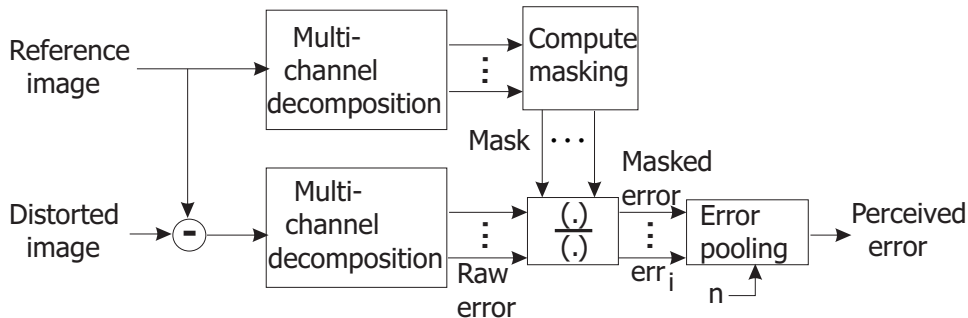


Figure 3.8: A generic image quality metric. The multi-channel decomposition can be multi-orientation only (as in [76]) or combined with multi-scale (as in [147]). The power n in error pooling determines how much emphasis is given to larger masked errors.

⁸An implementation of the quality metric proposed in [147] is not available due to copyright issues.

3.4 Empirical results for the CWT visual model

In this section, the procedures for identifying the various parameters in (3.7) are described. Two subjects, A and B (both 26 years of age with corrected short-sightedness), took part in the following tests. In each case, the values from the two subjects are averaged. We can also see that the results from the two subjects (tables 3.1 and 3.2) are very similar.

3.4.1 Finding C_{T_0} and the correcting factor β

A noise image of bipolar values ± 1 is generated in the spatial domain and its CWT coefficients in one of the subbands are kept (while others are set to zero), and the result is scaled by C_{T_0} , inverse transformed and added to a uniform mid grey image. C_{T_0} is adjusted until the noise is just visible. This is repeated for other subbands, each time using one subband only. Finally, all the subbands are used together and the various C_{T_0} are scaled down until the noise is no longer visible. The values of C_{T_0} for various subbands are listed in table 3.1. The luminance of the uniform background is then varied and the C_{T_0} are rescaled until the noise becomes just noticeable again. The relative change in just noticeable difference (JND) noise energy at various background luminance levels is shown in figure 3.9. We approximate β using quadratic regression as:

$$\beta = 4.46(|x_{dc}| - 0.56)^2 + 1.02, \quad (3.9)$$

where $|x_{dc}|$ is the amplitude of the level 4 lowpass CWT coefficient corresponding to the location concerned, which is normalised such that it has a value of 1/0 for a uniform white/black image.

3.4.2 Determining k

A bipolar noise image is again generated in the spatial domain and its CWT coefficients in a particular subband are scaled and combined with a sinusoidal grating of orientation and frequency corresponding to that subband. k is varied until the noise is visible in the presence of the grating. The procedure is repeated for each subband just like determining C_{T_0} . Finally, all the subbands are used and the noise is added to a test set of 16 images (found in appendix C and all are of size 256×256 pixels). All the k are scaled by k_0 until artifacts are just noticeable. The results are shown in table 3.2. We can see that k_0 does not vary greatly across different types of images

Level / Subband	Subj. A	Subj. B	Average
Level 1 $\pm 75^\circ, \pm 15^\circ$	1.7	1.3	1.5
Level 1 $\pm 45^\circ$	3.1	3.1	3.1
Level 2 $\pm 75^\circ, \pm 15^\circ$	0.7	0.8	0.75
Level 2 $\pm 45^\circ$	1.0	0.8	0.9
Level 3 $\pm 75^\circ, \pm 15^\circ$	1.0	1.1	1.05
Level 3 $\pm 45^\circ$	0.9	1.1	1.0

Table 3.1: Empirical C_{T_0} for various CWT subbands measured from two subjects: A and B. We do not use level 4 for watermarking as it tends to cause artifacts in the result image. We can see that the thresholds are higher for finer levels and that the thresholds for diagonal subbands are higher than the vertical/horizontal subband in general. This agrees with the model described in [37].

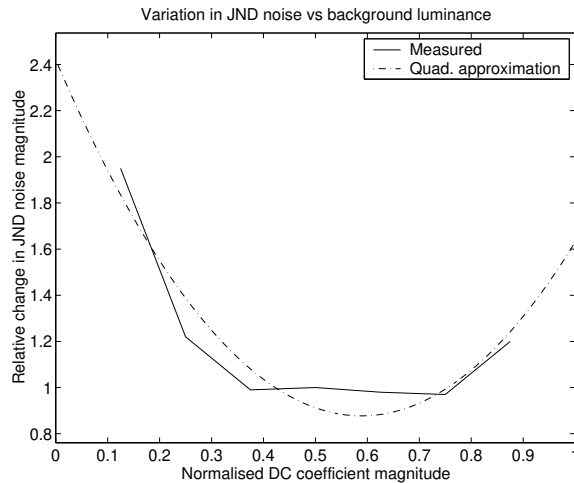


Figure 3.9: Relative change of just noticeable difference noise energy with respect to the normalised magnitude of the lowpass CWT coefficients.

(typical ones like Lena, textured ones like Bridge and smooth ones like Claire), and that the watermark energy is automatically adjusted according to the image content.

3.4.3 Comparison with other domains

In this section we compare the proposed visual model, equation (3.7), with visual models in the DWT and the DCT domains. Due to the lack of a satisfactory visual model for the DWT in the literature, we adopt our CWT visual model for the DWT domain and use Watson’s model [110, 158] (which is the one of the best to date) for the DCT domain. A summary of the three models is included in appendix A. A watermark is embedded into each of the 16 test images using the algorithm described in the next chapter, substituting in the appropriate transform domain and the corresponding visual model in each case. The overall gain is adjusted such that the

Level / Subband	Subj. A	Subj. B	Average
Level 1 $\pm 75^\circ, \pm 15^\circ$	$1.0k_0$	$1.0k_0$	$1.0k_0$
Level 1 $\pm 45^\circ$	$0.6k_0$	$0.5k_0$	$0.55k_0$
Level 2 $\pm 75^\circ, \pm 15^\circ$	$1.1k_0$	$1.0k_0$	$1.05k_0$
Level 2 $\pm 45^\circ$	$1.0k_0$	$1.1k_0$	$1.05k_0$
Level 3 $\pm 75^\circ, \pm 15^\circ$	$1.1k_0$	$1.1k_0$	$1.1k_0$
Level 3 $\pm 45^\circ$	$1.3k_0$	$1.4k_0$	$1.35k_0$

(a)

Image	k_0 , RMS wm	Image	k_0 , RMS wm
Lena	0.7, 3.2	Camera	0.55, 3.0
Baboon	1.0, 6.3	Claire	0.6, 2.1
Couple	0.8, 3.0	Tng2	0.9, 6.3
Pentagon	1.0, 3.3	Peppers	1.0, 3.2
Fishingboat	1.0, 4.0	Pills	1.0, 2.4
Indust	1.0, 3.9	F16	0.9, 3.9
Bridge	0.65, 3.8	Newyork	0.9, 9.0
Barbara	1.0, 3.8	Lochness	1.0, 3.2

(b)

Table 3.2: (a) Empirical k for various CWT subbands measured from two subjects: A and B. The overall scale k_0 as well as the corresponding RMS watermark for each of the 16 test images are shown in (b). The overall scale k_0 does not vary greatly across different types of image, which shows the CWT visual model is a good model which allows automatic adjustment of watermark energy according to the image content. For example, if the image has a lot of texture, the watermark energy is automatically increased.

watermark *energy* is approximately the same for each of the domains. The perceived error is then computed using the quality metrics in section 3.3.4 for each case. The results are shown in table 3.3.

In a nutshell, the image adaptability of the CWT and the DCT domain watermarks are very similar, and the DWT is slightly worse. The former quality metric suggests that the DCT watermarks produce lower perceived error than the CWT ones, whereas the CWT quality metric shows that the CWT watermarks look better in all cases (except for the Newyork image). The DWT watermarks produce higher perceived error under both metrics. There is only one subband representing the two opposing diagonal directions in the DWT, thus if we increase the watermark strength along a diagonal edge, we will inevitably produce a watermark oriented *orthogonal* to the edge, and this extra *component* will not be masked by the edge. This results in a higher perceived error. In the case of complex wavelets, one can separately orient a watermark along the two diagonal directions and the aforementioned problem does

Image	CWT		DWT		DCT	
	HVS err	CWT err	HVS err	CWT err	HVS err	CWT err
Lena	1.8	1.2	2.9	1.8	1.8	1.7
Camera	1.5	1.1	2.6	1.7	1.3	1.4
Baboon	1.7	1.4	2.8	1.9	1.4	1.6
Claire	1.7	1.2	2.7	1.7	1.4	1.4
Couple	1.7	1.3	2.7	1.7	1.4	1.4
Tng2	2.6	1.2	4.0	1.8	2.1	1.6
Pentagon	1.5	1.3	2.3	1.9	1.3	1.5
Peppers	2.0	1.5	3.0	2.0	1.7	1.7
Fishingboat	1.8	1.4	2.7	1.9	1.5	1.7
Pills	1.6	1.2	2.7	1.8	1.5	1.6
Indust	1.9	1.4	3.0	2.1	1.4	1.5
F16	1.5	1.3	2.0	2.0	1.3	1.4
Bridge	1.5	1.0	2.3	1.4	1.2	1.1
Newyork	2.1	1.9	3.2	2.5	1.6	1.7
Barbara	1.9	1.9	3.5	2.6	1.7	1.8
Lochness	1.4	1.2	2.1	1.7	1.1	1.3

Table 3.3: Results of comparison of HVS models in the CWT, DWT and the DCT domains. The model from Watson [110, 158] is used in DCT. The RMS watermark in the DWT and the DCT domains are adjusted so that they are the same as the RMS watermark in the CWT domain for each image (table 3.2b), and the perceived error is computed using [76] coupled with error pooling (HVS error) as well as using the CWT visual model (CWT error). The higher the perceived error, the more *visible* is the watermark.

not exist. Figure 3.10 shows the original ‘Lena’ and heavily watermarked versions of it in the CWT, DWT and DCT domains. The RMS watermark is around 9 in each case. Figure 3.11 shows the magnified version of the areas highlighted area in figure 3.10a with the corresponding watermark. We can see that only the CWT watermark manages to align itself completely along the dominant edges of the image. The quality of the images is compared *subjectively* and we found that the *subjective* quality of the CWT watermarked image is the best whereas the DCT watermarked image looks worst, mainly due to the undesirable block artifacts around the edges (see figure 3.11c). The DWT watermarked image looks slightly better than the DCT subjectively but a considerable amount of artifacts are still visible. Objectively, [76] returns a lower perceived error for the DCT marked image than for the CWT marked image. Therefore the quality metric in [76] may not be suitable in evaluating the quality of a watermark image and a multi-scale metric like the CWT model might be a better choice. We can also conclude that the CWT can adapt to an image better than the DWT and the DCT in general.



Figure 3.10: (a) Original Lena. (b) Lena watermarked in the CWT domain. (c) Lena watermarked in the DWT domain. (d) Lena watermarked in the DCT domain. All watermarked images have a RMS error of around 9.

3.5 Chapter summary

In this chapter, we reviewed the drawbacks of the real wavelet transform and described a dual tree implementation of the complex wavelet transform. We outlined how the CWT overcomes the problems with the conventional real wavelet transform. The human visual model was then described in terms of contrast sensitivity and masking. We demonstrated the close resemblance between the CWT and the HVS. A visual model for use in watermark embedding was derived and this model was compared with other visual models in the DWT and the DCT domains. Two quality metrics for assessing an image's quality were introduced and compared. Experimental results showed that, given a fixed level of RMS watermark, our CWT visual model can

adapt to the host image better than the DWT and the DCT. Therefore, we expect watermarking in the CWT domain to be more robust than in the DWT and the DCT domains and we will demonstrate this in the next chapter.

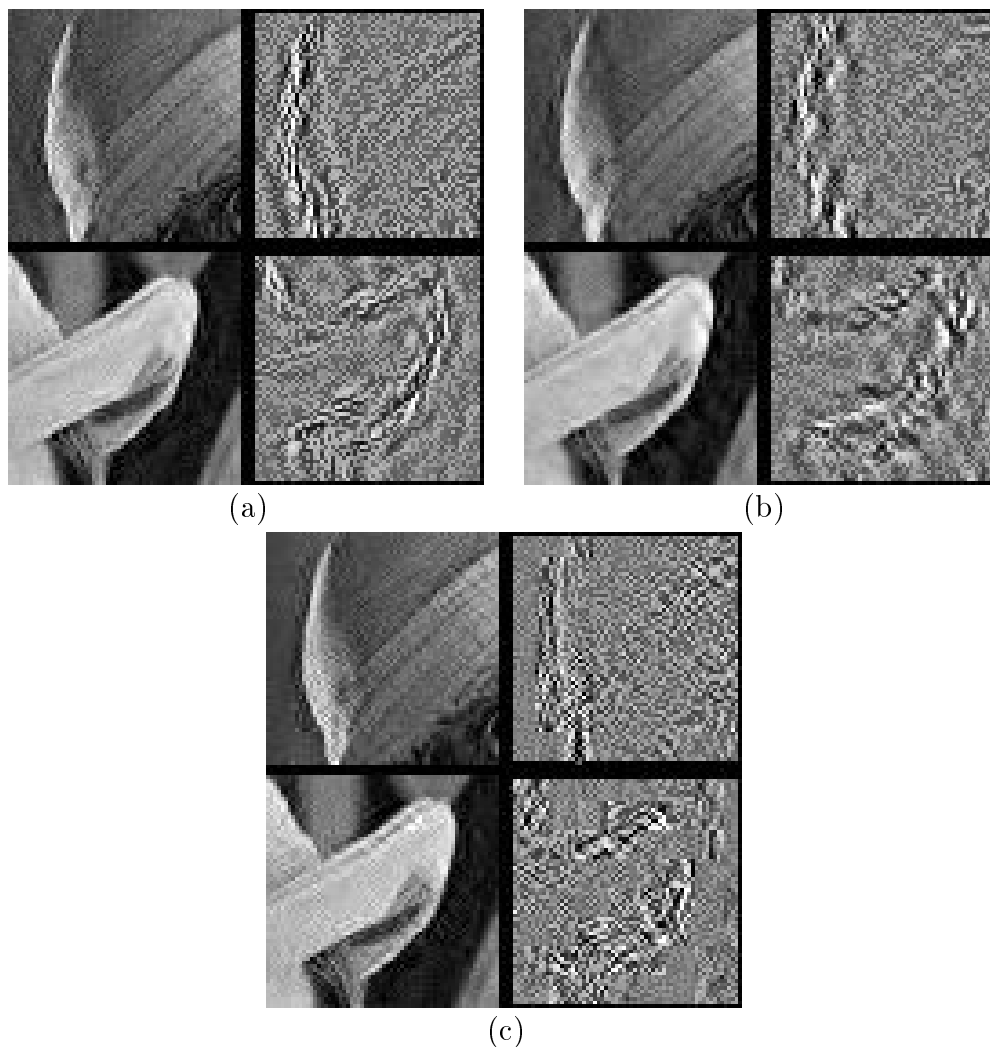


Figure 3.11: The two highlighted areas in fig. 3.10a and the corresponding watermark (amplified by a factor of 4). (a) The CWT domain. (b) The DWT domain. (c) The DCT domain. The DCT domain watermarked image looks worst subjectively whereas the CWT one looks best. We can also see that only the CWT watermark manages to align itself to the dominant edges in the image.

Chapter 4

Watermark Embedding and Extraction

4.1 Introduction

As mentioned in chapter 2, watermarking can be viewed as a digital communication problem through a very noisy channel¹. The most important aspect in digital communication is the modulation/demodulation process. Spread spectrum techniques are known to facilitate reliable communication in the presence of noise, which is why many existing watermarking algorithms employ spread spectrum as the modulation technique. The basic idea is that the information (payload) is coded with a (low power) pseudo-random code sequence, this leads to spreading of the information's frequency spectrum to increase its resilience against noise.

In our first design, we also employ the direct sequence spread spectrum approach. We begin this chapter by outlining our blind watermarking model and assumptions, and describe the watermark embedding algorithm. Then we investigate watermark decoding based on correlation. Three types of correlation decoder: *simple correlator*, *matched filter* and *modified matched filter* are considered and compared. The problem of combining outputs from multiple channels is also discussed. We compare blind watermarking in the CWT, DWT and the DCT domains and address the problem of watermark detection.

¹In this chapter, we consider the cover image as noise at the watermark decoder. In chapter 6, we will see the fact that the watermark embedder knows the cover image means we can design the watermark system such that the interference due to the cover image can be removed (almost) completely.

4.2 The watermark model and assumptions

The generic model for our blind watermark system is shown in figure 4.1. Our payload \mathbf{p} is an L -bit binary sequence which modulates some pseudo-random sequences. The watermark encoder constructs the watermark \mathbf{w} independently of the cover image \mathbf{x} (where we have omitted the fact that the watermark is usually shaped by a visual mask calculated from \mathbf{x} in order to simplify our model). The watermark is modulated by the payload and is simply added to the cover image, therefore the host image is additive noise with respect to the watermark. \mathbf{y} is the received (and possibly corrupted) image at the watermark detector/decoder. In this chapter, we treat each bit of the payload individually, in other words, binary symbols are used. The use of multilevel symbols was mentioned in section 2.4.2 in the context of error control codes. Typically, the host image is projected onto some space prior to watermark embedding. We consider using the CWT as the watermark domain here and we will compare it with the DWT and the DCT domains at the end of this chapter.

The transition function $p(y|x')$ in figure 4.1 represents the interference on the watermark, which includes any possible attacks on the watermarked image. Unfortunately, it is impossible to derive a generic model which describes all possible attacks. However, any attacks on the watermarked image should still render the image useful, and therefore the host image tends to dominate the interference at the decoder in most cases. We assume the interference due to the cover image in the transform domain follows a non-stationary Gaussian distribution. This assumption is valid as long as the transform domain, in which watermarking takes place, decorrelates the host image more or less completely. The wavelet transform (real and complex²) satisfies this requirement. The capacity of this channel is discussed in chapter 6. Finally, the watermark decoder performs two tasks. First it decides whether a watermark is likely to exist, and if so, proceeds to demodulate the payload.

4.3 Watermark embedding

4.3.1 Implication of CWT redundancy

As seen in the previous chapter, the CWT has a 4:1 redundancy in 2-D. This has important consequences on the design of the watermarking algorithm. First we will

²In the case of the CWT, the real and the imaginary parts of the coefficients are approximately Gaussian distributed.

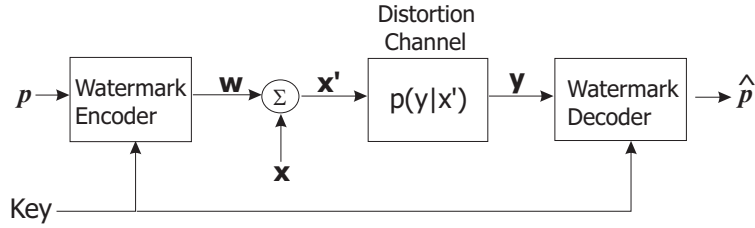


Figure 4.1: A simple watermark model. \mathbf{p} is the payload; \mathbf{x} is the cover image; \mathbf{w} is the watermark; \mathbf{x}' is the watermarked image and \mathbf{y} is the received and possibly corrupted image. $p(y|x')$ represents any attacks on the watermarked image. $\hat{\mathbf{p}}$ is the decoded payload.

define the concept of a *valid transform*. Let \mathbf{W}_F and \mathbf{W}_R be the forward and inverse transform matrices of some domain. If the transform is non-redundant, we have $\mathbf{W}_R = \mathbf{W}_F^{-1}$, and $\mathbf{W}_R = \mathbf{W}_F^T$ if the transform is orthogonal. If the transform is redundant, \mathbf{W}_F , \mathbf{W}_R will be rectangular and $\mathbf{W}_R \mathbf{W}_F = \mathcal{I}$, but $\mathbf{W}_F \mathbf{W}_R \neq \mathcal{I}$ in general, where \mathcal{I} is the identity matrix. A vector \mathbf{v} is said to be a *valid transform* if:

$$\mathbf{v} = \mathbf{W}_F \mathbf{W}_R \mathbf{v}. \quad (4.1)$$

Equation (4.1) is only satisfied if:

$$\mathbf{v} = \mathbf{W}_F \mathbf{x} \text{ for some } \mathbf{x}. \quad (4.2)$$

If the transform is non-redundant, (4.1) is always satisfied, and for any given \mathbf{v} , there is only one \mathbf{x} which satisfies (4.1), in other words, the mapping between the transform domain and the spatial/time domain is one to one and injective. On the other hand, the mapping is still one to one for redundant transforms, but it is no longer injective. In those cases, we can separate \mathbf{v} into two components such that:

$$\begin{aligned} \mathbf{v} &= \mathbf{v}_{valid} + \mathbf{v}_{orth}, \quad \text{and,} \\ \mathbf{v}_{valid} &= \mathbf{W}_F \mathbf{W}_R \mathbf{v}_{valid}, \\ \mathbf{0} &= \mathbf{W}_R \mathbf{v}_{orth}. \end{aligned} \quad (4.3)$$

If the transform is $m : 1$ redundant, then:

$$\frac{|\mathbf{v}_{orth}|^2}{|\mathbf{v}_{valid}|^2} \approx m - 1, \quad (4.4)$$

for an arbitrarily generated \mathbf{v} . If (4.2) is satisfied, then $\mathbf{v}_{orth} = \mathbf{0}$.

The essence of many existing spread spectrum based watermarking algorithms is the addition of a pseudo-random noise (PN) sequence (the watermark) to the host

image coefficients in some domain, and the domain is chosen to exploit masking in the human visual system. Typical choices include DCT, DFT, DWT and the spatial domain. All these domains (except the DFT) are critically sampled, which means (4.1) is satisfied for any random sequence \mathbf{v} . In the case of the DFT, the redundancy is well defined: the coefficients in any two quadrants are the complex conjugates of the other two. Hence we can design the PN sequence with this property when watermarking in the DFT domain.

Unfortunately, there is no well defined redundancy relationship for the CWT, and the relationship between coefficients at the same location in different subbands is signal dependent. Therefore, generating the PN sequence directly in the CWT domain does not work satisfactorily, because upon inverse transform, 75% of the energy of the added sequence will disappear on average. This follows directly from (4.3) and (4.4), and the fact that the CWT is 4 : 1 redundant in 2-D.

The redundancy of the CWT does not pose a problem as long as the watermark sequence satisfies (4.2). A simple way to achieve this is to use the CWT coefficients of a *random* image in the spatial domain as our PN sequence.

4.3.2 Embedding algorithm

A generic version of our proposed blind spread spectrum based watermark embedding scheme is illustrated in figure 4.2. The embedding algorithm consists of the following steps (assuming we are using the CWT as our transform domain):

1. A random image of ± 1 of the same size as the host image is generated based on a seed, which is in effect our private key.
2. The CWT of the host image and the random image are computed.
3. The scaling factors (visual mask) are computed from the host image's CWT coefficients using (3.7), independently for each subband.
4. The random image coefficients are modulated by the payload.
5. The modulated coefficients are scaled and then inverse transformed to form the watermark.
6. Finally the watermark is added to the host image in the spatial domain to obtain the marked image.

Unfortunately, both modulating the random image CWT coefficients and scaling them will in general render them as being from an *invalid* transform (equation 4.1 is not satisfied). However, experimental results showed that the loss of information by scaling the coefficients is negligible in practice, but this is not true for modulating the coefficients. In order to get round the modulation problem, only 1 bit is embedded in each random image. In other words, we either negate the random image to embed a ‘1’ or leave it as it is to embed a ‘0’. Multiple bits can be embedded by superimposing many random images on top of one another, as long as these random images are orthogonalised (e.g. via Gram-Schmidt (GS)). Ideally, one should orthogonalise the random images to be embedded, i.e. we apply the Gram-Schmidt process *after* the CWT coefficients are scaled. However, it was discovered that the performance of the system is worse than the case when the random images are orthogonalised *before* scaling. Table 4.1 shows the ratio of the energy of the inter symbol interference to that of the random images, when orthogonalisation is applied either before or after scaling the CWT coefficients. This ratio is defined as:

$$\text{energy ratio} = \frac{|\mathbf{s}' - \mathbf{s}|^2}{|\mathbf{s}|^2}, \quad (4.5)$$

where \mathbf{s} and \mathbf{s}' are the random images constructed in the watermark embedder and decoder respectively.

Energy ratio	No. of random images		
	1	2	4
GS before scaling	7.0×10^{-3}	5.0×10^{-3}	5.0×10^{-3}
GS after scaling	7.0×10^{-3}	0.5	1.3

(a)

Energy ratio	No. of random images		
	1	2	4
GS before scaling	1.2×10^{-2}	1.1×10^{-2}	1.1×10^{-2}
GS after scaling	1.2×10^{-2}	0.3	1.2

(b)

Table 4.1: Comparing inter symbol interference when orthogonalisation is applied before or after adaptive scaling of CWT coefficients. Energy ratio is defined as the ratio of the energy of the interference to that of the random sequence. (a) shows the energy ratio when Gram-Schmidt (GS) is applied to the random images either before or after adaptive scaling of the CWT coefficients in the case where the watermarked image is *not* attacked. (b) shows the same results when the watermarked image is compressed with JPEG at a quality factor of 70.

We can see that this ratio hardly varies with the number of random images, when orthogonalisation is applied *before* scaling. However, the discrepancies between the random images used in the embedder and the decoder increase dramatically if orthogonalisation occurs *after* scaling. The results hold even when the watermarked image is compressed. A possible explanation for this result is that the error in each vector, resulting from the slight inaccuracy of the visual mask estimated at the decoder, propagates to other vectors during the orthogonalisation process. Therefore we decided to orthogonalise the random images before they are scaled by the visual mask, and a small amount of interference between individual random images will remain due to the scaling.

Using a single random image for each bit of the payload solves the modulation problem, but we are now faced with greatly increased computational load. It is decided to use a block based CWT as a compromise and we embed bits independently in each block. The block size should be significantly larger than the wavelet and so we choose this to be 32×32 . However, using a block based CWT risks introducing blocking artifacts similar to those resulting from the block based DCT used in JPEG. Fortunately, no significant artifacts were observed in our simulations. The framework of the embedding algorithm remains the same as in figure 4.2. The embedding processing can be summarised as the following equation:

$$\begin{aligned} x'_i &= x_i + \alpha g_i b w_i \quad i = 0 \dots N - 1, \\ &= x_i + \alpha b s_i \quad \text{where } s_i = g_i w_i, \end{aligned} \tag{4.6}$$

where,

- x'_i is the watermarked transform coefficient.
- x_i is the original transform coefficient.
- α is the user specified watermark strength, which is constant throughout the image and controls the global watermark energy. α is usually about 1.
- g_i is the adaptive gain calculated based on the neighbourhood image characteristics (equation 3.7 for the CWT), and varies across subbands and scales. In the current implementation of the CWT scheme, 3 levels are used because modifying coefficients below level 3 tends to cause artifacts in smooth areas of the image.
- $b = \pm 1$ is the payload to be embedded.

- w_i is the watermark coefficient.
- N is the number of coefficients in the current subband in a particular block.

The above process is repeated in each subband for each 32×32 image block.

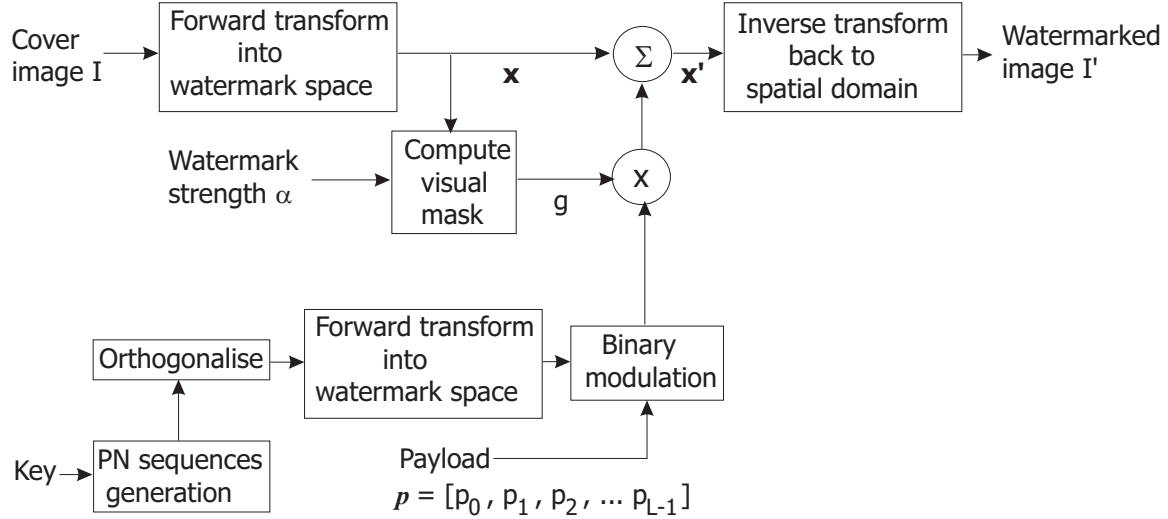


Figure 4.2: A generic spread spectrum based watermark embedder. The transform domain can be the DCT, the DWT or the CWT domains. The visual model is substituted accordingly.

4.4 Watermark decoding

The problem of watermark decoding is similar to the detection of known signal in noise. It is well known that under stationary white noise, the matched filter is optimal. However, the host image CWT coefficients, which are the dominant noise source at the decoder, are non-stationary. Hence the matched filter may no longer be optimal. We assume that a watermark is present and it is desired to minimise the bit error rate (BER) at the decoder output. We have the following hypothesis test:

hypothesis H_0 : a bit 0 is embedded ($b = 1$),

hypothesis H_1 : a bit 1 is embedded ($b = -1$),

which corresponds to (in the absence of any attack):

$$H_0 : \mathbf{x}' = \mathbf{x} + \alpha \mathbf{s}, \quad (4.7)$$

$$H_1 : \mathbf{x}' = \mathbf{x} - \alpha \mathbf{s}. \quad (4.8)$$

The log maximum likelihood ratio test results in the decision rule:

$$\log \left(\frac{p_{\mathbf{x}'}(\mathbf{x}'|H_1)}{p_{\mathbf{x}'}(\mathbf{x}'|H_0)} \right) \underset{H_0}{\overset{H_1}{\geq}} 0. \quad (4.9)$$

Most existing watermarking schemes assume the interference due to the cover image is distributed as *stationary* AWGN and (4.9) becomes the matched filter. A bit 0 will result in a positive correlation and vice versa.³ We assume the host and watermark CWT coefficients (where the real parts and the imaginary parts are concatenated together) in a particular channel are distributed as follows:

$$\begin{aligned} \text{host: } \mathbf{x} &\sim \mathcal{N}(0, h_i^2 \sigma_x^2) \quad 0, \dots, i, \dots, N-1, \quad h_i \geq 1 \quad \forall i, \\ \text{watermark: } \mathbf{s} &\sim \mathcal{N}(0, g_i^2 \sigma_w^2) \quad 0, \dots, i, \dots, N-1, \quad g_i \geq 1 \quad \forall i, \end{aligned} \quad (4.10)$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2 . Due to the imperceptibility requirement, the watermark energy is much smaller than the image, so $\sigma_x^2 \gg \sigma_w^2$. Also, h_i and g_i can vary greatly for different i , but when h_i is large, g_i will also be large. This is because wavelets tend to concentrate the image energy into a few large coefficients and the watermark is adapted to the image. We now consider the three types of correlator and compare their performance.

4.4.1 Simple correlator

The simple correlator computes the correlation between the received coefficients \mathbf{y} and the known (unscaled) watermark random sequence \mathbf{w} (coefficients of the random image). For the sake of clarity we have omitted the conjugate operator on \mathbf{w} in the remainder of our analysis. This does not affect our results because we can treat \mathbf{w} as a vector formed by the concatenation of the real parts and the imaginary parts of the CWT coefficients. The correlator r is defined as:

$$r_1 = \frac{1}{N} \sum_{i=0}^{N-1} y_i \cdot w_i. \quad (4.11)$$

This is clearly a suboptimal correlator but it is nevertheless useful as a comparison because many earlier watermarking systems use this form of correlation. If we derive

³In the case of complex coefficients like the CWT, we compute the correlation of the real part and the correlation of the imaginary part separately and add them together. This is easily implemented as $\Re\{\sum \mathbf{y} \cdot \mathbf{w}^*\}$, where \mathbf{y} are the received coefficients.

the decoder structure for a single channel from (4.9) the factor $1/N$ does not appear, but it is important if we have multiple channels carrying the same information and each channel has different number of coefficients. Without loss of generality, we take $b = 1$ and $\alpha = 1$ for the time being. The expectation and the variance of r can be derived as follows:

$$\begin{aligned}
E(r_1) &= \frac{1}{N} \sum_i E(y_i w_i) \quad \text{where } y_i = x_i + s_i, \\
&= \frac{1}{N} \sum_i (E(x_i)E(w_i) + E(s_i w_i)), \\
&= \frac{1}{N} \sum_i g_i \sigma_w^2.
\end{aligned} \tag{4.12}$$

$$\begin{aligned}
var(r_1) &= \frac{1}{N^2} \sum_i var(y_i w_i), \\
&= \frac{1}{N^2} \sum_i (E(x_i^2)E(w_i^2) + E(s_i^2 w_i^2) - E^2(y_i w_i)), \\
&= \frac{1}{N^2} \sum_i (h_i^2 \sigma_x^2 \sigma_w^2 + g_i^2 E(w_i^4) - g_i^2 \sigma_w^4), \\
&= \frac{1}{N^2} \sum_i (h_i^2 \sigma_x^2 \sigma_w^2 + 2g_i^2 \sigma_w^4),
\end{aligned} \tag{4.13}$$

where the last line follows from the fact that the Pearson Kurtosis of a Gaussian variable is 3. Since r is a sum of independent random variables, the *Central Limit Theorem* applies. (This applies to the other two correlators which follow as well.) r will distribute approximately as Gaussian with mean $E(r)$ and variance $var(r)$. The statistics of r for a bit 0 and a bit 1 will therefore have identical variances and opposite means (figure 4.3). The bit error rate (BER) (assuming equal probability of bit 0 and bit 1) can be easily obtained as:

$$P_{error} = Q\left(\frac{E(r)}{\sqrt{var(r)}}\right), \tag{4.14}$$

where

$$Q(x) = \frac{1}{\sqrt{2\pi}} \int_x^\infty e^{-\frac{x^2}{2}} dx. \tag{4.15}$$

Hence we can compare the performance of the different correlators in terms of the ratio $\rho = E(r)/\sqrt{\text{var}(r)}$. The bigger ρ is, the smaller is the BER. The signal power to noise ratio (SNR) at the correlator output is defined as

$$\begin{aligned} SNR &= \frac{E(r)^2}{\text{var}(r)}, \\ &= \rho^2. \end{aligned} \tag{4.16}$$

4.4.2 Matched filter

The decoder first estimates g_i from \mathbf{y} and scales w_i with the estimated weights before correlating with \mathbf{y} . For the purpose of analysis, we assume the decoder knows g_i exactly. In practice, the estimated weights \hat{g}_i will be very similar to g_i since they depend on the CWT coefficient magnitudes in a local neighbourhood (equation 3.7), and the watermark coefficients are random and small. The decoder is defined as:

$$r_2 = \frac{1}{N} \sum_{i=0}^{N-1} y_i \cdot s_i. \tag{4.17}$$

We can derive the statistics of the correlator just as before (again assuming both b and α are 1):

$$E(r_2) = \frac{1}{N} \sum_i g_i^2 \sigma_w^2, \tag{4.18}$$

$$\text{var}(r_2) = \frac{1}{N^2} \sum_i (h_i^2 g_i^2 \sigma_x^2 \sigma_w^2 + 2g_i^4 \sigma_w^4). \tag{4.19}$$

The matched filter gives bigger weights to larger coefficients and the terms associated with these coefficients will dominate the sum in (4.17). We prove in appendix B.1, that the existence of a few large coefficients in the sequence will lead to a lower SNR at the matched filter output compared with the other two decoders described in this section. This can be explained by the fact that the matched filter amplifies the non-stationarity of the host distribution.

4.4.3 Modified matched filter

The modified linear correlator is a hybrid combination between the two correlators we have considered so far. We would like to convert the sequence such that the noise is approximately stationary so we can use the matched filter. Instead of scaling the watermark coefficients with the estimated weight, the received coefficients are *divided* by the weights. However, if the weight gets arbitrarily small, the received signal will be highly magnified, which is undesirable. This is similar to the problem in deconvolution of images, where one has to divide the image spectrum by the estimated blurring filter spectrum. There is one difference, however. In the case of watermarking, the weight is limited by the detection threshold, i.e. βC_{T_0} for the CWT and the DWT domains, and by t_{ijk} for the DCT (see appendix A) and can *never* be zero. Nevertheless, one can use a similar technique to *pseudo-inverse filtering*, which is to *multiply* the receive sequence by:

$$f = \frac{g}{g^2 + \kappa^2}, \quad (4.20)$$

and then compute the correlation between $(\mathbf{f} \cdot \mathbf{y})$ and \mathbf{w} . When $g \gg \kappa$, this is almost the same as inverse scaling, and $f \approx g/\kappa^2$ when g is small. Our simulations show that the performance of the decoder does not depend strongly on κ , and that it is unnecessary to use (4.20) for both real and complex wavelet domains. In the case of the DCT, setting $\kappa = 1$ gives good results. In all our simulations in this chapter, we use normal inverse scaling for the CWT and the DWT and use (4.20) with $\kappa = 1$ for the DCT.

By dividing the received coefficients by the estimated weights, we reduce the watermark channel to an additive one, where the embedded signal strength is *nearly* unrelated to the host coefficients. Thus correlating this *inversely scaled* sequence with the watermark sequence is *almost* equivalent to the matched filter. We again assume the decoder knows the weights exactly for the purpose of analysis. The decoder is defined as

$$r_3 = \frac{1}{N} \sum_{i=0}^{N-1} \frac{y_i}{g_i} \cdot w_i. \quad (4.21)$$

Deriving the expectation and variance in the same way as before, we get:

$$E(r_3) = \sigma_w^2, \quad (4.22)$$

$$\text{var}(r_3) = \frac{1}{N^2} \sum_i \left(\frac{h_i^2}{g_i^2} \sigma_x^2 \sigma_w^2 + 2\sigma_w^4 \right). \quad (4.23)$$

In the special case where $h_i = g_i$, (4.23) becomes:

$$\text{var}(r_3) = \frac{1}{N} (\sigma_x^2 \sigma_w^2 + 2\sigma_w^4). \quad (4.24)$$

In other words, the correlator becomes *completely independent* of the non-stationarity. In practice, h_i will not exactly equal to g_i , but we can assume that $h_i \approx g_i$ throughout the sequence, if the watermark is to adapt closely to the host image. We show in appendix B.1 that the modified matched filter performs best theoretically while the matched filter has the worst performance due to the non-stationarity of the noise. Before we verify their performance experimentally, the problem of combining information from different channels is addressed.

4.4.4 Multiple channel decoding

Many spatial watermarking schemes (for example, [74, 91]) prefilter the watermarked image before correlating with the known watermark sequence. The purpose of the preprocessing is to remove most of the energy due to the host image in the low frequency components as it is the dominant interference at the decoder. Preprocessing improves the SNR at the decoder output significantly, but it is not necessary if the decoder operates in the frequency domain, as we can combine the outputs from different frequency bands to obtain the most likely watermark, but how should we combine these outputs?

Suppose we have K *independent* channels with outputs r_i ($i = 0, \dots, K-1$), whose expectations and variances are μ_i ($i = 0, \dots, K-1$) and σ_i^2 ($i = 0, \dots, K-1$). We can model these outputs jointly as a multivariate Gaussian where a bit 0 and a bit 1 will have opposite mean (μ_i and $-\mu_i$ ($i = 0, \dots, K-1$)) and identical variance. Using the ML decoder, we get:

$$\frac{p_{\mathbf{y}}(\mathbf{y}|H_1)}{p_{\mathbf{y}}(\mathbf{y}|H_0)} = \frac{(2\pi)^{K/2} |\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{r} - \mathbf{r}_\mu)^T \Sigma^{-1} (\mathbf{r} - \mathbf{r}_\mu))}{(2\pi)^{K/2} |\Sigma|^{-\frac{1}{2}} \exp(-\frac{1}{2}(\mathbf{r} + \mathbf{r}_\mu)^T \Sigma^{-1} (\mathbf{r} + \mathbf{r}_\mu))} \underset{H_0}{\overset{H_1}{\geq}} 1, \quad (4.25)$$

where Σ is a diagonal matrix whose diagonal elements are σ_i^2 ; $\mathbf{r} = [r_0, \dots, r_{K-1}]$ and

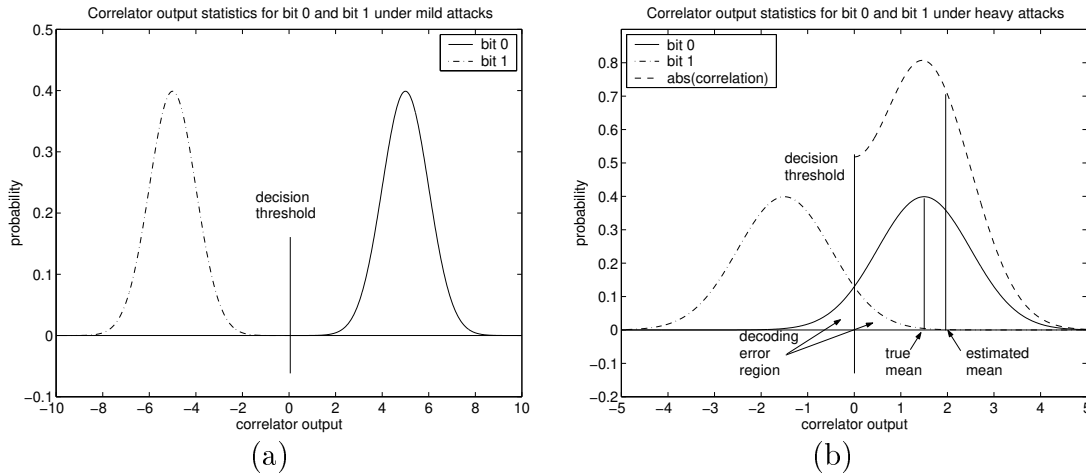


Figure 4.3: Correlator output statistics for bit 0 and bit 1 under mild (a) and heavy (b) attacks on the watermarked image. The figures have been arbitrarily normalised for illustration purpose. We can see that, under high level of attacks, the two distributions overlap. The region where decoding error will occur is also highlighted. If we use the absolute value of the correlations of the payload for estimating the mean and variance of the correlator outputs (required for combining outputs from different channels), we will overestimate the weights for the more unreliable channels. One should therefore use the correlations of the reference watermark for estimating the mean and variance.

$\mathbf{r}_\mu = [\mu_0, \dots, \mu_{K-1}]$. The log-likelihood ratio of (4.25) is:

$$\sum_{i=0}^{K-1} \frac{r_i \mu_i}{\sigma_i^2} \underset{H_0}{\overset{H_1}{\geq}} 0. \quad (4.26)$$

Hence the required weight for each channel is the ratio $\nu_i = \mu_i/\sigma_i^2$. If the correlation from each channel is not averaged (i.e. without the $1/N$ factor), then the weights will become $\nu_i = \mu_i/(N_i\sigma_i^2)$, with N_i being the number of coefficients in the i^{th} channel. The statistics depend on the embedded bit which we in general do not have prior knowledge of. This is illustrated in figure 4.3. When the watermark image undergoes no or mild attacks, the output statistics for bit 1 and bit 0 are well separated, and we can measure μ_i and σ_i^2 reliably using the *absolute* value of the correlations of individual bit. However, when the watermarked image suffers from heavy attack, the two statistics overlap (figure 4.3b). Using the absolute value of the correlation in this case leads to an overestimate of μ_i and an underestimate of σ_i^2 , which results in higher weights given to more unreliable channels. We therefore use a reference watermark, which is orthogonal to our payload, in order to estimate the channel characteristics. This assumes the interference on the payload watermark is similar to that on the reference one. The decoder decodes the reference watermark as a fixed

payload, and μ_i, σ_i^2 are estimated from the corresponding correlator outputs. This reference watermark is also needed for watermark *detection*, where the reference is detected as a whole (see later), when no error control code is used. The overall SNR is given by:

$$\begin{aligned}
 SNR_{overall} &= \frac{E(r_{overall})^2}{var(r_{overall})}, \\
 &= \frac{(\sum_{i=0}^{K-1} \nu_i \mu_i)^2}{\sum_{i=0}^{K-1} \nu_i^2 \sigma_i^2}, \\
 &= \sum_{i=0}^{K-1} \frac{\mu_i^2}{\sigma_i^2}, \\
 &= \sum_{i=0}^{K-1} SNR_i, \quad \text{where } SNR_i \text{ is the SNR of channel } i. \quad (4.27)
 \end{aligned}$$

Hence the overall SNR is equal to the sum of SNR from each channel. As a result, using multiple channels is always better than using a single one.

We can now compare the performance of the three correlators. A 64-bit watermark is embedded in the ‘Lena’ image and the BER and the SNR of the three decoders are compared. The watermark strength α is varied from 0.1 to 1 and 30 watermarks are embedded and decoded for each parameter setting.

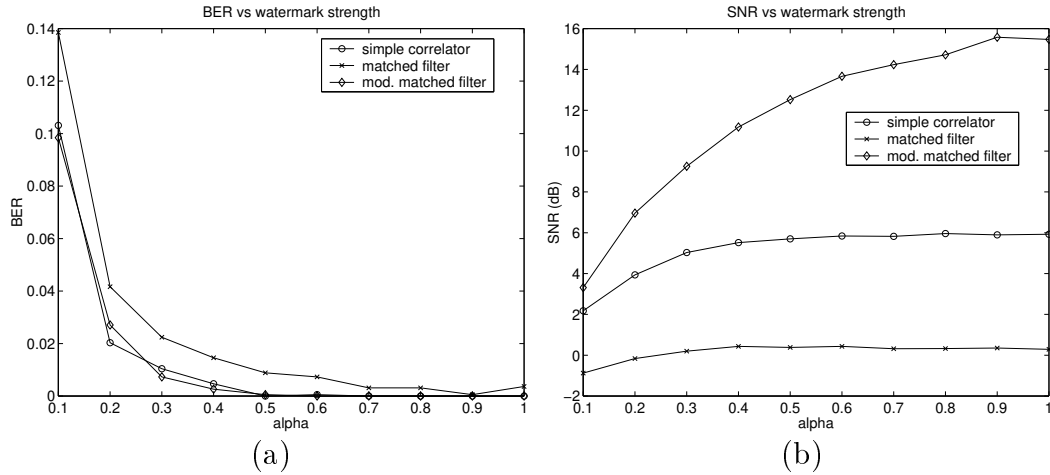


Figure 4.4: Comparison of the (a) BER and (b) SNR of the three correlators discussed in section 4.4, simple correlator, matched filter and modified matched filter. The modified matched filter has the best performance, just as predicted by theory.

Figure 4.4 shows the BER and the SNR using the three decoders combined with multiple channel decoding. We can see that the experimental results match with our predictions. The superior performance of the modified matched filter over the simple

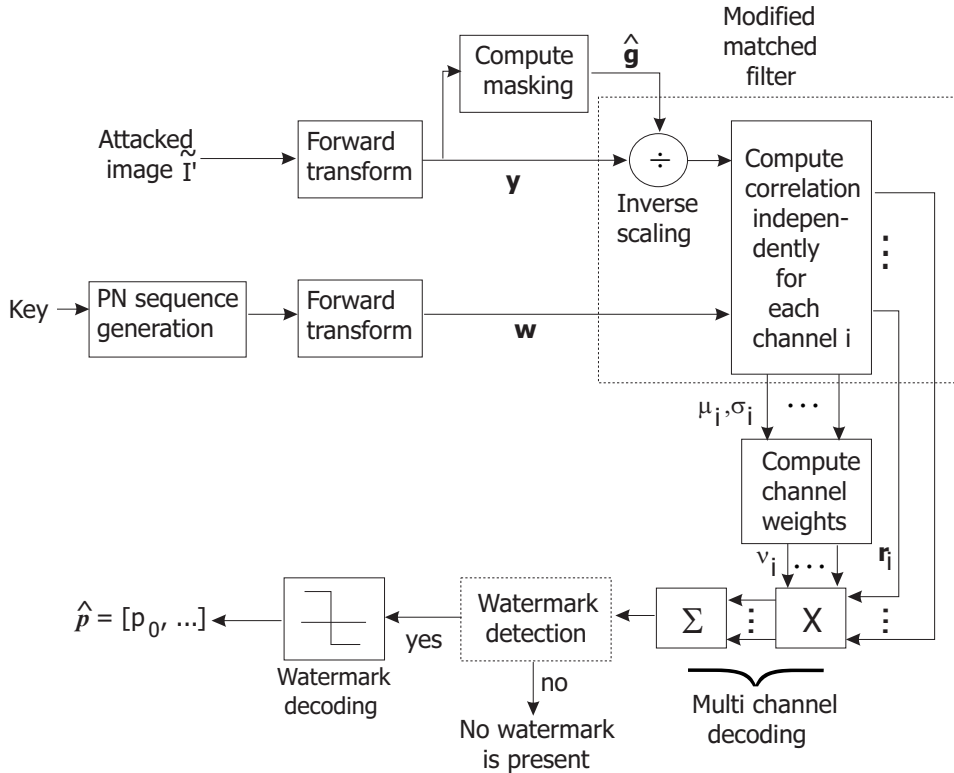


Figure 4.5: A generic blind spread spectrum watermark decoder using modified matched filter. Correlation is always performed regardless whether the suspected image is actually watermarked or not. The watermark detector uses the correlator outputs to determine whether there is likely to be a watermark or not. In the simulations presented in this chapter, the attacked image is always assumed to be watermarked and we skip the watermark detection step.

correlator is only apparent when their respective SNR is compared. The final decoder is illustrated in figure 4.5.

4.5 Comparison with other domains

In this section we compare our proposed blind spread spectrum based watermarking scheme in the CWT, DWT and the DCT domains. The DWT based watermark scheme proposed by Barni *et al.* [17] was also used as a comparison. Barni's scheme is chosen because it is also based on embedding a pseudo-random sequence in the wavelet domain, and uses a perceptual model [99] for weighting the watermark. However, the scheme in [17] is only a *yes/no* watermark⁴ and we need to modify our algorithm to

⁴During research the author failed to find a publicly available implementation of a blind multi-bit spread spectrum based watermark scheme in the wavelet domain which also incorporates a human visual model. Therefore only a *yes/no* watermark scheme is used for comparison.

make a proper comparison. Measuring BER is not possible for a yes/no watermark. Nevertheless, we can measure the output SNR (defined in equation (4.16)) of the detector, where $E(r)$ and $var(r)$ are now given by the following:

$$E(r) \approx \frac{1}{N} \sum \mathbf{z} \cdot \mathbf{w}, \quad (4.28)$$

$$var(r) \approx \frac{1}{N} var(z), \quad (4.29)$$

where \mathbf{z} and \mathbf{w} are the sequences of *inversely scaled* received coefficients and the *unscaled* watermark sequence respectively and N is the total number of coefficients. Our algorithm is modified to embed only 1 bit. The watermark is correlated independently in each subband, and combined according to (4.26) to obtain the overall correlation. The overall SNR is then calculated using (4.27). In [17], the authors fix the false positive probability to be 10^{-8} , which leads to a threshold of $3.97\sqrt{2 \cdot var(r)}$. This corresponds to a threshold SNR of 15dB. If the output SNR is below this level, the watermark will be missed with a probability of 0.5.

	0	1	2	3	4	5	6	7
0	DC	7	4		1			
1	9	8						
2	6		5					
3								
4	3				2			
5								
6								
7								

Figure 4.6: The partitioning of the DCT coefficients into 9 channels to mimic the way the DWT partitions the frequency plane. We use multiple DCT blocks to embed 1 bit, such that the number of coefficients used to encode one bit is the same as in the DWT. The CWT has 4:1 redundancy and therefore has 4 times as many coefficients for encoding a particular bit.

Two sets of tests were conducted. We first compare the performance of the proposed spread spectrum scheme in the CWT, DWT and the DCT domains. The embedder depicted in figure 4.2 is used in each case with the transform domain (and the visual model) substituted with CWT, DWT, DCT accordingly. In the case of the DCT, the coefficients in an 8×8 block are divided into 9 groups as shown in figure 4.6 to mimic the way the frequency plane is partitioned in the wavelet domain. Multiple blocks are used to embed 1 bit so that the total number of coefficients used to encode 1 bit is the same as in the DWT. During the decoding process, coefficients

in the same group from different blocks are concatenated together to form a channel and the outputs from different channels are combined as discussed before. In the case of wavelet transforms, each subband is a separate channel, and since we are using 3 levels of transform, we have 9 channels for the DWT and 18 for the CWT. A 128-bit watermark (uncoded) is embedded and the gain is adjusted such that the RMS watermark for the three domains is approximately the same⁵. The watermarked images then undergo attacks and the modified matched filter is used to decode the watermark. The attacked image is always assumed to be watermarked and no watermark detection is performed before decoding. In the second set of tests we compare the 1-bit version of our scheme in the CWT domain with the DWT scheme in [17] and the output SNR of the two schemes are compared.

4.5.1 Simulation results

We evaluate the performance of our watermark system under JPEG compression, JPEG2000 compression, additive white Gaussian noise (AWGN), mean and median filtering. Attacks which alter the geometry of the image will be discussed in the next chapter. Three test images: Lena (typical image), Baboon (significant high frequency components) and Pills (significant flat areas) were used. The watermarks in the 3 domains for the three images are shown in figures 4.7, 4.8 and 4.9 respectively. Only the CWT watermark manages to align itself to the dominant edges in each case.

JPEG compression Figures 4.10 (a-c) show the BER after the watermarked images (Lena, Baboon and Pills) have been compressed by JPEG with quality factor (QF) from 10 to 100, whereas figure 4.13 (a-c) show the SNR for the 1-bit version of the scheme under the same attack for the three images. One can see that the CWT watermark is in general more robust than DWT and DCT watermarks under JPEG compression. The performance of DWT and DCT watermarks under JPEG are very similar, with the DCT watermark being more robust at higher level of compression for the Lena and Pills images. The BER for the Baboon image is lower than the other two images as a higher level of watermark can be embedded in the image without causing significant perceptual error. Figures 4.11 (a and b) show the same test for

⁵In each case, one third of the watermark energy is allocated to a reference watermark, which is used in the estimation of individual channel characteristics during decoding. This allocation is somewhat arbitrary. Since we are comparing the relative performance of the three domains, the exact energy allocation between the payload and the reference is not important. For example, Kundur *et al.* in [89] divide the watermark energy equally between the reference and payload.

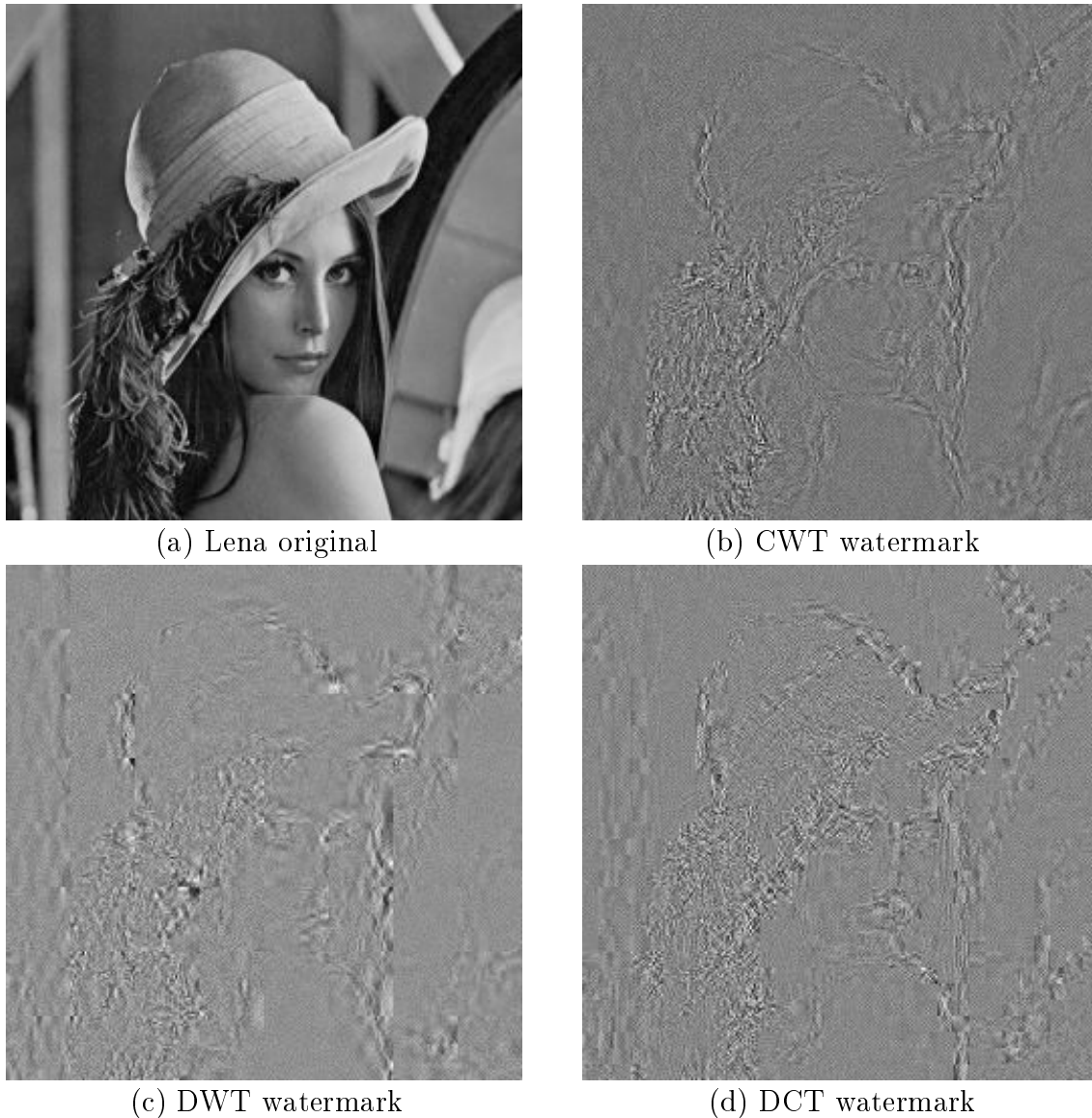


Figure 4.7: Lena and the watermark (enhanced by a factor of 8) in the CWT, DWT, DCT domains

Lena when the perceived error produced by the watermark is constrained to be the same in the three domains, in which case the DWT has the worst performance. This is because the DWT watermark produces higher perceived error than the other two domains (table 3.3 in section 3.4.3) and the watermark energy has to be lowered to produce the same perceptual error. Consequently, the CWT watermark has even better performance compared with the DWT and the DCT in practice, when the subjective quality of the watermarked image (rather than the watermark energy) is constrained. However, due to the lack of a standardised image quality metric, we will

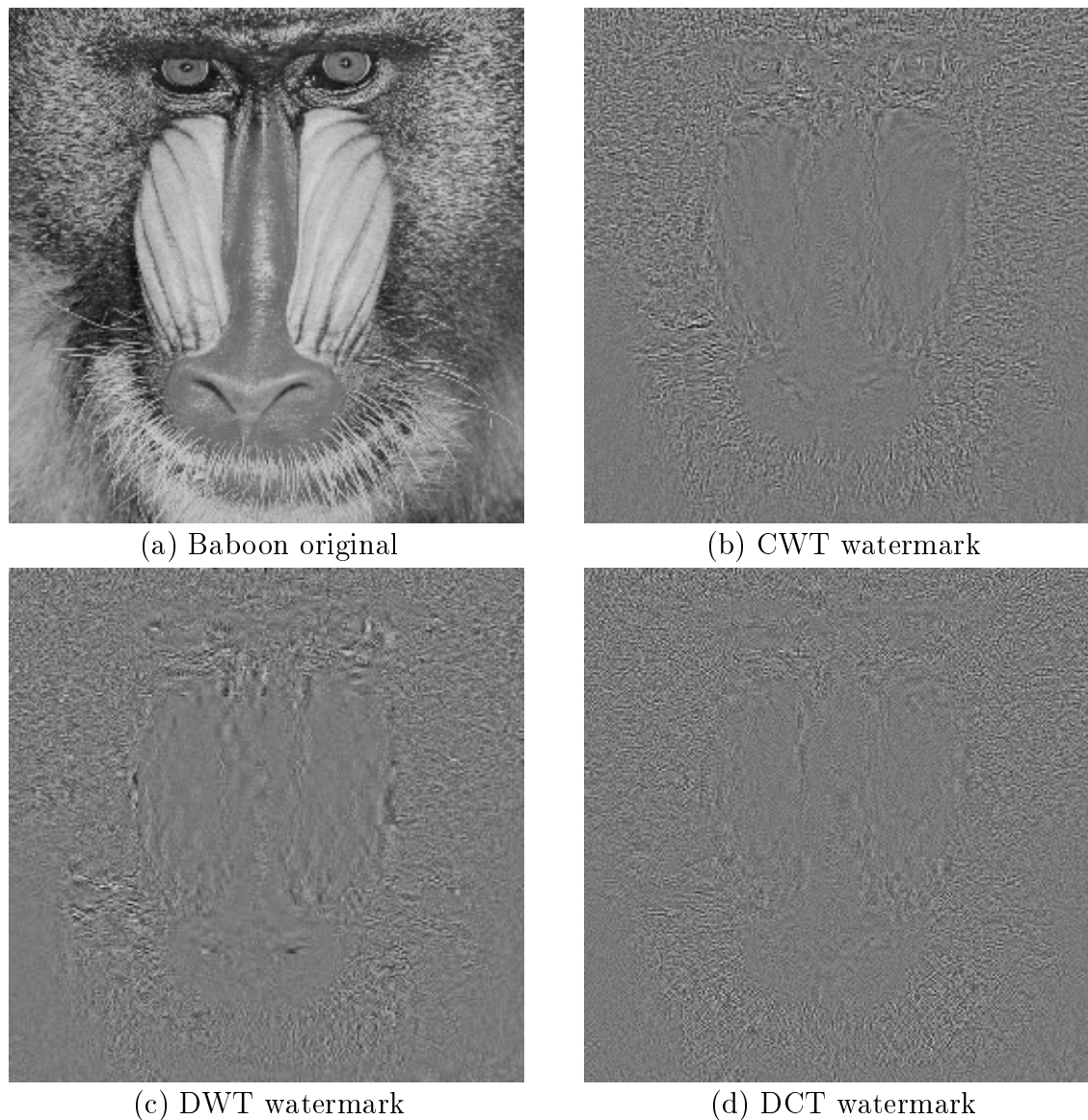


Figure 4.8: Baboon and the watermark (enhanced by a factor of 8) in the CWT, DWT, DCT domains

use RMS error as the means for comparison between watermarking schemes in most of the simulations in this thesis.

In the case of a yes/no watermark, the CWT also provides better robustness against JPEG compression than the scheme in [17]. The watermark is detectable in all three images even when the quality factor is 10. However, Barni's scheme will probably fail to detect the watermark at this quality for Lena and Pills.

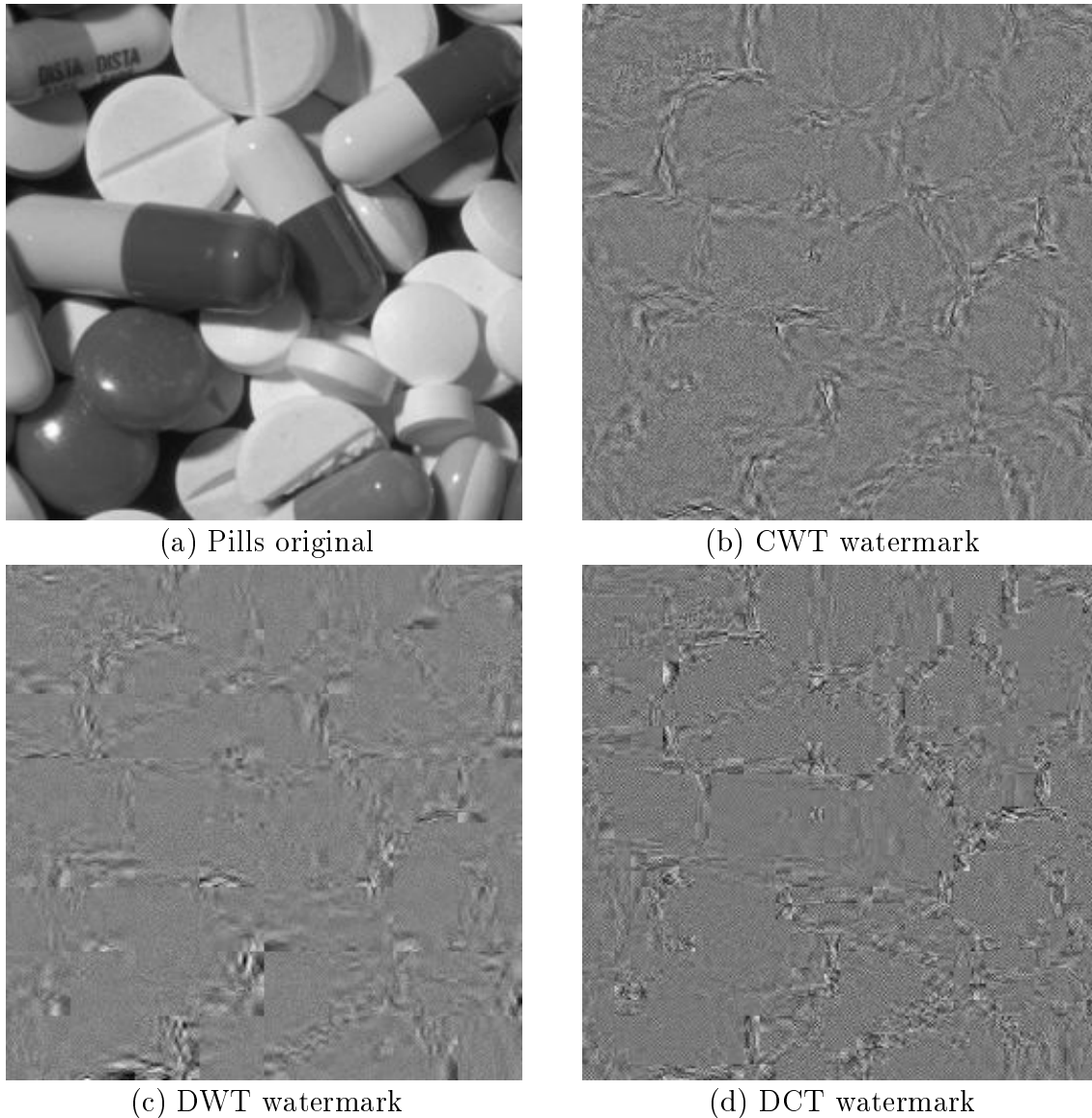


Figure 4.9: Pills (photo courtesy of Karel de Gendie) and the watermark (enhanced by a factor of 8) in the CWT, DWT, DCT domains

JPEG2000 compression JPEG2000 is the state of the art image compression technique which employs the DWT as the compression domain (instead of the DCT as in JPEG). The compression parameters (bit per pel (bpp)) are chosen such that the RMS error between the compressed unwatermarked image and the original image is roughly the same as in JPEG with quality factor 10 to 100. The JJ2000 package (version 4.1) from [71] was used in all simulations. Figures 4.10 (d-f) show the BER of the watermark in the three domains after JPEG2000 compression. The results are similar to that under JPEG compression. The CWT watermarks perform best and the

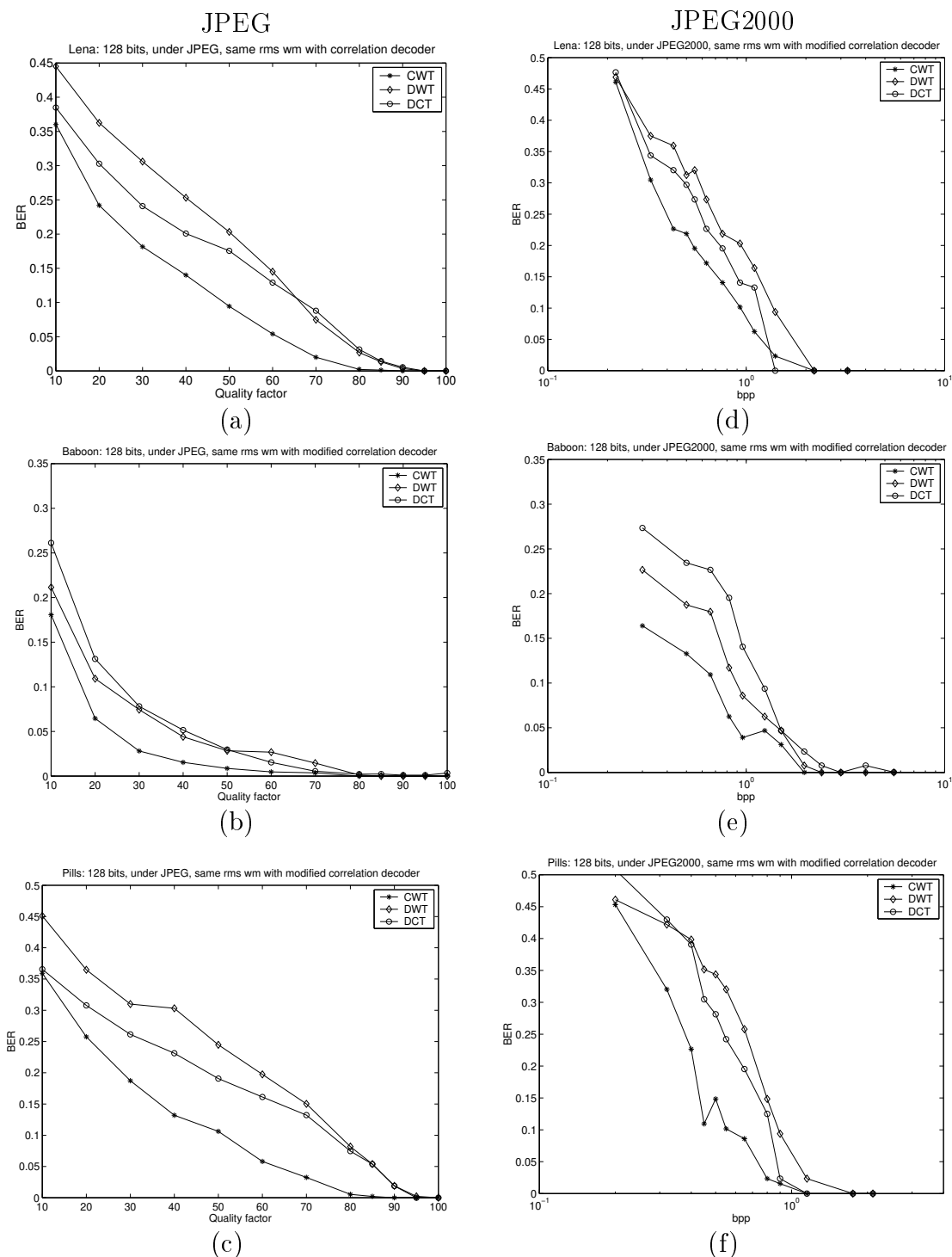


Figure 4.10: Comparison of blind correlation based watermark decoding in the CWT, DWT and the DCT domains under compression. Results under JPEG (a-c) and JPEG2000 (d-f) for 3 images: Lena (a,d), Baboon (b,e) and Pills (c,f) are shown. For each image, the RMS watermark in the three domains is adjusted to be the same. The CWT watermarks outperform the others in all cases whereas the DWT and the DCT watermarks have similar performance.

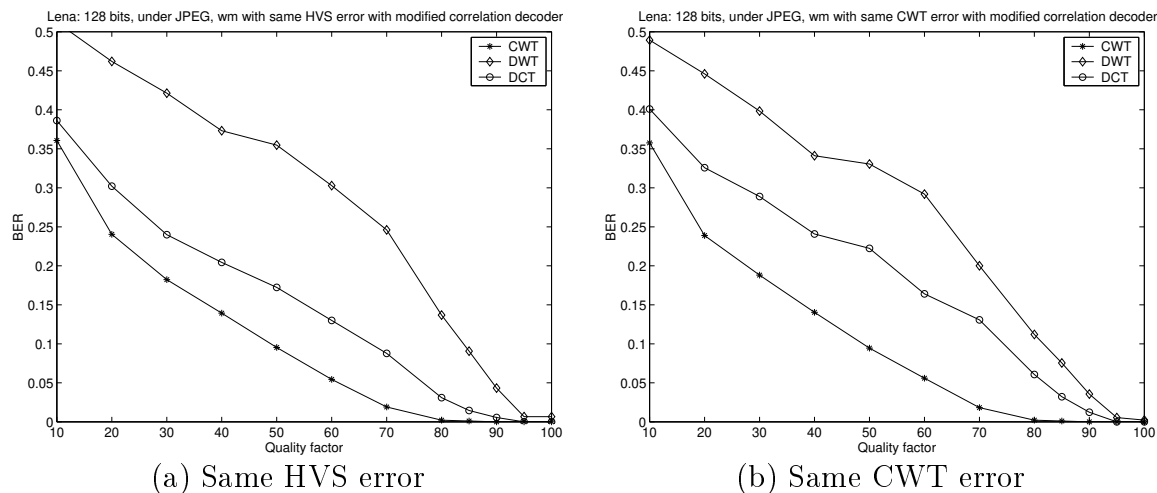


Figure 4.11: When the perceptual error (using the model in [76] (a) or our CWT model (b)) instead of the RMS watermark is constrained to be the same, we obtain the above results for Lena under JPEG. The DWT domain has much worse performance than in the previous figure (4.10). This is because the DWT watermarks produce higher perceptual distortion than both CWT and DCT watermarks. The CWT watermarks thus have better performance (relatively) than the simulation results shown here in practice, when the subjective quality of the watermarked image, rather than the RMS watermark, is constrained.

DWT and the DCT again have very similar performance. Figures 4.12 (a-c) compare the effects of JPEG and JPEG2000. At a given level of RMS error, JPEG2000 is more effective in removing watermark energy than JPEG. The results for the yes/no watermark case under JPEG2000 (figures 4.13 (d-f)) are similar to those under JPEG. The CWT watermark is significantly better than the DWT scheme in [17].

AWGN, mean and median filtering Figures 4.14 (a-f) show the results of the Lena image under AWGN, mean and median filtering attack respectively. In the case of AWGN, the level of RMS noise is from 4 to 28, whereas the window size of both mean and median filtering ranges from 1×1 (no attack) to 9×9 (severe attack). The CWT watermarks outperform watermarks in other domains in all cases. In the case of a yes/no watermark, only the CWT watermark remains detectable under mean and median filtering and Barni DWT scheme fails even for the lowest level of filtering (3×3 window). However, the error introduced to the watermarked images by any of these three attacks is very perceptible and in general the images cease to be useable before the embedded watermark becomes sufficiently degraded. These attacks are therefore relatively ineffective and will not be considered further for the remainder of

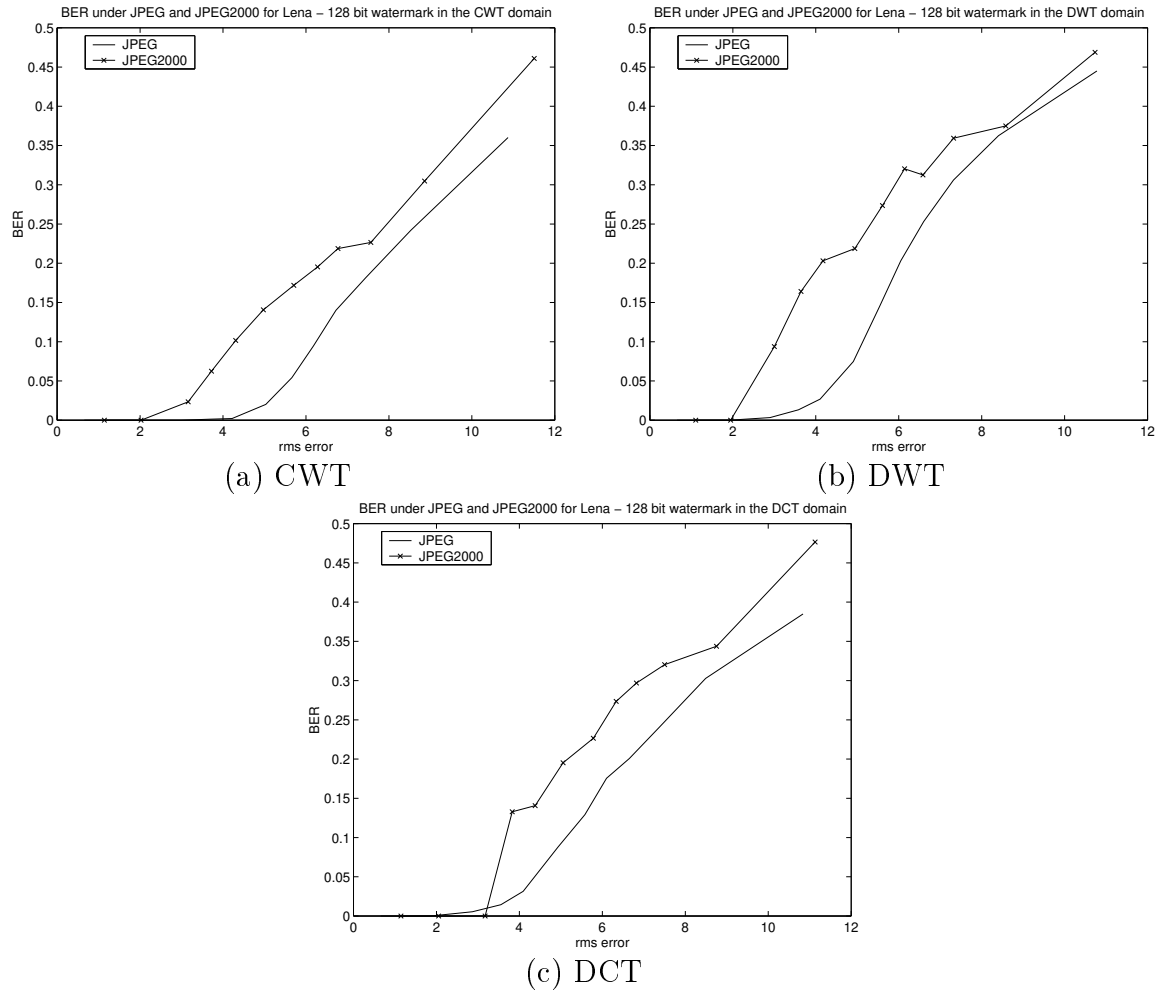


Figure 4.12: Figures (a-c) compare the effects of JPEG and JPEG2000 on correlation based watermark decoding. For a given RMS error, JPEG2000 is more effective in removing watermark energy as expected. Thus future watermarking schemes must be designed to be robust against JPEG2000.

the thesis.

4.5.2 Discussion

No error control codes (ECC) are employed in the results presented in the previous section, as the purpose of the simulations is to show that the CWT watermarks are more adapted to the host image and are consequently more robust. In practice, one would use a good ECC to encode the payload prior to watermark embedding and the decoder output BER will be much lower than shown. A typical ECC decoder can correct almost all the errors in the extracted bitstream as long as the input BER is lower than a certain threshold, which depends on actual ECC being used. For example, the threshold corresponding to an output BER $< 10^{-4}$ for the rate 1/3

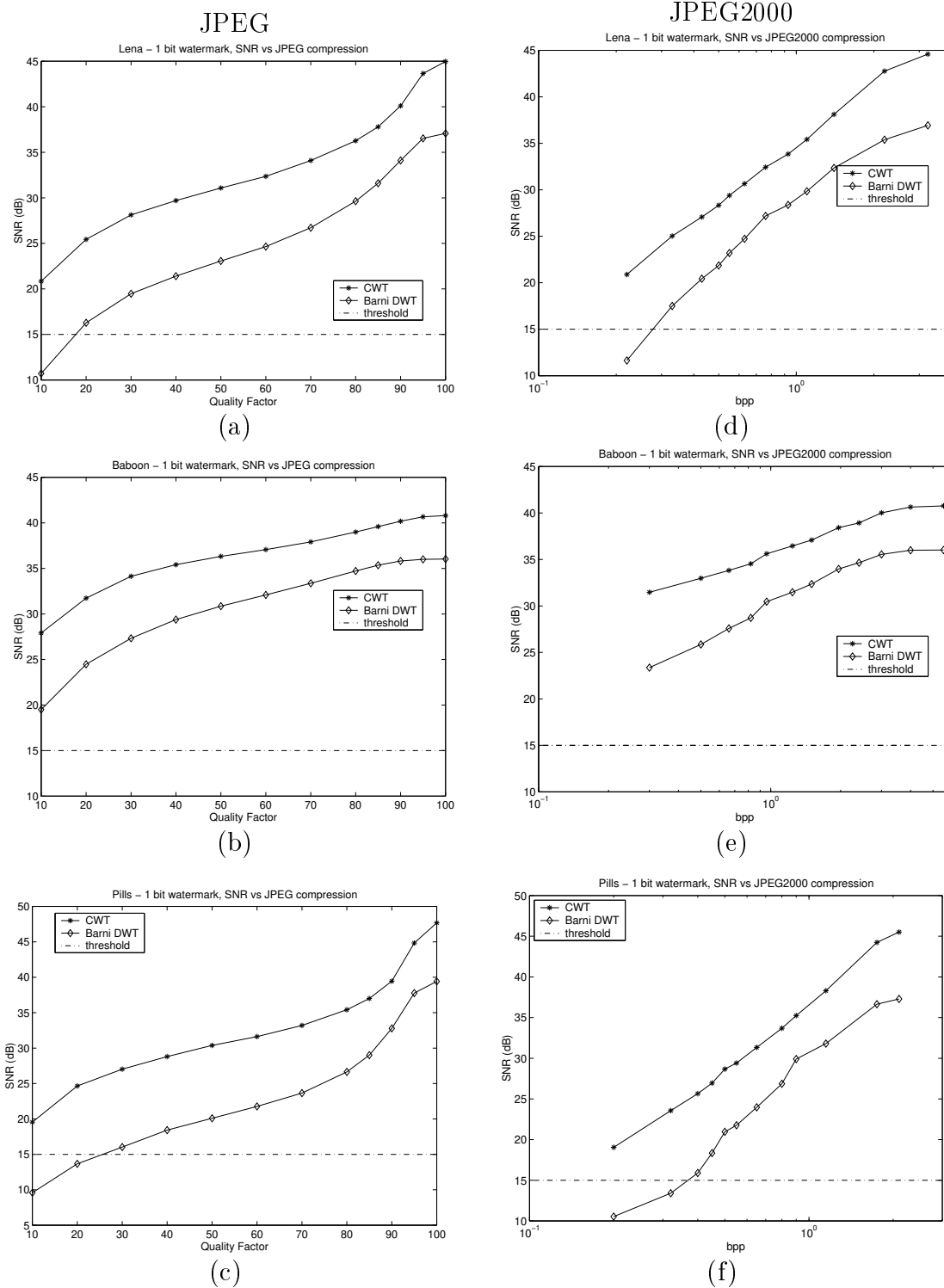
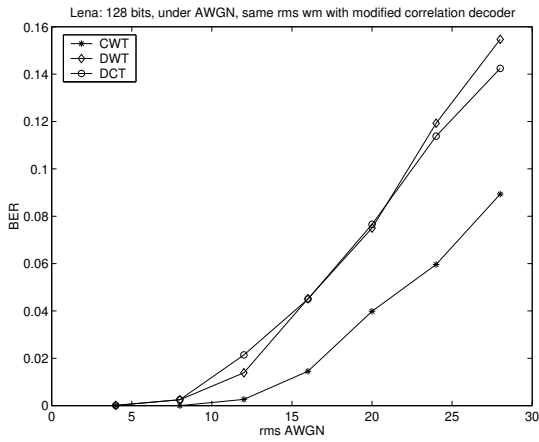
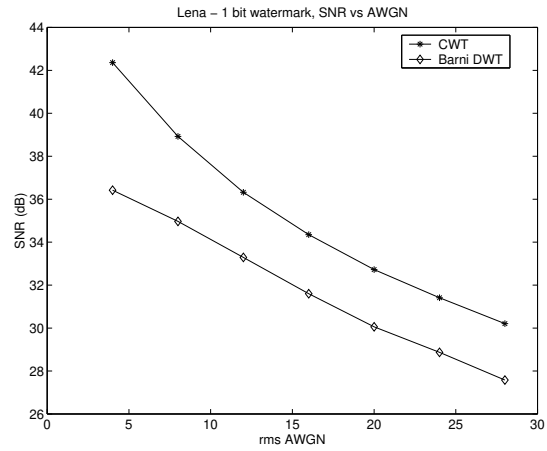


Figure 4.13: Comparing the 1 bit version of our CWT watermark to the DWT scheme in [17] under compression. The results for JPEG (a-c) and JPEG2000 (d-f) are shown using the same test images as in figure 4.10. The CWT watermark again outperforms the DWT watermark by a significant margin.

AWGN

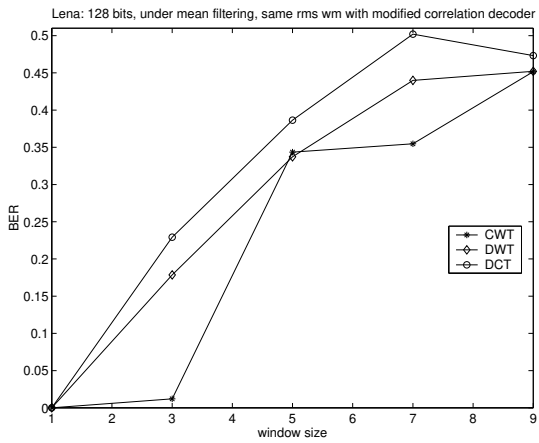


(a) multi-bit watermark

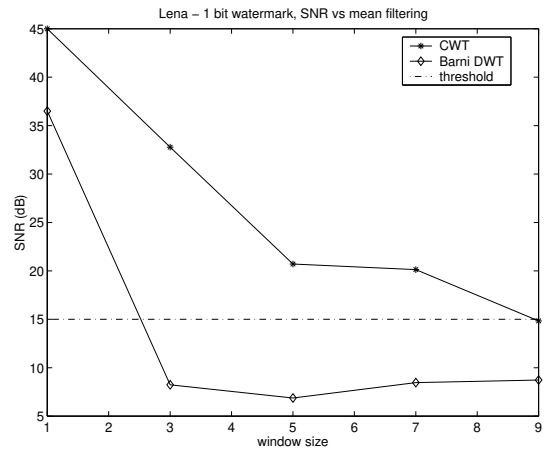


(b) yes/no watermark

Mean filtering

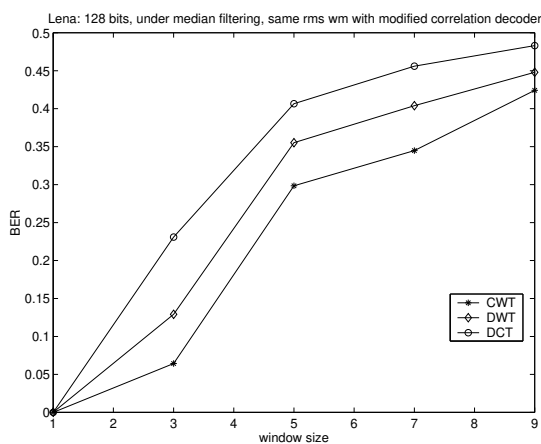


(c) multi-bit watermark

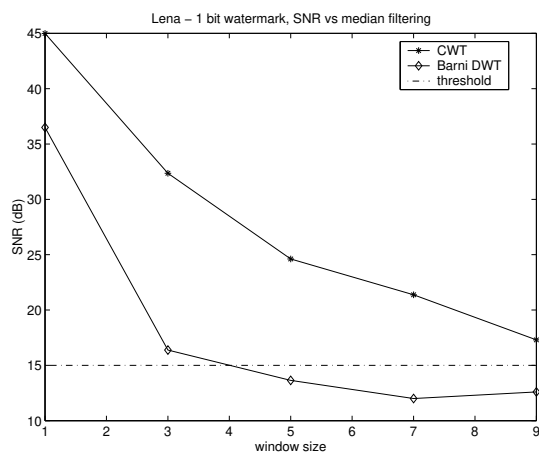


(d) yes/no watermark

Median filtering



(e) multi-bit watermark



(f) yes/no watermark

Figure 4.14: Results of correlation based watermark decoding under AWGN, mean and median filtering attacks for both the multi-bit watermark (left column) and the 1 bit version (right column). Only the results for the Lena image are shown. The CWT watermark outperforms the rest in all cases.

turbo code (discussed in section 2.4.2) is about 0.15. When the decoder BER exceeds the threshold, the BER of the payload output by the ECC decoder approaches 0.5 and the watermark system fails catastrophically. Therefore a practical way of comparing watermark decoder performance is to compare the level of attack at which the BER reaches this critical value. If we assume the highest acceptable level of compression is quality factor 70 for JPEG and 0.9 bpp for JPEG2000 for images with no significant high frequency components, we can see that the CWT watermarks perform very well under compression, where the critical attack levels are well over the acceptable level of compression for both JPEG and JPEG2000. On the other hand, watermarks in the DWT and the DCT domains can only tolerate much lower level of compression. The AWGN, mean and median filtering attacks are not effective because these attacks do not consider the masking effect of the host image and critical attack levels for these attacks will produce unacceptable distortion in the watermarked images. In the case of a yes/no watermark, the scheme proposed by Barni [17] degrades more rapidly than our CWT scheme, because the former algorithm only uses the finest resolution wavelet coefficients, which become very unreliable even under low level of compression. This highlights the importance in using *both* high and middle frequency components in watermarking.

4.6 On watermark detection

In the previous sections we analysed and discussed how to decode a watermark given that a watermark is present in an image. However, in any watermark application, one must be able to distinguish unwatermarked documents from the watermarked ones. In order to measure watermark detection false alarm rate as low as, say, 10^{-8} experimentally, one would ideally need to test the watermark detector over a large database of unwatermarked images and images watermarked with the incorrect key, which is impractical. Therefore, we will only provide theoretical results here. We also assume the watermark is white. If the watermark is not white, then the watermark spectrum will have an influence on the detector performance [102]. The problem of watermark detection can also be formulated into a hypothesis test (with the null hypothesis being no watermark is present):

$$\begin{aligned} H_0 : \mathbf{y} &= \mathbf{e}, \\ H_1 : \mathbf{y} &= \eta \mathbf{s} + \mathbf{e} \quad 0 < \eta \leq 1, \end{aligned} \tag{4.30}$$

where \mathbf{e} is the noise vector which includes the host image coefficients. η takes into account the possibility that the watermarked image is attacked and the amplitude of the watermark \mathbf{s} is reduced as a result. Lu *et al.* [104] and Martinez *et al.* [109] propose the use of *locally optimal detector* (LOD) which maximises the slope of the power function when η is close to zero. If the noise statistics are known, the corresponding LOD can be derived as:

$$-\sum_i s_i \cdot \frac{f'_e(y_i)}{f_e(y_i)} \underset{H_0}{\overset{H_1}{\gtrless}} \lambda, \quad (4.31)$$

where $f_e(\cdot)$ and $f'_e(\cdot)$ are the probability density function (pdf) of the noise statistics and its derivative. The threshold λ can be chosen according to the Neyman-Pearson principle [130] by fixing the false positive probability. Determining the optimal threshold is not always easy as it depends on the attacks the image has suffered. In case of watermark *decoding*, no threshold is necessary as decoder statistics are often symmetrical about zero, so we can conveniently use zero as the threshold in distinguishing between bit 1 and bit 0. Unfortunately, watermark decoders cannot in general be used for watermark detection as they will always return *something* even when applied on an unwatermarked image (see appendix B.2).

In order to make a watermark *detectable*, a portion of the watermark must be *known in advance*. This is satisfied in yes/no watermarks where the detector knows exactly what the embedded sequence is, but these watermarks carry no extra payload and are thus not very useful. If we have a multi-bit payload, watermark detection can be achieved by one of the followings:

1. Fix L bits of the payload to be of known value. The watermark decoder decodes the watermark as usual and compares these L bits to the known pattern. If all L bits match, then the watermark is detected.
2. Allocate part of the payload energy to a separate watermark, which is embedded as a reference sequence. A watermark detector (e.g. LOD) is used to detect this reference watermark and the decoder proceeds only when the detector successfully detects the reference mark.

These two approaches are approximately equivalent and the second approach can be treated as decoding all L bits as a whole. The false positive P_{FP} , miss P_{Miss} and detection P_D probabilities are derived in appendix B.2, and are shown below. We assume *no* error control code (ECC) is used for the moment.

Case 1:

$$\begin{aligned}
 P_{FP} &= 2^{-L}, \\
 P_{Miss} &= 1 - \left(1 - Q\left(\sqrt{\frac{M}{L+M}} \cdot SNR\right) \right)^L, \\
 P_D &= \left(1 - Q\left(\sqrt{\frac{M}{L+M}} \cdot SNR\right) \right)^L.
 \end{aligned} \tag{4.32}$$

Case 2:

$$\begin{aligned}
 P_{FP} &= p \quad \text{which is chosen by user,} \\
 P_{Miss} &\approx Q\left(\sqrt{\frac{LM}{L+M}} \cdot SNR - Q^{-1}(p)\right), \\
 P_D &\approx 1 - Q\left(\sqrt{\frac{LM}{L+M}} \cdot SNR - Q^{-1}(p)\right),
 \end{aligned} \tag{4.33}$$

where $Q^{-1}(\cdot)$ is the inverse of the Q function defined in (4.15); M is the length of the payload (*excluding* the fixed, known part) and SNR is the signal to noise ratio *per bit* at the decoder output if *no* reference watermark is embedded. In the case of a yes/no watermark ($M = 0$), the first approach splits the mark into L bits and uses a watermark decoder for detection. The false alarm rates for both approaches stay the same and the miss and detection probabilities become:

Case 1:

$$\begin{aligned}
 P_{Miss_{yes/no}} &= 1 - \left(1 - Q\left(\sqrt{\frac{SNR}{L}}\right) \right)^L, \\
 P_{D_{yes/no}} &= \left(1 - Q\left(\sqrt{\frac{SNR}{L}}\right) \right)^L.
 \end{aligned} \tag{4.34}$$

Case 2:

$$\begin{aligned}
 P_{Miss_{yes/no}} &\approx Q(\sqrt{SNR} - Q^{-1}(p)), \\
 P_{D_{yes/no}} &\approx 1 - Q(\sqrt{SNR} - Q^{-1}(p)).
 \end{aligned} \tag{4.35}$$

Figures 4.15 show the receiver operating characteristics (ROC) curve⁶ for a yes/no watermark at a SNR of 15dB (the detection threshold at $P_{FP} = 10^{-8}$) and for a 16-bit watermark at the same overall SNR, so the total energy of watermark are the same in the two cases. We can see that in both cases, using a reference watermark is much better as expected. Intuitively, using a L -bit fixed pattern seems unwise. However, using a full-frame reference watermark is not always possible, as we will see in chapter 6. In practice, we will use some form of error control coding on the payload and we can actually use the code to detect the watermark. This is explained in detail in chapter 6 (section 6.6).

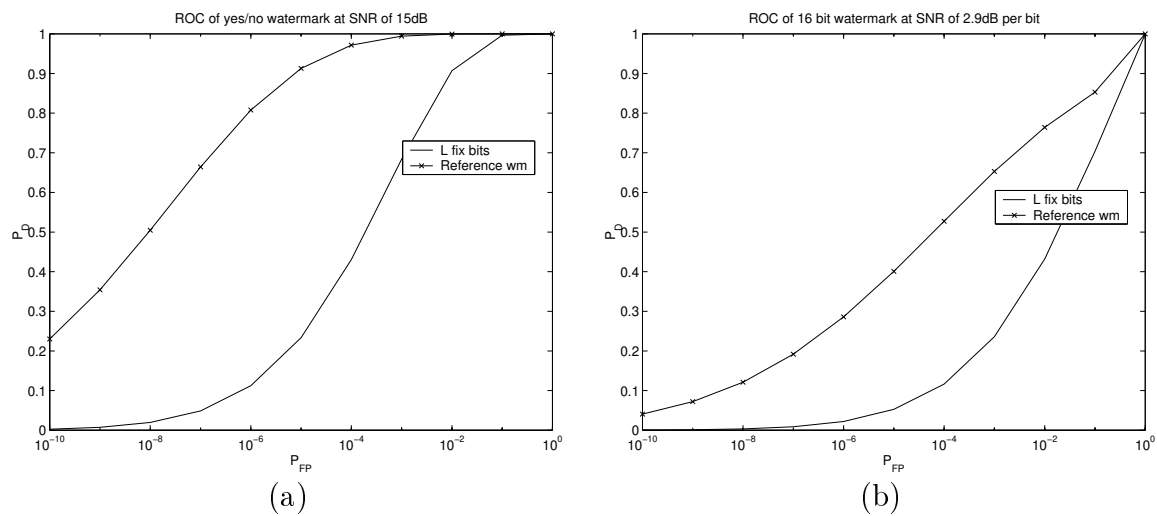


Figure 4.15: Receiver Operating Characteristics (ROC) of the two watermark detectors in section 4.6 in the case of (a) 1-bit watermark at SNR 15dB. (b) 16-bit watermark at the same overall SNR. We can see that in both cases detecting the reference watermark as a whole is better.

4.7 Chapter summary

In this chapter, we discussed the basic watermarking model and the use of spread spectrum in watermarking. We proposed a generic blind spread spectrum based watermarking scheme, which gets round the redundancy of the CWT. We assumed the interference due to the host image follows a non-stationary Gaussian distribution, and derived an optimal correlation based decoder under this scenario. Experimental results confirmed that the modified matched filter works better than a conventional

⁶The receiver operating characteristics of a detector are defined as the relationship between the false positive (P_{FP}) and detection (P_D) probabilities as the detection threshold is varied.

matched filter under non-stationary noise. We also discussed how to combine information from multiple channels to improve the performance of the watermark decoder. Watermarking in the CWT, DWT, DCT domains were compared under different compression conditions as well as under AWGN, mean and median filtering. In all cases, the CWT watermark performed best. The proposed CWT watermarking scheme also outperforms the proposed yes/no watermark scheme in [17], which shows one should always use both high and middle frequency components for watermarking. Finally, the issue of watermark *detection* is discussed and we concluded that, in the absence of error control codes, one should always use a separate sequence as the reference watermark. In the next chapter, we will look at watermark decoding after attacks in more detail.

Chapter 5

Reliable Decoding of Watermarks after Attacks

5.1 Introduction

In the previous chapter we discussed blind spread spectrum based watermark embedding as well as decoding based on correlation. We concluded that the modified matched filter works best because of the non-stationary nature of the host coefficients. In our theoretical analysis, the watermarked image is assumed to be not attacked. In practice, this is unlikely as potential pirates will try to remove the watermark without degrading the commercial value of the image. In addition, honest users may also process the image to suit their needs. A robust watermarking system must be able to detect and decode a watermark from a modified watermarked image, as long as the image is still considered to be *useful*. As discussed in chapter 2, there exist numerous attacks on watermarking systems [95]. Some are intentional (e.g. geometric distortion, denoising), while the others may be unintentional (e.g. cropping, rotation, compression). In this chapter, we concentrate on the following three attacks: compression, geometric distortion and denoising. These attacks are chosen because compression is a very common operation performed on images, whereas geometric distortion and denoising are shown to be effective attacks on watermarks. We first discuss the effects of compression on a watermark and derive an alternative decoder for decoding after compression. Then we describe a novel image registration algorithm based on motion estimation to combat geometric distortion attacks. Finally, we address robustness of watermarks to denoising attacks.

5.2 Watermark decoding after compression

5.2.1 Effects of compression on watermarks

Compression is one of the most common operations on images and hence a robust watermarking system must survive under a reasonable level of compression. Figure 5.1 shows the scenario of a watermarked image being compressed. \mathbf{T}_F and \mathbf{T}_C are the transform domains used for watermarking and compression respectively, and \mathbf{T}_F^{-1} , \mathbf{T}_C^{-1} are their inverses. Q_Δ is a uniform quantiser with step size Δ . Suppose the quantiser introduces a quantisation error of \mathbf{e}_Q , then the error at the input of the decoder will be

$$\mathbf{e}'_Q = \mathbf{T}_F \mathbf{T}_C^{-1} \mathbf{e}_Q. \quad (5.1)$$

We ignore the visual masking of the watermark (and the inverse scaling at the decoder) for the moment to simplify our discussion, (i.e. the watermarked signal is given by $\mathbf{x}' = \mathbf{x} + \mathbf{w}$), and the expectation of the correlator is given by:

$$\begin{aligned} E(r) &= E\left(\frac{1}{N} \sum \mathbf{y} \cdot \mathbf{w}\right), \\ &= E\left(\frac{1}{N} \sum (\mathbf{x}' + \mathbf{e}'_Q) \cdot \mathbf{w}\right), \\ &= \sigma_w^2 + \frac{1}{N} E\left(\sum \mathbf{e}'_Q \cdot \mathbf{w}\right). \end{aligned} \quad (5.2)$$

Ideally we would like the quantisation error to be *independent* of the watermark (the second term of (5.2) is zero), in which case the only effect on the mean of the correlation will be due to signal attenuation by compression and the variance will be the sum of the variance due to the compressed host coefficients and the variance of the quantisation noise. We first discuss the following two cases:

- $\mathbf{T}_F = \mathbf{T}_C$ - the compression domain is the *same* as the watermarking domain,
- $\mathbf{T}_F \neq \mathbf{T}_C$ - the two domains are different,

and then we will derive an alternative decoder.

5.2.1.1 $\mathbf{T}_F = \mathbf{T}_C$

When the compression and the watermark domains are the same, $\mathbf{e}'_Q = \mathbf{e}_Q$. Wolfgang *et al.* [160] argue that one should use the same domain for watermarking and com-

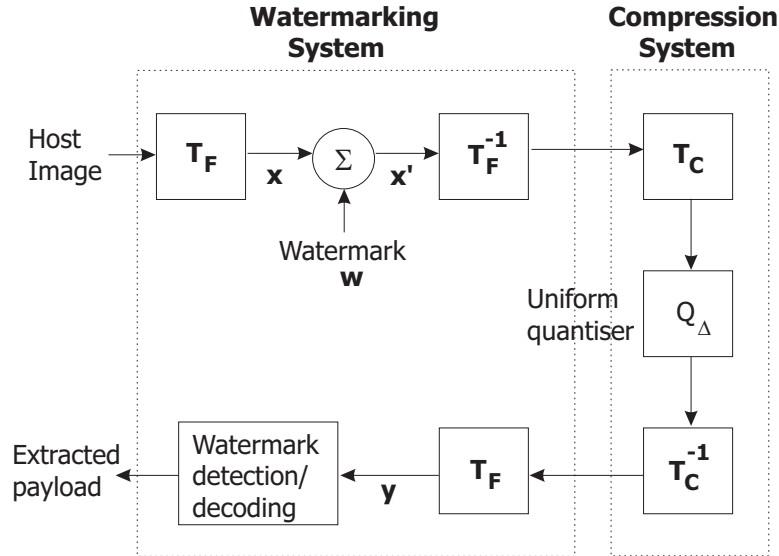


Figure 5.1: A model of compression of watermarked images. The watermark domain T_F and the compression domain T_C can be identical or different. In general, the watermark is more robust when T_F and T_C are different. This is because when the two domains are the same, the quantisation error tends to be correlated with the watermark and degrades the performance of the decoder.

pression, but their simulations do not fully agree with their arguments. Eggers [51] analyses compression of watermarked images as a *non-subtractive dithered quantiser*, where the watermark is the dither, and derived expressions for the mean and the variance of a correlator. He concluded that, $E(\mathbf{w}^T \mathbf{e}'_Q) = 0$ is satisfied only under fine quantisation, where the distribution of the host image (after quantisation) can be expressed as a convolution between the original distribution of the host and a uniform distribution in the range $[-\frac{\Delta}{2}, \frac{\Delta}{2})$. Under medium and coarse quantisation, the quantisation error will be *negatively* correlated with the watermark and hence the SNR at the correlator will be lower than the case where \mathbf{e}_Q were some independent noise. The solution suggested by Eggers for optimal detection of watermark after quantisation [50] is to calculate the weights ν (defined as μ/σ^2) for combining outputs of correlator from different channels, which is the same as what we proposed for multichannel decoding in section 4.4.4. μ, σ^2 can be precalculated from the equations given in [51] when the level of compression is known in advance. If the level of compression is not known, which is typically the case in practice, Eggers suggested that a pessimistic worst case compression level can be assumed for calculating channel weights, in which case the detector performance will still be good if the actual level of compression is less severe than the worst case. Fixing the weights given to each channel in advance also removes the need for using a reference watermark for calcu-

lating μ, σ^2 at the decoder, which in turn allows us to put the energy of the reference watermark into the payload to increase the SNR at the decoder. The significance of this will be explained further in the following chapter.

5.2.1.2 $\mathbf{T}_F \neq \mathbf{T}_C$

When the two domains are different, the matrix $\mathbf{T}_F \mathbf{T}_C^{-1}$ has the effect of *decorrelating* the quantisation error. In the extreme case where the error is completely decorrelated, i.e. $E(\mathbf{w}^T \mathbf{T}_F \mathbf{T}_C^{-1} \mathbf{e}_Q) = 0$, the quantisation error just appears as an additional noise source at the decoder. In practice, the quantisation noise will not be completely decorrelated, but it is not straightforward to analyse the performance of the correlator given arbitrary \mathbf{T}_F and \mathbf{T}_C , because quantisation is a non-linear process. Fei *et al.* [55] linearise the uniform quantiser in figure 5.1 and showed that the resulting linear model has higher capacity when different domains for watermarking and compression are used. However, the authors in [55] assume the image coefficients follow a stationary Gaussian distribution after compression, which is not generally true. Ramkumar *et al.* [133, 134] also suggest that the use of transforms with poor energy compaction properties, which are unsuitable for compression, provides greater watermark capacity. Xia *et al.* [164] derive explicit expressions of the quantisation noise distribution under the same scenario in terms of the characteristic function of the distribution of the image coefficients. In general, image coefficients do not necessarily follow Gaussian distribution, especially after compression. Nevertheless we can expect the following when a watermarked image undergoes compression:

1. Any compression system will remove *perceptually insignificant* components of an image. Therefore the watermark energy in the high frequency components will be reduced, or even be removed altogether, whereas the watermark energy in middle or lower frequency components will be more or less preserved.
2. The more *different* the compression and the watermarking domains are, the less correlated will the quantisation error be with the watermark, and performance of the decoder/detector will be likely to improve.

The optimal solution for decoding/detection of watermark when using correlation is the same as the previous case, i.e. we calculate the weights $\nu = \mu/\sigma^2$ for each channel and combine the outputs from multiple channels accordingly. We can also fix the weights in advance as in the previous case.

5.2.2 The generalised (modified) matched filter

The use of correlation for watermark decoding is based on the assumption that the noise at the decoder is Gaussian. Unfortunately, this does not hold after compression. As the compression level increases, more and more coefficients in the higher frequency bands are quantised to zero. The distribution of the watermark and the host coefficients in these subbands thus become more peaky. We can model the received signal after compression as:

$$\mathbf{y} = \eta \mathbf{w} + \mathbf{e}. \quad (5.3)$$

η accounts for the fact the watermark amplitude is decreased after compression, but since this has the same effect on the signal for both a bit 0 and a bit 1, we can ignore it in decoding. The channel noise \mathbf{e} can be described by a generalised Gaussian distribution $\mathcal{GG}(\mu, c, \sigma)$:

$$p_{\mathcal{GG}(\mu, c, \sigma)}(x) = \frac{c}{2\Gamma(1/c)} b(\sigma, c) \exp(-b(\sigma, c)^c |x - \mu|^c), \quad (5.4)$$

where,

$$b(\sigma, c) = \frac{1}{\sigma} \sqrt{\frac{\Gamma(3/c)}{\Gamma(1/c)}} \quad \text{and} \quad \Gamma(a) = \int_a^\infty u^{a-1} e^{-u} du. \quad (5.5)$$

μ is the mean of the distribution; σ is the standard deviation and c is the shape parameter. The Gaussian and the Laplacian distribution are special cases of (5.4) with shapes $c = 2$ and $c = 1$ respectively. As the shape parameter c decreases, the distribution becomes more peaky and has heavier tails, whereas the distribution tends to uniform as c tends to infinity. If we substitute the probability distribution (5.4) into the ML decoder and take the log-likelihood (equation (4.9)), we arrive at the following generalised (modified) matched filter (or generalised Gaussian decoder):

$$r_{gg} = \frac{1}{N} b(\sigma, c)^c \sum_i (|y_i + w_i|^c - |y_i - w_i|^c) \underset{\text{bit 1}}{\overset{\text{bit 0}}{\gtrless}} 0. \quad (5.6)$$

If the underlying noise follows the generalised Gaussian distribution with a shape parameter other than 2, then the generalised Gaussian decoder should perform better than the correlator. We first estimate the perceptual mask from the compressed image as before and inversely scale the received coefficients to make the input approximately stationary and then use (5.6) for decoding. When $c = 2$, (5.6) reduces

to the conventional correlator, normalised by $\frac{2}{\sigma^2}$. When $c < 2$, the non-linearity of the decoder has the effect of suppressing outliers in the received signal, similar to the locally optimal detector discussed in section 4.6. The use of generalised matched filter was discussed by various authors (for example, Hernández *et al.* [66] and Cheng *et al.* [33]) in the context of DCT domain watermarking, but in many cases the distribution of the DCT coefficients is assumed to be Laplacian (i.e. c is fixed to 1). In practice, we estimate the shape dynamically using moment matching as described in [20, 108].

When there is more than one channel available, we estimate the shape *independently* for each channel, substitute the shape into (5.6) and add the outputs from different channels together. No scaling factors ν for combining channel outputs are necessary as (5.6) is the log-likelihood ratio¹. In the case of watermark *detection*, it is not possible to use the local optimal detector in (4.31) directly because there are no closed form expressions of the probability density function and its derivative of the generalised Gaussian distribution with arbitrary shapes. However, we can, for example, model the generalised Gaussian distribution as a mixture of Gaussians [35, 22] and use the LOD on this mixture. Precautions should be taken when the underlying noise cannot be modelled by the generalised Gaussian distribution. In such scenario, the generalised Gaussian decoder may be suboptimal and using the modified correlation decoder as described in the previous chapter is a good compromise.

5.2.3 Simulation results and discussion

We repeat the test in section 4.5 for both JPEG and JPEG2000 compression and investigate any possible improvement using the generalised matched filter. A 128 bit watermark (uncoded) is embedded as described in the last chapter, and the generalised Gaussian decoder is used to decode the watermark after inverse scaling. Note that we do not need a reference watermark to estimate ν_i (the weights for combining multiple channel outputs) in the case of the generalised Gaussian decoder as discussed in the previous section, and the RMS watermark payload is adjusted to be the same as in the simulations in section 4.5 for fair comparison.

Figure 5.2 shows the results of the generalised Gaussian decoder in the CWT,

¹Equation 5.6 is actually the normalised log-likelihood ratio. If all channels have the same number of coefficients, the factor $1/N$ is not necessary. However, in wavelet transform, the finest resolution subbands have the most coefficients and are also less reliable after compression than the coarser resolutions. Normalising (5.6) prevents the more unreliable channels from dominating the overall log-likelihood.

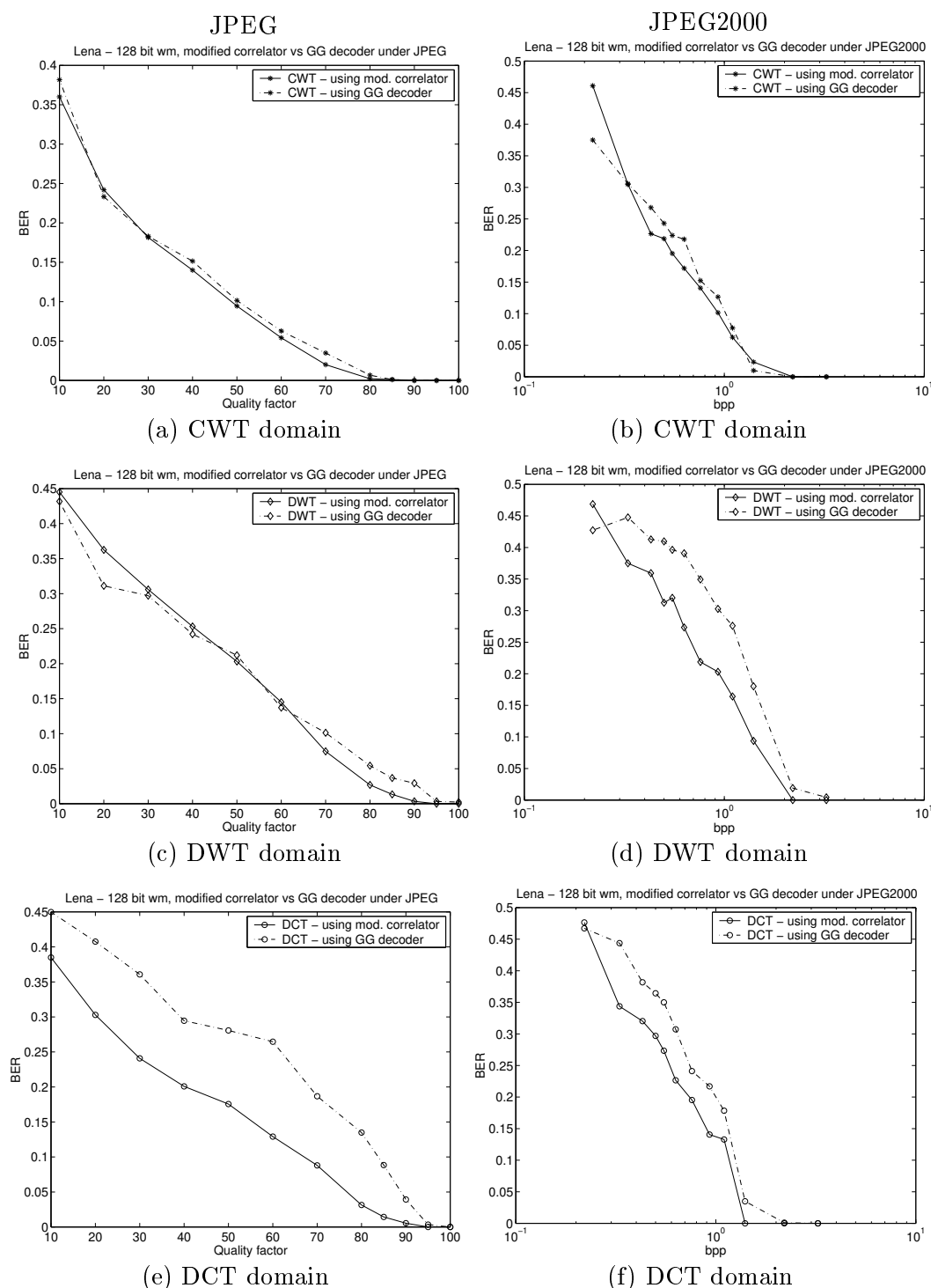


Figure 5.2: Comparing the modified correlator and the generalised Gaussian decoder under JPEG and JPEG2000 for the CWT (a,b), DWT (c,d) and DCT (e,f) domains. All the results are for the Lena image. Results for Baboon and Pills images are similar and are not shown here. The generalised Gaussian decoder has similar performance to the modified correlator in the CWT domain under both JPEG and JPEG2000. However, the generalised Gaussian decoder performs worse in the DWT and DCT domains under JPEG2000 and JPEG respectively.

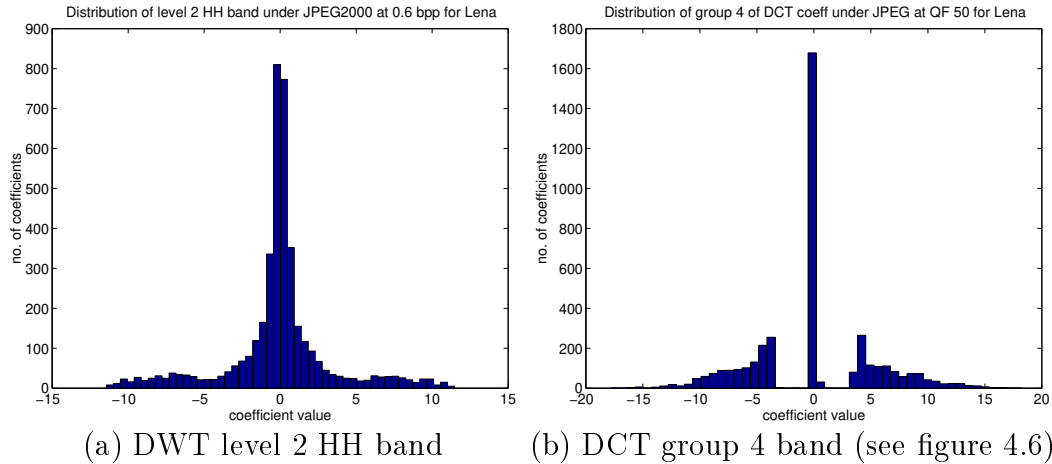


Figure 5.3: Example distributions of inversely scaled coefficients after compression for the DWT (a) and the DCT (b) middle frequency bands of the Lena image. The distribution on the left is after JPEG2000 at 0.6bpp and the one on the right is after JPEG at quality factor 50. The RMS error introduced by compression is about the same in both cases. Neither of the two distributions can be described well by a generalised Gaussian distribution. Using the modified correlator in these cases is a good compromise. The distribution of inversely scaled CWT coefficients does not exhibit this behaviour after compression.

DWT and DCT domains under both JPEG and JPEG2000. The generalised Gaussian decoder has similar performance compared to the modified correlator in the CWT domain, but performs worse in the DWT and DCT domains under JPEG2000 and JPEG respectively. The results are a bit surprising. The lack of improvement in the CWT domain is probably because inverse scaling has the effect of reducing the peakiness of the coefficient distribution. On the other hand, the distribution of inversely scaled coefficients in the DWT and DCT domains does not always follow a generalised Gaussian distribution, especially in the medium and low frequency bands. Figure 5.3 shows two examples of the distribution of inversely scaled DWT and DCT coefficients of one of the middle frequency bands of Lena under medium level of compression (quality factor of 50 for JPEG and 0.6 bpp for JPEG2000²). The distributions cannot be described by a generalised Gaussian distribution. In such cases, using correlation for decoding is a good compromise because the exact form of the distribution is not known. The distribution of the inversely scaled CWT coefficients does not exhibit such behaviour. In a nutshell, we conclude that using the modified form of correlator described in section 4.4.3 is good enough for decoding spread spectrum based watermarks in compressed images.

²This level of compression produces a noticeable degradation of the image

5.3 Watermark decoding after geometric distortion

5.3.1 The geometric distortion attack

The geometric distortion attack belongs to the group of geometric attacks (which include rotation and scaling, and they are discussed at the end of this section), where the attacker does not remove the watermark, but aims to destroy the *synchronisation* of the watermark sequence. This attack was first proposed by Petitcolas *et al.* in [123], and resulted in the first benchmark (*Stirmark*) [95] on watermarking systems. Most existing watermarking systems employ the principle of spread spectrum because it allows reliable communication through a noisy channel. Unfortunately, spread spectrum systems are prone to timing errors, even a slight shift in the synchronisation sequence can confuse the detector. The geometric distortion attack exploits this weakness by shifting each pixel by a small distance in a random but smooth manner such that the resulting image still resembles the original. An example of the geometric distortion attack is illustrated in figure 5.4. The top row shows the original Lena and the grid while the bottom row shows the distorted version. The grid illustrates the effect of the distortion on the pixel positions. If the distorted image is not resynchronised, a spread spectrum based watermark decoder will be defeated.

5.3.2 Image registration based on motion estimation

5.3.2.1 Introduction

The process of registration aims to resynchronise the watermark random sequence in a distorted watermarked image. Existing approaches fall mainly into two categories: template insertion [120] and feature matching [72]. The former involves embedding a template, typically a known pattern of peaks in the DFT domain, in addition to the watermark. The template extracted from the watermarked image is used to invert any possible distortions, i.e. this method is blind. Unfortunately this method is susceptible to the template removal attack [68]. The second approach uses the locations of feature points in the distorted image as well as those in a *reference* copy to infer the distortion. The underlying algorithm of these two approaches are identical. The distortion suffered by the watermarked image is modelled as a *global*

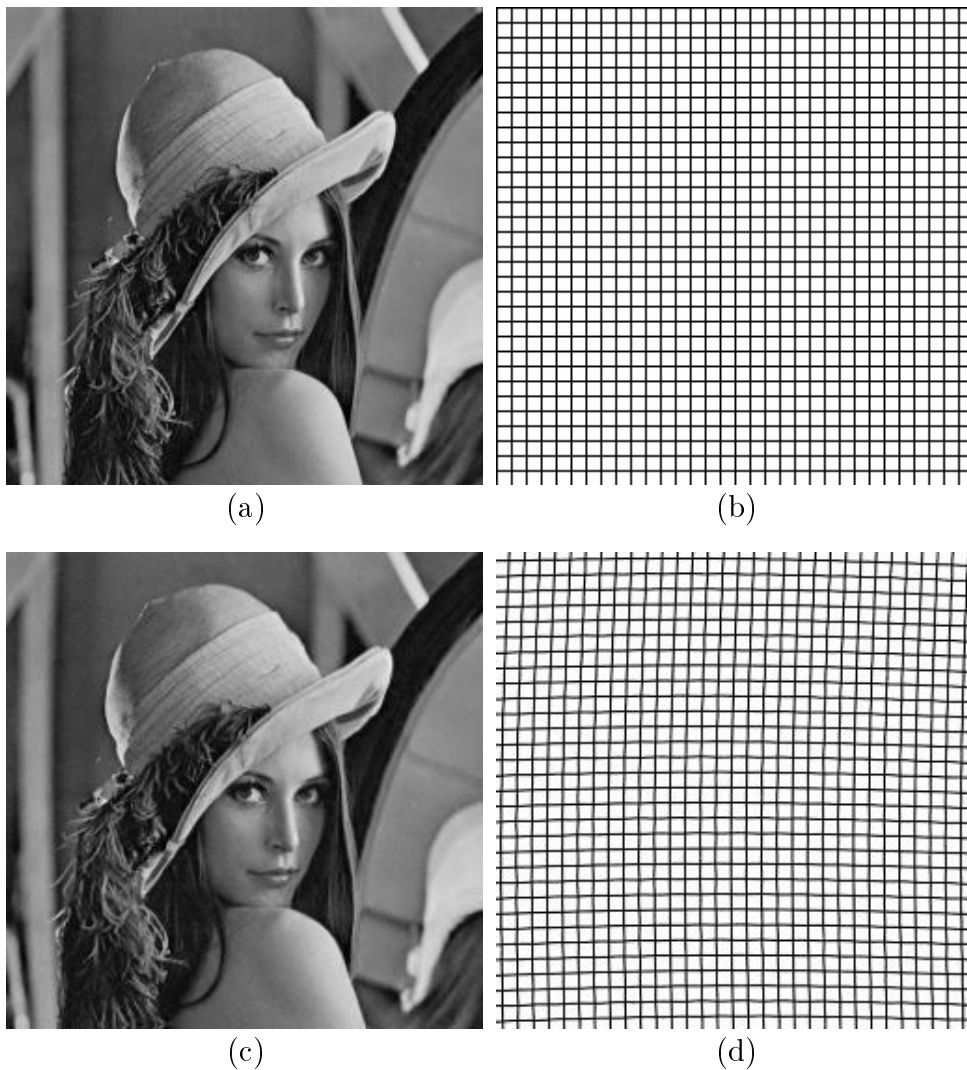


Figure 5.4: An example of the geometric distortion attack. The top row shows the undistorted version of Lena (a) and a grid (b), while the bottom row shows the effect after the geometric distortion attack (c) and (d). The grid illustrates the effect on the pixel locations after the attack. We can also see that the distorted version of Lena looks very similar to the original. In general, the distorted image resembles the original unless the image contains many vertical or horizontal lines (like the grid image), in which case the distortion will be perceptible.

affine transform:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} b_{11} \\ b_{21} \end{bmatrix}, \quad (5.7)$$

where (x', y') and (x, y) are the new and the original pixel location. The matrix parameters in (5.7) are solved via linear least square using the matched points in the

distorted image and reference image (or matched peaks of the template in the first case). This model assumes the underlying transformation is *constant* throughout the image, which, unfortunately, does not necessarily hold for deliberately localised transforms like geometric distortion attacks. Such distortion, however, can be reasonably well approximated by an affine transform on a small scale, if the image is to be a close approximation of the original.

An improved version of second registration scheme has been used in the past in registering satellite images [61]. Instead of modelling the distortion as an affine transform, a mapping function from the original image to the distorted image is estimated. The matched points are used as *anchors* and the mapping function is interpolated from these points using radial basis functions (see section 5.3.2.3).

In the following sections we will describe our novel image registration scheme, which is based on a similar idea as in [61]. However, instead of using matched feature points, we use motion vectors, thus eliminating the problem of identifying matching correspondences. Geometric distortion is viewed as a form of *motion* between the reference and the distorted image. Although the requirement of a reference image seems to conflict with blind watermark detection, it is not necessary to use the original host image as the reference. Ideally we would use an undistorted image, which has the *same* watermark as the distorted image, as the reference, so that the watermark can help with motion estimation in otherwise smooth regions. In practice, however, we can use an undistorted (or even compressed) image watermarked with a different algorithm as reference, with little degradation in performance. This is because the motion estimation algorithm is robust enough to cope with the presence of different watermarks embedded in the distorted and the reference images. Typically, a motion field is much denser than the distribution of feature points, and this allows us to estimate the mapping function more accurately. The overall image registration process consists of two steps: motion estimation and motion compensation. We will first summarise the basic motion estimation algorithm. Then we will describe how to detect and correct erroneous motion vectors and show some experimental results.

5.3.2.2 Motion estimation with complex wavelets

Motion estimation is performed in the CWT domain and the algorithm is derived from the one due to Magarey [107]. Magarey showed that complex wavelets allow more accurate motion estimation compared with traditional techniques like block matching [85] or gradient descent [86]. In addition, using complex wavelets allows seamless integration of image registration with our watermarking algorithm in the

CWT domain. Figure 5.5 depicts the structure of the algorithm. The inputs to each level are the detail CWT subimages of the reference and distorted images (plus the CSD surfaces of the previous level, which are defined later), while the output is a set of real value surfaces $CSD^l(\mathbf{n}, \mathbf{f})$, indexed by the subpel location $\mathbf{n} = (x, y)$, each of which defines a motion vector and its associated confidence at a given CWT level l .

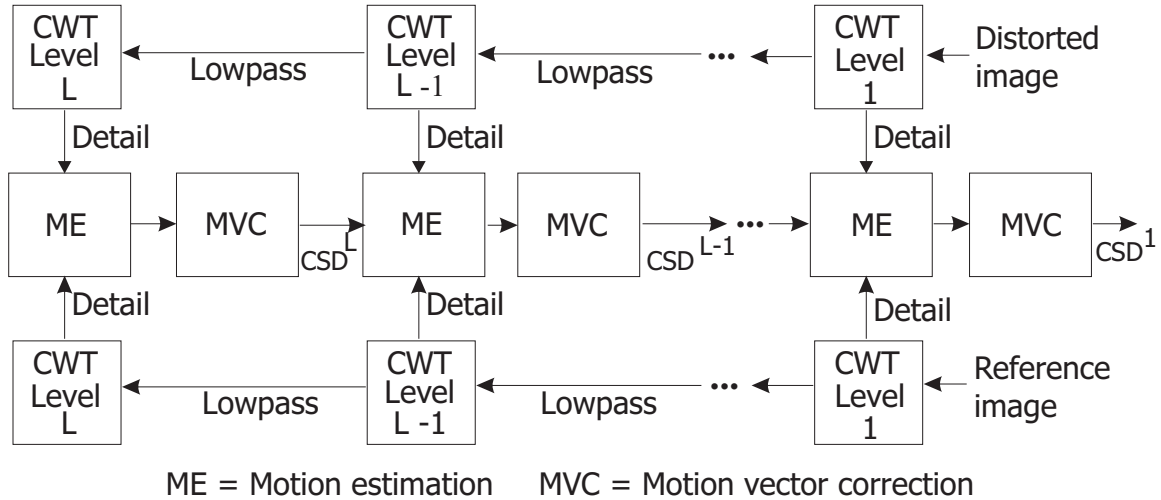


Figure 5.5: Block diagram of the motion estimation algorithm in the CWT domain. The algorithm is hierarchical and starts from the coarsest level of the CWT. The density of the motion field increases by a factor of 4 as we go up each resolution. The algorithm terminates when the desired motion estimate density is obtained.

The shift-dependent properties of the CWT are the essence of motion estimation. For relatively small displacements, the phase of the CWT coefficient at any given subband varies almost linearly with shift, while the amplitude is approximately shift invariant³. The CWT also has better directional selectivity than the conventional real wavelet transform as there are six subbands at each level instead of just three with real wavelets (figure 3.4). This allows us to distinguish between motion along opposing diagonals.

In order to estimate motion to subpixel accuracy, we need to estimate CWT coefficients at non-integer-indexed locations from the known integer-indexed coefficients in the same subband. This is achieved by modulating an interpolation kernel to the

³We could have used this property and use the CWT coefficients magnitude for embedding watermarks, and in this way the watermark would automatically be shift-invariant and be robust against geometric distortion without the need of registration. This is the reason why some authors (for example, de Rosa *et al.* [47] and Solachidis [137]) use the DFT domain for watermarking. Unfortunately we cannot use the magnitude of the host image CWT coefficients (and without changing their phase) in watermarking, as the resulting watermark will violate equation (4.2) and there will be information loss upon inverse transform (see section 4.3.1).

centre frequency of the filter of the subband concerned, and convolving the resulting kernel with the CWT coefficients. If we denote the CWT coefficients at level l , subband θ ($\theta \in \{\pm 75^\circ \pm 15^\circ \pm 45^\circ\}$) as $X^{(l,\theta)}(\mathbf{n})$ and the coefficients from the same subband with a shift $\mathbf{f} = (f_x, f_y)$ as $X^{(l,\theta)}(\mathbf{n} + \mathbf{f})$, the interpolation process can be written as:

$$X^{(l,\theta)}(\mathbf{n} + \mathbf{f}) \approx \sum_{\mathbf{k}} W_{\mathbf{f}}^{(l,\theta)}(\mathbf{k}) X^{(l,\theta)}(\mathbf{n} + \mathbf{k}), \quad (5.8)$$

where the modulated kernel W is given by:

$$W_{\mathbf{f}}^{(l,\theta)}(\mathbf{k}) = H_{\mathbf{f}}(-\mathbf{k}) e^{j(\boldsymbol{\Omega}^{(l,\theta)})^T(\mathbf{f}-\mathbf{k})}. \quad (5.9)$$

$H_{\mathbf{f}}(\mathbf{k})$ is a 2-D windowed-sinc interpolation kernel, and $\boldsymbol{\Omega}^{(l,\theta)} = (\omega_x^{(l,\theta)}, \omega_y^{(l,\theta)})$ is the horizontal and vertical centre frequency (measured in radians per pixel) of the filter at level l and subband θ . Equation (5.8) holds for any fractional shift \mathbf{f} as long as $f_x, f_y \in [-0.5, 0.5]$. In other words, we can only estimate a displacement less than half of the sampling interval of the current scale (e.g. 4 pixels at level 3). However, there is an easy solution to estimate larger displacement, which is discussed later.

Motion estimation begins at the coarsest level l_{max} , which is fixed at level 4 in our implementation. We define the *subband squared difference* surface ($SD^{(l,\theta)}$) at subpel \mathbf{n} over the variable \mathbf{f} as:

$$SD^{(l,\theta)}(\mathbf{n}, \mathbf{f}) = |X_1^{(l,\theta)}(\mathbf{n} + \mathbf{f}) - X_2^{(l,\theta)}(\mathbf{n})|^2, \quad (5.10)$$

where X_1 and X_2 are the CWT coefficients in the same subband from the reference and distorted images respectively. Equation (5.10) gives rise to an approximately valley-shaped quadratic surface (figure 5.6a):

$$SD^{(l,\theta)}(\mathbf{n}, \mathbf{f}) \approx |X_1^{(l,\theta)}(\mathbf{n})|^2 + |X_2^{(l,\theta)}(\mathbf{n})|^2 - 2|X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})| \cos(\phi_1^{(l,\theta)}(\mathbf{n} + \mathbf{f}) - \phi_2^{(l,\theta)}(\mathbf{n})), \quad (5.11)$$

where $\phi_1^{(l,\theta)}(\mathbf{n} + \mathbf{f})$ and $\phi_2^{(l,\theta)}(\mathbf{n})$ are the phases of the CWT coefficients $X_1^{(l,\theta)}(\mathbf{n} + \mathbf{f})$ and $X_2^{(l,\theta)}(\mathbf{n})$ respectively. Equation (5.11) shows how the phase difference between the CWT coefficients of the reference and distorted images is related to the spatial displacement. The minimum of (5.11) lies along a line parallel to the dominant orientation (the vector $\boldsymbol{\Omega}^{(l,\theta)}$) of the corresponding subband filter. This minimum line gives us the maximum likelihood estimate of the motion in the direction orthogonal

to the orientation of the filter. We combine the $SD^{(l,\theta)}$ surfaces from all the six subbands to form a single *level squared difference* surface ($SD^{(l)}$) at each subpel location \mathbf{n} (figure 5.6b), which is also approximately quadratic:

$$SD^{(l)}(\mathbf{n}, \mathbf{f}) = \sum_{\theta} SD^{(l,\theta)}(\mathbf{n}, \mathbf{f}), \quad \text{and,}$$

$$SD^{(lm)}(\mathbf{n}, \mathbf{f}) \approx a(f_x - f_{x_{min}})^2 + b(f_y - f_{y_{min}})^2 + c(f_x - f_{x_{min}})(f_y - f_{y_{min}}) + d. \quad (5.12)$$

The parameters in (5.12) are derived from X_1 and X_2 [107] and are listed below for completeness:

$$a = \sum_{\theta} |X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})|(\omega_x^{(l,\theta)})^2;$$

$$b = \sum_{\theta} |X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})|(\omega_y^{(l,\theta)})^2;$$

$$c = \sum_{\theta} |X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})|2\omega_x^{(l,\theta)}\omega_y^{(l,\theta)};$$

$$\iota = \sum_{\theta} |X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})|(-2)\omega_x^{(l,\theta)} \left(\angle \left[\frac{X_2^{(l,\theta)}(\mathbf{n})}{X_1^{(l,\theta)}(\mathbf{n})} \right] \right);$$

$$\varrho = \sum_{\theta} |X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})|(-2)\omega_y^{(l,\theta)} \left(\angle \left[\frac{X_2^{(l,\theta)}(\mathbf{n})}{X_1^{(l,\theta)}(\mathbf{n})} \right] \right);$$

$$\varpi = \sum_{\theta} (|X_1^{(l,\theta)}(\mathbf{n})| - |X_2^{(l,\theta)}(\mathbf{n})|)^2;$$

$$+ \sum_{\theta} |X_1^{(l,\theta)}(\mathbf{n})X_2^{(l,\theta)}(\mathbf{n})| \left(\angle \left[\frac{X_2^{(l,\theta)}(\mathbf{n})}{X_1^{(l,\theta)}(\mathbf{n})} \right] \right)^2;$$

and,

$$\mathbf{f}_{\min} = \frac{1}{c^2 - 4ab} (2b\iota - c\varrho, \quad 2a\varrho - c\iota);$$

$$\delta = \varpi - af_{x_{min}}^2 - bf_{y_{min}}^2 - cf_{x_{min}}f_{y_{min}}. \quad (5.13)$$

At the coarsest level, the location of the minimum of this surface $\mathbf{f}_{\min} = (f_{x_{min}}, f_{y_{min}})$ (scaled by $2^{l_{max}}$) is the coarse motion estimate of the block of $2^{l_{max}} \times 2^{l_{max}}$ pixels centred on $2^{l_{max}}\mathbf{n}$ in the distorted image. We define the *curvature matrix* Λ around

the motion estimate as:

$$\Lambda = \begin{bmatrix} 2a & c \\ c & 2b \end{bmatrix}. \quad (5.14)$$

The eigenvalues Λ_1, Λ_2 (with $\Lambda_1 > \Lambda_2$) of Λ and the corresponding eigenvectors \mathbf{e}_{\max} , \mathbf{e}_{\min} are used as a directional confidence measure at level l_{max} . In other words, the motion vector is most reliable in the direction of \mathbf{e}_{\max} and vice versa.

If we suspect the motion is larger than half of the sampling interval at the coarsest level, we calculate the $SD^{(l)}$ surfaces using corresponding CWT coefficients in the distorted and reference images, and also from CWT coefficients in the distorted image and those in the adjacent locations in the reference image. We then choose the neighbour (at each subpel \mathbf{n}) which gives the smallest $SD^{(l)}$. This increases the range of motion estimation by $2^{l_{max}}$ pixels and is done only at the coarsest level.

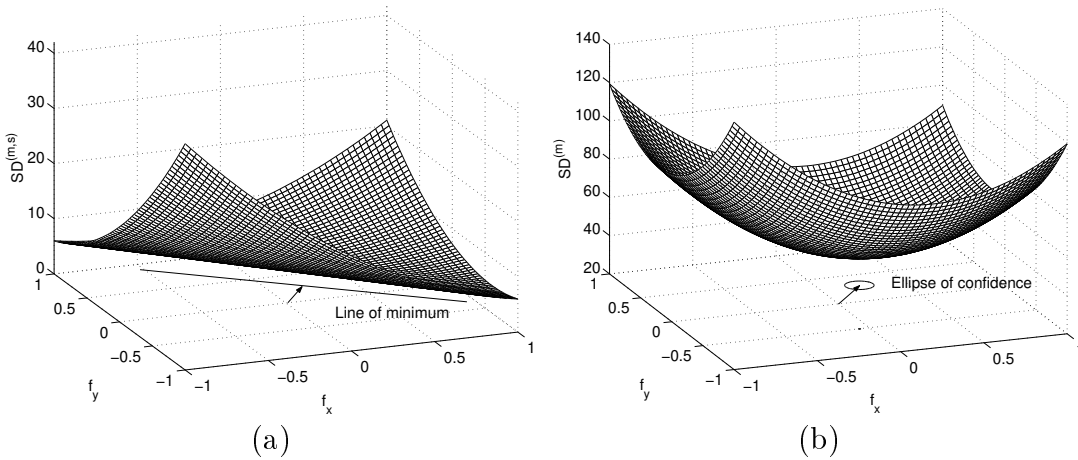


Figure 5.6: (a) A typical *subband squared difference* surface of the -45° subband. The motion estimate orthogonal to this direction is the vector from the origin to the projection of the line of minimum of the surface. We have one such surface per subpel location per subband. (b) When we combine the $SD^{(l,\theta)}$ surfaces from all six subbands, we form the *level squared difference* surface $SD^{(l)}$ at each subpel. The location of the minimum of this surface is the motion vector and the projection of the surface contour onto the xy plane is the elliptic contour of confidence.

The CWT coefficients at finer levels are used to refine the coarse motion estimate. We incorporate the *level squared difference* surface into a *cumulative squared difference*

surface ($CSD^{(l)}$) defined as follows:

$$CSD^{(l)}(\mathbf{n}, \mathbf{f}) = \begin{cases} CSD^{(l+1)}(\mathbf{n}, \mathbf{f}) + SD^{(l)}(\mathbf{n}, \mathbf{f}) & \text{if } l < l_{max} \\ SD^{(l)}(\mathbf{n}, \mathbf{f}) & \text{if } l = l_{max} \end{cases}. \quad (5.15)$$

The location of the minimum of $CSD^{(l)}$ now gives us the motion estimates at level l . The curvatures⁴ at each location \mathbf{n} give us the confidence measure of individual motion vectors \mathbf{f} . The field $CSD^{(l+1)}(\mathbf{n}, \mathbf{f})$ is a bilinearly interpolated version of $CSD^{(l+1)}(\mathbf{n}, \mathbf{f})$ because the density of $SD^{(l)}(\mathbf{n}, \mathbf{f})$ is four times that of $CSD^{(l+1)}(\mathbf{n}, \mathbf{f})$. In addition, the curvatures of $SD^{(l)}$ are corrected by subtracting a ‘shallow circular bowl’ centred on the minimum of $SD^{(l)}$. We call this *curvature correction*. The effect of this is to minimise the curvature of the quadratic surfaces parallel to straight edges. This is important in correctly identifying motion vectors which suffer from the *aperture problem*. Finally, any erroneous motion vectors are corrected (see next section) and the associated $SD^{(l)}$ surfaces recalculated before being incorporated into $CSD^{(l)}(\mathbf{n}, \mathbf{f})$. The corrected motion field forms the starting motion estimates at level $l - 1$. The level at which the algorithm terminates depends on desired density of the final motion field. In image registration, we found that having one motion vector per 4×4 pixels (i.e. terminating at level 2) is sufficient. Besides, the CWT coefficients at level 1 are in general too noisy to allow reliable motion estimation.

5.3.2.3 Motion vector correction

Identifying erroneous motion vectors Under ideal conditions when there are corner features everywhere in the image and the distorted image suffers no degradation, most of the motion vectors will be correct. However, this rarely happens in practice. It is important to correct wrong motion vectors, otherwise the watermark decoder performance will be poor in areas where the registered image is still misaligned with the watermark sequence. There are two scenarios which may result in erroneous motion vectors:

1. When a motion vector lies near an edge, only the component of motion perpendicular to the edge can be reliably estimated. This is the well known aperture problem. We denote this as *Type I Error*. Each of these vectors has one component which is reliable.

⁴We calculate Λ using parameters from $CSD^{(l)}$ rather than $SD^{(l)}$.

2. When the underlying region is either featureless, too noisy or has repetitive patterns, the estimated motion vectors may be completely wrong, even though the associated confidence (equation (5.14)) may be high. We denote this as *Type II Error*. Such vectors are completely unreliable and should be removed.

Figure 5.7 shows an example of the initial motion vectors at the coarsest level, together with the elliptic contour of confidence around each motion vector. After curvature correction (see section 5.3.2.2), the associated elliptic contours around the motion vectors, which suffer from aperture problem, appear as ‘long and thin’, with the component of motion along the minor axis being reliable. The contours of other motion vectors are not affected. Thus we can identify *Type I Error* when the ratio of the major axis to the minor axes of the elliptic contour exceeds a threshold. To identify *Type II Error*, we utilise the smoothness constraint of the motion field. Since geometric distortion results in an image that resembles the original, the true motion field must be smooth, as any discontinuity will result in visual artifacts. Unreliable vectors thus manifest themselves as discontinuities. We detect discontinuity in both magnitude and phase of the motion vectors. If the magnitude or the phase of a vector differs too much from its (reliable) neighbours, we flag it as being erroneous. Once all the erroneous motion vectors have been identified and removed, we need to fill in the ‘missing information’. Before describing the interpolation algorithm, we will first discuss the interpolation technique employed.

Surface interpolation using radial basis functions Radial basis functions (RBF) have been widely used in medical imaging for reconstructing surfaces [28]. They are particularly useful in cases where the holes to be filled are relatively large⁵. A surface is regarded as a single-valued function \mathcal{P} of two variables: $\mathcal{P} : \mathbb{R}^2 \mapsto \mathbb{R}$, where the interpolation nodes, at which the value of the function is known, do not generally lie on a regular grid. The problem is to approximate \mathcal{P} with a smooth function which provides at least C^1 continuity so that there are no creases in the reconstructed surface. Given the values of \mathcal{P} at locations $(x_1, y_1), \dots, (x_N, y_N)$, the radial basis function q , which approximates \mathcal{P} , is given by:

$$q(\mathbf{n}) = g(\mathbf{n}) + \sum_{i=1}^N \lambda_i \psi(r_i), \quad (5.16)$$

⁵If the holes were small, we could use much simpler techniques such as cubic splines or piecewise polynomial interpolation.

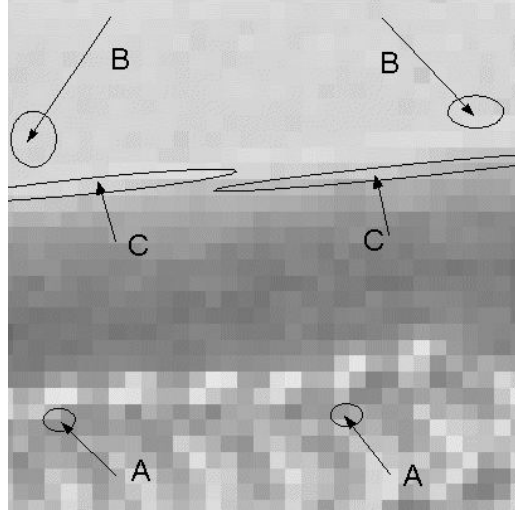


Figure 5.7: Example of motion vectors (black arrows) in a 32×32 pixel block, together with individual elliptic contour of confidence. The smaller the contour, the more reliable the vector. The vectors labelled ‘A’ are thus relatively reliable, whereas the vectors labelled ‘B’ are probably wrong since they are much bigger and point in directions which are different from other vectors. There is a dominant horizontal edge in the middle of the pixel block and the contours of the vectors lying near the edge (labelled ‘C’) appear as ‘long and thin’. The true motion is 4 pixels upward and 4 pixels to the left.

where,

$$\begin{aligned} \mathbf{n} &= (x, y); \quad \mathbf{n}_i = (x_i, y_i); \quad \lambda_i \in \mathbb{R}; \\ r_i &= |\mathbf{n} - \mathbf{n}_i| = \sqrt{(x - x_i)^2 + (y - y_i)^2}; \\ g(\mathbf{n}) &= g_0 + g_1x + g_2y. \end{aligned}$$

Thus the RBF is a linear combination of translated versions of a radially symmetric function, plus a low degree polynomial. $\psi(r)$ is a fixed function and is given by:

$$\psi(r) = r^2 \log(r). \quad (5.17)$$

This is the so called thin-plate spline. The coordinates (x, y) also have to be normalised such that all data points lie on a unit square. The parameters g_0, g_1, g_2 and λ_i in (5.16) must satisfy the following equations:

$$q(\mathbf{n}_i) = \mathcal{P}(\mathbf{n}_i), \quad i = 1, 2, \dots, N, \quad \text{and}, \quad (5.18)$$

$$\sum_{i=1}^N \lambda_i h(\mathbf{n}_i) = 0, \quad \text{for } \forall h \in \pi^2. \quad (5.19)$$

π^2 is the space of all polynomials h of two variables with degree at most the same as g . Equation (5.18) says q must pass through all the nodes at which the value of \mathcal{P} is known, while equation (5.19) makes q tend to a flat surface with normal in the direction $(-g_1, -g_2, 1)$, (which should be orthogonal to the planar component of the surface being fitted), as we get further away from the nodes. The solution is given by:

$$\begin{bmatrix} \lambda_1 \\ \vdots \\ \lambda_N \\ g_0 \\ g_1 \\ g_2 \end{bmatrix} = \begin{bmatrix} \psi(r_{11}) & \psi(r_{12}) & \cdots & \psi(r_{1N}) & 1 & x_1 & y_1 \\ \vdots & \vdots & \cdots & \vdots & \vdots & \vdots & \vdots \\ \psi(r_{N1}) & \psi(r_{N2}) & \cdots & \psi(r_{NN}) & 1 & x_N & y_N \\ 1 & 1 & \cdots & 1 & 0 & 0 & 0 \\ x_1 & x_2 & \cdots & x_N & 0 & 0 & 0 \\ y_1 & y_2 & \cdots & y_N & 0 & 0 & 0 \end{bmatrix}^{-1} \begin{bmatrix} \mathcal{P}(\mathbf{n}_1) \\ \vdots \\ \mathcal{P}(\mathbf{n}_N) \\ 0 \\ 0 \\ 0 \end{bmatrix}, \quad (5.20)$$

where $r_{ij} = |\mathbf{n}_i - \mathbf{n}_j|$. The bottleneck of finding the parameters of the RBF is the inversion of the $(N + 3) \times (N + 3)$ matrix in (5.20). At present, we form a window around the missing data and expand the window until it captures enough reliable motion vectors (the neighbourhood size is set to 200 vectors) and then fit an RBF to the resulting window. We found that the algorithm is efficient enough even if we are only using brute force matrix inversion. Unfortunately, this limits the maximum number of vectors we can handle at a time to about 200. Once we obtain the parameters of q , we evaluate (5.16) at all places where there is missing data. In other words, we only need to evaluate (5.20) once for each surface to be interpolated. Figure 5.8 shows an example of a surface interpolated with an RBF.

Correcting rejected motion vectors There are two phases in interpolating erroneous motion vectors. The vectors suffering from the aperture problem are interpolated first before the completely unreliable vectors are dealt with. For each *Type I Error* motion vector, we resolve the surrounding reliable motion vectors along the direction of the unreliable component of the vector concerned, which results in a surface. We then fit an RBF to this surface and evaluate a probable value for the unreliable component of this vector, which is then combined with the reliable component to form the corrected vector. The procedure is repeated for each *Type I Error* vector.

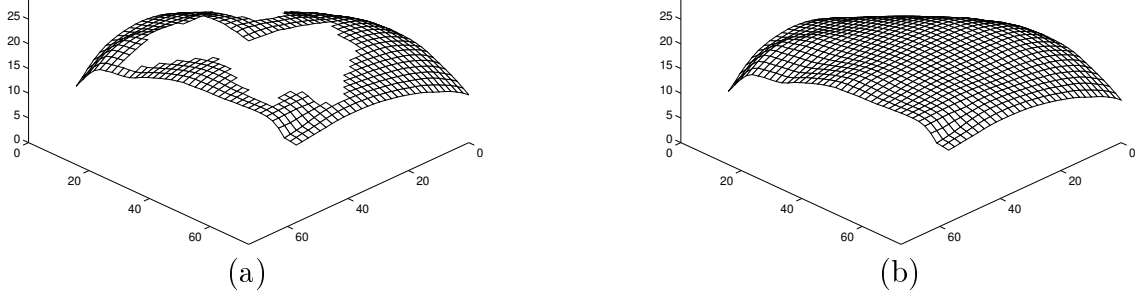


Figure 5.8: An example of RBF interpolation. (a) shows a surface with holes. (b) shows the reconstructed surface interpolated by an RBF.

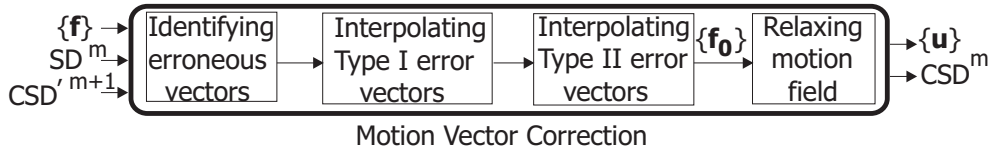


Figure 5.9: Inside the black box “Motion Vector Correction” of figure 5.5. The inputs are the initial motion estimates $\{\mathbf{f}\}$, the current *level squared difference* surface $SD^{(l)}$ and the bilinearly interpolated *CSD* surface from the previous level. The outputs are the smoothed and corrected motion field $\{\mathbf{u}\}$, and the *CSD* surface of the current level.

The newly corrected *Type I Error* vectors are then treated as reliable vectors during the interpolation of *Type II Error* vectors. Fortunately, the *Type II Error* vectors tend to form clusters, which means we typically need fewer RBFs than for *Type I Error* vectors. For each cluster of such vectors, we form two surfaces; with one using the vertical component of the surrounding reliable vectors, and the other using the horizontal component. After RBFs are fitted to them, these two surfaces are recombined to form the corrected *Type II Error* vectors. After all the vectors are corrected, the $SD^{(l)}$ surfaces (5.10) are recalculated at the corrected vectors. Finally, we apply a relaxation procedure [15, 106] to the corrected motion field, in order to take into account the relative confidence of each motion vector, and to reduce the effect of noise in our measurements. The relaxed motion field $\{\mathbf{u}\}$ now becomes the initial motion estimate of the next level. Figure 5.9 summarises the overall correction procedure.

5.3.2.4 Experimental results

The motion estimation algorithm terminates at level 2, resulting in one motion vector per 4×4 pixels. At such small scale, the distortion can be approximated as simple

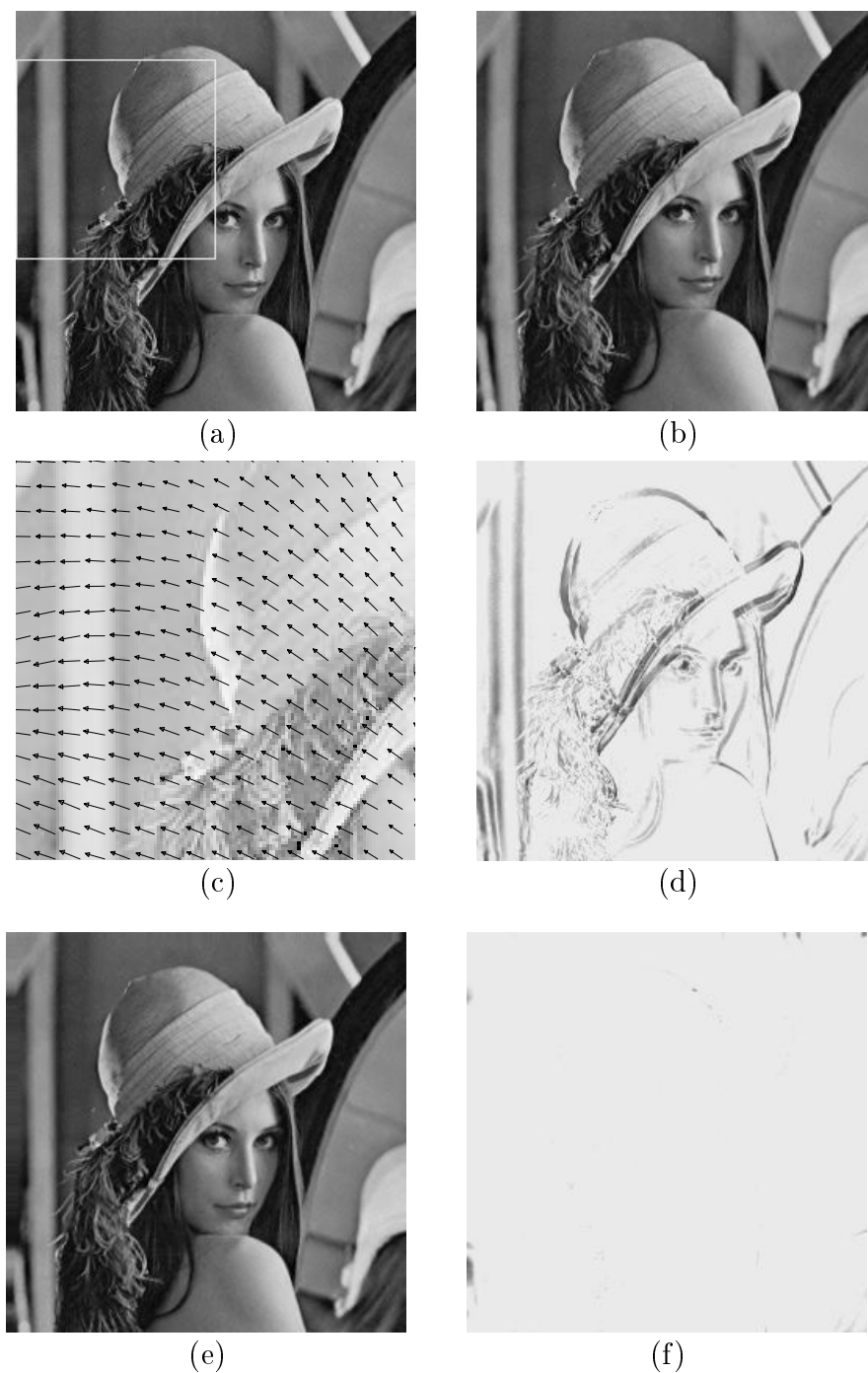


Figure 5.10: A registration example. (a) The reference image, which is watermarked. (b) After applying StirMark with a bending factor of 3.0. (c) The estimated motion field at level 3 in the highlighted area in (a). The distortion is clearly not affine across the whole region, but in a small neighbourhood, the distortion is close to being affine. (d) Difference between (a) and (b). The darker the colour, the bigger the difference. RMS error is 39.9. (e) The registered image. (f) Difference between (a) and (e). The registration process has almost completely inverted the distortion. RMS error (excluding the boundary regions) is reduced to 7.1.

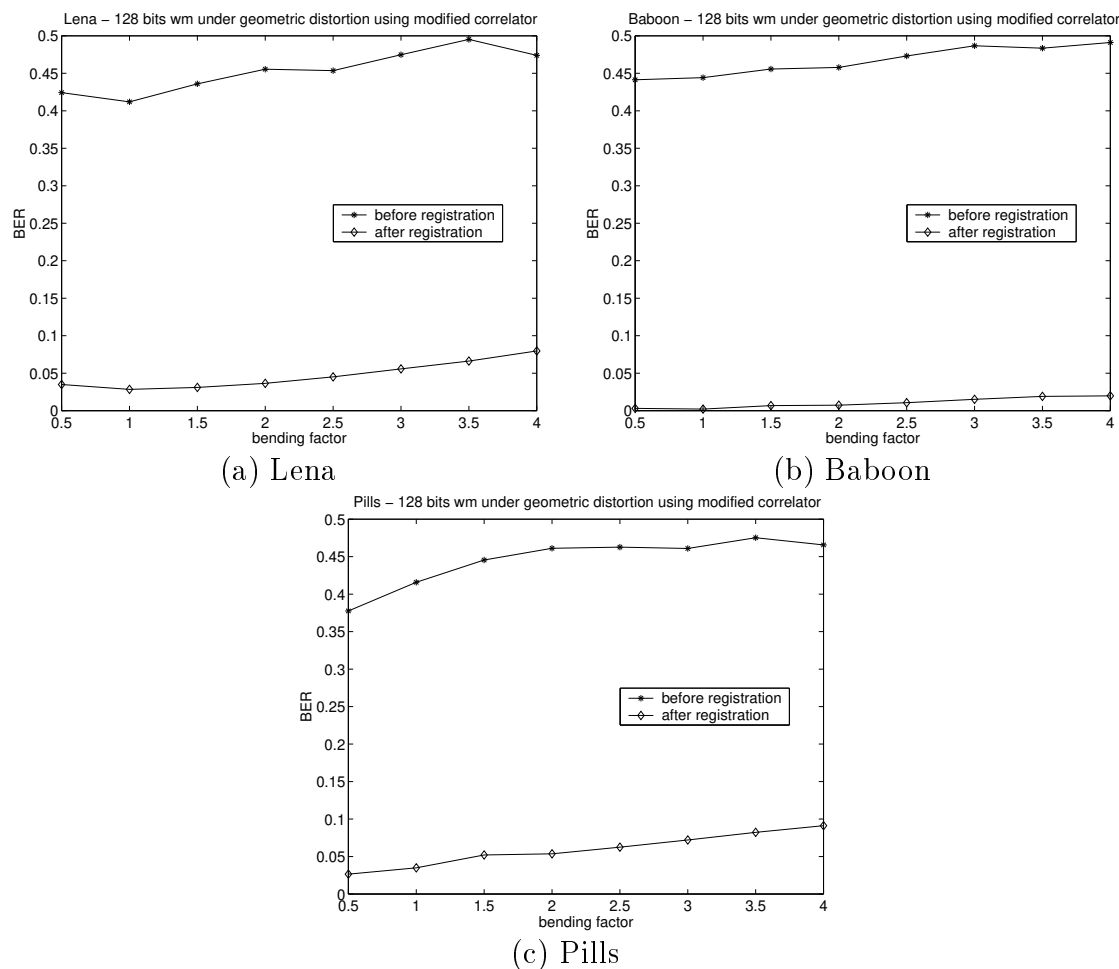


Figure 5.11: Watermark decoding on the 3 test images attacked by geometric distortion, before and after image registration. The higher the bending factor, the bigger is the distortion induced. The BER before and after registration is more or less independent of the bending factor, which shows that even a slight degree of distortion is enough to jeopardise the decoder and registration is essential to correct watermark decoding. The residual BER after registration is due to part of the image boundary regions being removed by the attack which cannot be recovered.

translation given by the motion vector. To invert the distortion, we simply have to move each pixel block back to its original location. Cubic spline interpolation [145] is used to obtain image intensities at subpixel accuracy. Since the geometric distortion attack removes some parts of the image which lie close to the image boundaries, these regions cannot be reconstructed during registration. However, since these regions are small compared with the image, this has small effects on the decoder performance. Figure 5.10 shows an example of the registration algorithm. We can see that the distortion is not affine in a global scale, but is close enough to affine on a small scale. The proposed registration algorithm succeeds in aligning most of the edges in the

image correctly. The effectiveness of registration is also tested with our blind spread spectrum watermarking scheme (in the CWT domain) proposed in chapter 4. The benchmark software (version 3.1) described in [95] is used in all simulations. Figure 5.11 shows the effect of registration on watermark decoding on the 3 test images Lena, Baboon and Pills. The bending factor indicates the amount of distortion introduced. We can see that even a small amount of distortion (small bending factor) is enough to confuse the decoder, and the BER prior to registration approaches 0.5. The BER after registration is greatly reduced, no matter how much the image has been distorted. There is a *residue* BER after registration because part of the image boundary has been removed during the geometric distortion. However, this is only an implementation issue, which can be eliminated if we do not use the image boundary regions for watermarking. The effects of registration on other existing spread spectrum based watermarking schemes ([40, 84, 165, 166]) are also tested and the results can be found in [103]. In all cases, our proposed registration algorithm improves the performance of existing spread spectrum based watermarking schemes significantly under geometric distortion attacks.

5.3.3 Registration of rotated or scaled images

Unfortunately the registration algorithm described previously was designed specifically to handle relatively small distortion and it cannot handle larger distortions like rotation or scaling. Template matching, where a known pattern of peaks is embedded in the DFT domain, as mentioned in section 5.3.2.1 and discussed in [120], can be used to invert rotation, scaling, shearing, etc. However, an attacker can simply compute the DFT of the watermarked image, identify the peaks and remove them, followed by affine transforming the image to defeat the registration mechanism. This is the so called template removal attack [68]. In order to be robust against the template removal attack, the attacker must not be able to remove the template without destroying the image. A solution is to use the autocorrelation function of the watermark as the template [91]. If multiple copies of the same watermark, each shifted by a different amount, are overlaid on top of one another, the resulting autocorrelation function will feature additional peaks, which can be used as a template. The maximum translation detectable by this technique is limited by the displacement between the different copies of the watermark (chapter 5 of [92]), and the larger is this displacement, the smaller are the peaks in the autocorrelation function. Hence there

is a tradeoff. A better implementation is to use a periodic watermark sequence⁶ as suggested by Voloshynovskiy *et al.* in [149]. The DFT of the watermark will have a regular grid pattern of peaks and the grid period is related to watermark sequence period. This grid pattern can be easily detected by computing another DFT of the *magnitude* of the watermark DFT, and grid period will show up as a prominent peak in this second DFT. The rest of the registration algorithm is identical to the template matching approach discussed at the beginning of this section. In both of these approaches, the watermark detector has to compute an estimate of the watermark before extracting the template. Yet another approach for watermark decoding in rotated/scaled images is to embed watermarks in a rotation/scale invariant domain (e.g. the Fourier-Mellin domain [120]). However, this approach has some known problems: existence of interpolation error during inverse transform and the inability to detect a general affine transform which cannot be described by rotation or scaling, although the former problem was later solved by the authors using the Chirp-Z transform [121]. Registration of rotated, scaled or cropped images is a solved problem and is not considered in this thesis.

As mentioned in section 5.3.2.1, the geometric distortion attack can be well approximated as an affine transform on a small scale. Therefore one can perhaps use a template approach (for example, [149]) on a blockwise basis and perform the registration process separately in each block prior to watermark detection. This is probably a direction of future research. A possible solution to watermarking resistant to geometric distortion is suggested by Kutter *et al.* in [94], where the image is segmented according to some features and the watermark is embedded in each segment separately.

5.4 Robustness of watermarks against denoising attacks

5.4.1 The denoising attack

The denoising attack aims to estimate the watermark from the image and removes it. This is in contrast compared with the geometric distortion attack described in the previous section, which alters the geometry of the image *without* removing the

⁶The watermark is still a pseudo-random sequence, but the entire watermark sequence is made by concatenating one (shorter) random sequence. Thus the watermark sequence exhibits periodic structure.

watermark. As mentioned in section 4.4.4, the detector of many additive watermarking schemes first estimates the watermark prior to detection. The denoising attack uses this weakness to its advantage. This problem arises because all spread spectrum based watermarking schemes are linear, and the watermark is generated *independently* (apart from the perceptual mask) from the host image and is simply added to the image afterwards. Thus it is straightforward to estimate the watermark component in higher frequency bands, where the host image usually has little power, without the knowledge of the secret key used in embedding. A commonly used estimator is the MAP estimator given by:

$$\hat{s} = \arg \max_s \{p_x(x'|s) \cdot p_s(s)\}, \quad (5.21)$$

where p_x is the host pdf and x' is the watermarked image. Under the assumption that both the watermark and the image are locally i.i.d. Gaussian, (5.21) becomes the familiar Wiener filter:

$$\hat{s} = \frac{\sigma_s^2}{\sigma_s^2 + \sigma_x^2} (x' - \bar{x}'), \quad (5.22)$$

where,

- \hat{s} is the estimated watermark.
- x' is the watermarked image with \bar{x}' being its local mean.
- σ_s^2 and σ_x^2 are the local variances of the watermark (including perceptual scaling) and the host image respectively.

The denoised image y is obtained by subtracting the estimated watermark from the watermarked image.

$$y = \bar{x}' + \frac{\sigma_x^2}{\sigma_s^2 + \sigma_x^2} (x' - \bar{x}'). \quad (5.23)$$

Wavelet shrinkage [151] is also a special case of the denoising attack, where the image prior is Laplacian. Su *et al.* [140, 138] derive an optimal filter and additive noise spectrum based on minimising the capacity of the image. Their filter differs from the Wiener filter under the Gaussian image distribution assumption. The authors argued that, in order for the image to resist estimation based attacks, the watermark should perceptually look like the host image, which is referred to as the *Power Spectrum*

Condition [139] by the authors. In the frequency domain, this translates to:

$$\frac{P_w(\omega)}{G(\omega)} \propto P_x(\omega), \quad (5.24)$$

where $P_w(\omega)$ and $P_x(\omega)$ are the power spectrum of the watermark and the host image, and $G(\omega)$ is the gain of the watermark. This assumes the gain of the watermark is approximately equal to the reciprocal to the visual sensitivity of that particular frequency. In [140], the authors assume the signal is stationary and Gaussian, which does not hold for images in general. However, typical images can be treated as locally stationary. Thus we can apply the attack on a blockwise basis. We can see that (5.24) is satisfied by the gain factor used in our CWT watermarking scheme (equation 3.7), with the term $k^2|x|^2$ adapting the watermark energy closely to the local activity of the host image and we expect the proposed scheme to resist denoising attacks. Denoising attacks are actually closely related to compression. Compression aims to remove the insignificant components of the image, whereas denoising also tends to remove insignificant components of the image, which are usually dominated by noise. As we will see later, the effects of denoising attacks are very similar to that due to compression. We use the modified correlator for decoding the watermark under denoising attacks, because the simulation results in section 5.2.3 show that we can obtain hardly any improvement by using the generalised Gaussian decoder, for decoding CWT watermarks under compression. Recently, a more advanced attack which involved changing the sign of the estimated watermark at random locations, so that the correlation between the estimated watermark and the expected watermark is reduced to as close to zero as possible, is suggested by Voloshynovskiy *et al.* in [152]. The attack involves adding random noise to the denoised image, with the sign of the random noise chosen to be *opposite* to that of the estimated watermark at about half of the locations, and the same for the rest of the locations. In other words, the noise is *dependent* on the denoised image. The authors refer to this as the *remodulation* attack and they showed the attack is very effective against common spread spectrum based schemes and none of the schemes the author tested survives the attack.

5.4.2 Simulation results and discussion

We investigate the robustness of our spread spectrum watermarking scheme under denoising and denoising with remodulation. In the denoising attack, we divide the image wavelet coefficients into trees and use the attack proposed by Su [140] on each

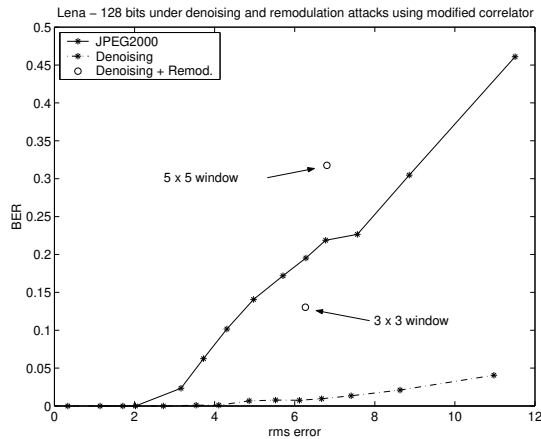


Figure 5.12: Comparing denoising attack (with and without remodulation) with JPEG compression. The linear filter suggested by Su [140, 138] is used as the denoising filter and remodulation attack uses a 3×3 and 5×5 window to estimate the watermark. The denoising attack from Su is not as effective as JPEG2000 but the remodulation attack seems effective. However, the remodulation attack produces perceptually slightly worse images than compression and denoising (see figure 5.13).



Figure 5.13: Lena after JPEG2000 compression (a), denoising (b) and denoising with remodulation attacks with 3×3 window for estimating the watermark sign (c). Denoising attacks blur the image in a manner similar to compression, whereas the remodulated image looks noisier. The RMS error of the all images is about 6.3.

of the trees independently. A tree is defined as the wavelet coefficients at a particular spatial location in all subbands at level 3 of the CWT transform, plus the children (in level 2), and the grandchildren (in level 1) of these coefficients. Each tree is treated as a stationary signal. The squared magnitude of the CWT coefficients is used as an estimate of the power spectrum of the signal and the CWT coefficients are scaled down in magnitude according to Su's filter followed by addition of *independent* noise with power spectrum given in [140]. In the second test, we employ the benchmark

software developed by Pereira *et al.* [122] to implement the remodulation attack with 3×3 and 5×5 windows for estimating the watermark⁷. Figure 5.12 shows the results of Su's linear filtering attack together with denoising with remodulation compared with JPEG2000 compression. Only the results for the Lena image are shown and the modified correlation decoder (section 4.4.3) is used in all cases. The attack suggested by Su is not as effective as compression in removing the watermark when we compare the BER at a given level of RMS error. This is because the authors' assumption of Gaussian signal does not hold for images. The remodulation attack is more effective than denoising with the addition of independent noise as in [140]. However, one must note that the RMS error does not give an accurate picture of the distortion introduced to the image, because the denoising and remodulation attacks result in different kinds of artifacts. This is apparent in figure 5.13, which shows the Lena image after JPEG2000 compression (at 0.5 bpp), denoising and remodulation attacks respectively. Denoising attack blurs the image in a manner similar to compression, whereas the remodulated image on the right looks noisier, even though all the images have approximately the same RMS error. The remodulation attack implemented in [122] is actually not as effective as the results suggest. This is partly because the current version of the remodulation attack employs a very primitive perceptual model. Nevertheless, we expect the remodulation attack will be very effective when a better perceptual model is used.

5.5 Chapter summary

In this chapter, we discussed watermark decoding in the presence of compression, geometric distortion and denoising attacks. We modelled the host image coefficients under compression as a generalised Gaussian distribution and introduced the generalised modified matched filter for decoding in such noise. However, experimental results of the new decoder showed little improvement over the modified matched filter discussed in the previous chapter, and in the case of the DCT and the DWT domain, it got worse. This is probably due to the fact that inverse scaling has the effect of flattening the distribution of the CWT coefficients and that the inversely scaled DWT and DCT coefficients in middle frequency bands cannot be described well by a generalised Gaussian distribution. Therefore, using the modified matched filter is good enough for decoding under compression. We then described a novel image reg-

⁷The window used for remodulation is always 3×3 , only the window used for estimating the watermark sign changes. Therefore the RMS error introduced in the two cases are very similar.

istration algorithm based on motion estimation for combating geometric distortion. Experimental results showed that our proposed algorithm can greatly improve the performance of spread spectrum based watermarking systems. Finally, we discussed denoising and remodulation attacks and showed that our CWT watermark scheme allows the watermark to closely follow the image power spectrum and resist denoising attacks. On the other hand, the remodulation attack is effective against our system. Although the current implementation of the attack produces more distorted images, we expect the remodulation attack will be effective in practice if a sophisticated visual model is used. In the next chapter, we present a new approach to watermarking which exploits the knowledge of host image at the embedder.

Chapter 6

Watermarking as Communications with Side Information

6.1 Introduction

Early watermarking schemes usually treat watermarking as a communication process through a very noisy channel, where the original host image acts as interference at the decoder. Spread spectrum techniques are used to overcome this noise. However, the interference at the decoder is not *completely* unknown, because the encoder has access to the original image. Thus watermarking should be treated as communication with side information at the encoder [43]¹. (This is sometimes also known as informed watermarking). In his remarkable paper [38], Costa argues that if the channel state (side information) is known at the encoder, then the capacity of the communication channel does *not* depend on whether the decoder has access to the channel state or not. Instead of trying to cancel out the channel interference, the encoder chooses codewords in the direction of the interference such that, with high probability, the decoder can distinguish between these codewords. This is explained in more detail later. We start this chapter by reviewing the theory of communication with side information due to Costa and two practical implementations of Costa's idea based on uniform quantisation. We then contrast and compare spread spectrum and quantisation based watermarking and introduce the concept of spread transform watermarking as a hybrid combination between quantisation and spread spectrum. Our implementation of image adaptive spread transform based watermarking scheme will

¹The way we combine information from different subbands during watermark decoding is sometimes referred to as communication with side information at the decoder [148].

be described and we will compare it with our spread spectrum based watermarking scheme proposed earlier. The application of error control codes and the problem of watermark detection in quantisation based watermarks are also addressed.

6.2 Communication with side information and quantisation based watermarking

6.2.1 Definition of problem

The watermarking process with side information is depicted in figure 6.1. The following symbols will be used in our discussions.

- \mathbf{p} is our payload of L bits to be transmitted. There are 2^L possible messages.
- \mathbf{x} is the vector of host image coefficients in some transform domain.
- \mathbf{x}' is the vector of watermarked coefficients.
- $\mathbf{w} = \mathbf{x}' - \mathbf{x}$ is the vector of watermark signal.
- \mathbf{y} is the vector of received (and possibly corrupted) coefficients, from which we extract the best estimate of the embedded payload $\hat{\mathbf{p}}$.
- $\mathbf{e} = \mathbf{y} - \mathbf{x}'$ is the noise vector.
- $p(\mathbf{y}|\mathbf{x}')$ is the transition function representing the attacks on the watermarked image.

It is desired to send a message \mathbf{p} of length L bits through the watermarking channel. The watermark channel $p(\mathbf{y}|\mathbf{x}')$ is not known to either the encoder or the decoder, but the encoder has access to the cover signal \mathbf{x} . This is in contrast to figure 4.1, where no knowledge of the host is exploited at the encoder, which is typically the case for spread spectrum based watermarking systems. By exploiting the knowledge of the cover image at the watermark encoder, one can design an embedding scheme where the interference due to the host is almost eliminated at the decoder. How to design such a watermarking system is explained in the next section.

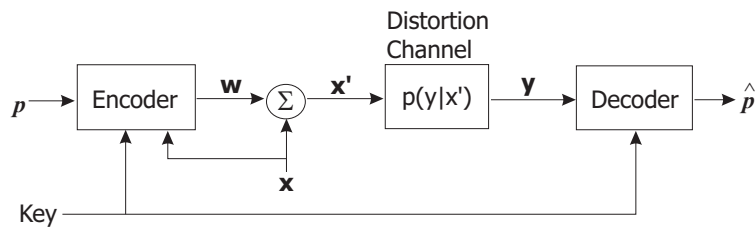


Figure 6.1: Watermarking as a communication process with side information. \mathbf{p} is the payload. The encoder produces a watermark \mathbf{w} based on \mathbf{p} and the cover signal \mathbf{x} , which is then added to cover signal to form the watermarked signal \mathbf{x}' . The decoder receives a (possibly) corrupted signal \mathbf{y} and returns an estimate of the original message $\hat{\mathbf{p}}$.

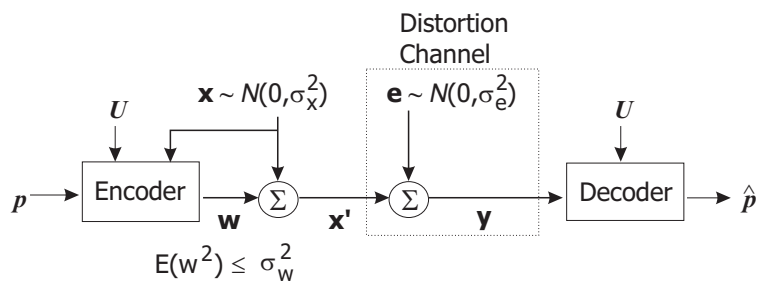


Figure 6.2: A special case of communication with side information where both the cover signal and the interference are Gaussian with variance σ_x^2 and σ_e^2 respectively, and the variance of the embedded signal is constrained to σ_w^2 . U is an auxiliary variable defined in section 6.2.2. Costa [38] shows that the capacity of such a channel is independent of σ_x^2 and only depends on σ_e^2 and σ_w^2 .

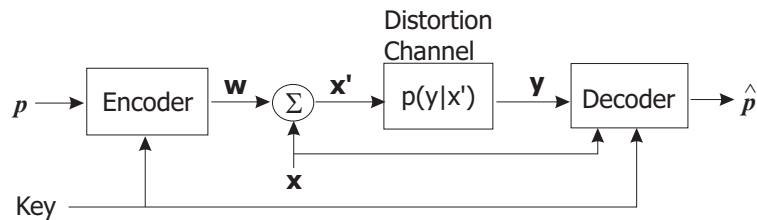


Figure 6.3: In non-oblivious watermarking schemes, the decoder knows the host image, so the host image does not interfere with the watermark. Costa showed that the capacity of a blind watermarking scheme, where the knowledge of the host is exploited at the encoder, is the same as this case.

6.2.2 Costa's solution

Costa [38] considers the system in figure 6.1 where \mathbf{x} and the additional interference \mathbf{e} are *independent* and distributed as Gaussian with variances σ_x^2 and σ_e^2 . The energy of the signal to be transmitted \mathbf{w} is constrained such that $E(w^2) \leq \sigma_w^2$. This is shown in figure 6.2. The most important part of Costa's solution is the design of the auxiliary variable U , the codebook of the encoder. One can think of the codebook as the pseudo-random sequences used to generate the watermark signal. The capacity

C of the channel in figure 6.2 is given by [60, 64]:

$$C = \max_{p(x)p(u,w|x)p(y|w,x)} I(u; y) - I(u; x), \quad (6.1)$$

where $I(a; b)$ is the mutual information between variables a and b , and the maximisation is over the joint distribution $p(x)p(u, w|x)p(y|w, x)$. Costa gives the optimal codebook of the form:

$$u = w + \alpha x, \quad (6.2)$$

where α is given by $\sigma_w^2/(\sigma_w^2 + \sigma_e^2)$, and $w \sim \mathcal{N}(0, \sigma_w^2)$, $x \sim \mathcal{N}(0, \sigma_x^2)$. w and x are our embedded signal and the host signal as defined previously. The corresponding capacity is:

$$C = \frac{1}{2} \log \left(1 + \frac{\sigma_w^2}{\sigma_e^2} \right). \quad (6.3)$$

In other words, this is the same as the case of non-oblivious watermarking with Gaussian interference (figure 6.3), which is the upper bound of achievable capacity, since the only interference in the system is due to the attacks on the watermarked image. Spread spectrum based watermarks are a special case of (6.2) with $u = w$, i.e. $\alpha = 0$, because the watermark is the same as the codebook, the pseudo-random sequences. The capacity of spread spectrum systems (assuming Gaussian host and Gaussian interference) is given by:

$$C = \frac{1}{2} \log \left(1 + \frac{\sigma_w^2}{\sigma_e^2 + \sigma_x^2} \right), \quad (6.4)$$

which is clearly suboptimal because the host power limits the capacity of the channel. However, the codebook $u = w$ will be optimal if \mathbf{x} is available at the decoder [115] (non-oblivious watermarking). We now summarise the main steps of Costa's scheme (for more details please refer to [38, 39]):

1. First we design \mathcal{U}^N , our codebook with entries \mathbf{u} in N dimensions, where N is the length of the signal. The entries \mathbf{u} are random Gaussian sequences generated as follows:

$$\mathcal{U}^N = \{\mathbf{u}_i = \mathcal{N}(0, \sigma_w^2 + \alpha \sigma_x^2) \mathbf{1} \mid i \in \{1, \dots, M\}\}. \quad (6.5)$$

The size of the codebook M is $2^{\lceil N \cdot I(\mathbf{u}; \mathbf{y}) - \epsilon \rceil}$. The codebook is then partitioned

uniformly into 2^L non-overlapping sub-codebooks \mathcal{U}_i^N ($i \in \{1, \dots, 2^L\}$), each with approximately $M/2^L$ entries. These sub-codebooks are available at both the encoder and the decoder.

2. Assume the host vector is \mathbf{x} and a message p ($1 \leq p \leq 2^L$) is to be transmitted. First we search in the p^{th} sub-codebook \mathcal{U}_p^N for a sequence \mathbf{u}_0 which is jointly typical with \mathbf{x} . In other words, we look for \mathbf{u}_0 such that $\mathbf{w} = \mathbf{u}_0 - \alpha\mathbf{x}$ is approximately orthogonal to \mathbf{x} . The encoder declares an error if no such sequence is found. However, the probability of this happening is arbitrarily small as $N \rightarrow \infty$.
3. The watermarked signal is computed from $\mathbf{x}' = \mathbf{x} + \mathbf{w}$, which is then sent through the channel.
4. The decoder receives signal \mathbf{y} and searches through the *whole* codebook \mathcal{U}^N for a sequence \mathbf{u}_1 such that $(\mathbf{u}_1, \mathbf{y})$ is jointly typical. An error occurs if none or more than one such sequence exists. With high probability, the decoder will only find one such sequence, which is the same as \mathbf{u}_0 . The index \hat{p} of the sub-codebook $\mathcal{U}_{\hat{p}}^N$, from which \mathbf{u}_1 is found, is the estimated message transmitted. The error probability approaches zero exponentially as $N \rightarrow \infty$.

This is illustrated in figure 6.4.

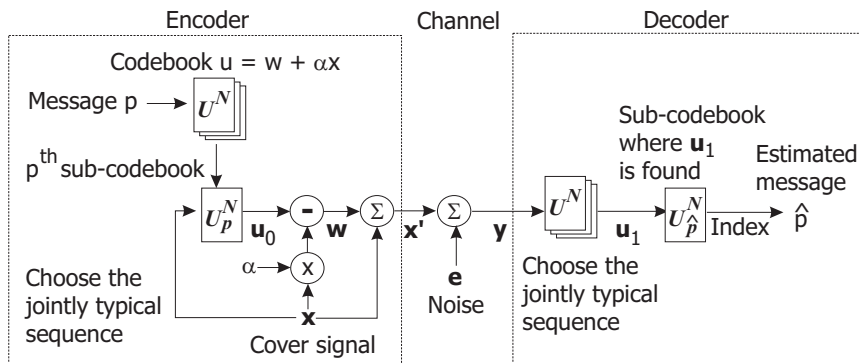


Figure 6.4: Costa's scheme for communication with a known Gaussian host in an AWGN channel. The operations of the encoder and the decoder are described in section 6.2.2.

6.2.3 Practical implementation of Costa's solution

Unfortunately, Costa's proposed solution cannot be implemented in practice. This is because the codebook size M can get very large. If the σ_e^2 is small, \mathcal{U} must provide an accurate description of any possible realisation of \mathbf{x} . It is not practical to store nor

search in such a large codebook. We describe two practical implementations of Costa's scheme in this section: *Quantisation Index Modulation* (QIM) due to Chen [29, 30] and *Scalar Costa Scheme* (SCS) due to Eggers [52, 54]. Both schemes operate in the spatial domain and are based on uniform quantisation of host samples, but with slightly different forms of codebook.

6.2.3.1 Quantisation Index Modulation

Quantisation Index Modulation (QIM) was not originally formulated as an implementation of Costa's ideas, but as a solution of an information hiding problem where one aims to maximise the information capacity subjected to some distortion constraints. The basis of QIM is an ensemble of uniform dithered quantisers. All quantisers in the ensemble have identical step size but different dithers. The size of the ensemble is equal to the number of possible values a watermark symbol can take. This is illustrated in figure 6.5 for the 1-D signal case with an ensemble of two quantisers for binary signalling. We assume scalar quantisers are used at the moment, and we will explain later how quantisation based watermarking is extended to embed one bit in multiple host samples. If a bit 0 is to be embedded, then the quantiser corresponding to bit 0 is chosen and the host sample is quantised to the nearest reconstruction point of that quantiser and similarly for a bit 1. The decoder quantises the received sample with each of the quantiser in turn, and the index of the quantiser which gives the smallest quantisation error is the estimated symbol. This is equivalent to searching for the nearest reconstruction point in the whole ensemble of quantisers. The encoding and decoding processes (assuming binary coding) can be expressed as follows:

$$\text{Encoding: } x' = Q_{\Delta}(x - d_i) + d_i \quad i \in \{0, 1\}. \quad (6.6)$$

$$\text{Decoding: } err_i = Q_{\Delta}(y - d_i) + d_i - y \quad i = 0, 1; \quad (6.7)$$

$$\text{and output } \begin{cases} \text{bit 0,} & \text{if } |err_0| < |err_1|, \\ \text{bit 1,} & \text{otherwise.} \end{cases} \quad (6.8)$$

Q_{Δ} is a scalar uniform quantiser with step size Δ as before and d_0, d_1 are the dithers for bit 0 and bit 1. d_0 is generated as a uniform distribution over $[0, \Delta)$ and d_1 is

given by:

$$d_1 = \begin{cases} d_0 - \frac{\Delta}{2}, & \text{if } d_0 > \frac{\Delta}{2}, \\ d_0 + \frac{\Delta}{2}, & \text{otherwise,} \end{cases} \quad (6.9)$$

such that d_0 and d_1 are as far apart as possible and the quantisation errors are independent of the host samples. Chen's QIM scheme can also be considered as a special case of Costa's solution with the random codebook replaced by the product of N scalar codebooks:

$$\mathcal{U}^1 = \{k\Delta + d_i \mid i \in \{0, 1\}, k \in \mathbb{Z}\}, \quad (6.10)$$

which is equivalent to $u = w + x$, i.e. $\alpha = 1$. However, $\alpha = 1$ is optimal (at least under AWGN) only if there is no additional noise ($\sigma_e^2 = 0$) [38]. The optimal α at which the capacity is maximised for a given noise level is given by $\sigma_w^2 / (\sigma_w^2 + \sigma_e^2)$ in the case of AWGN interference.

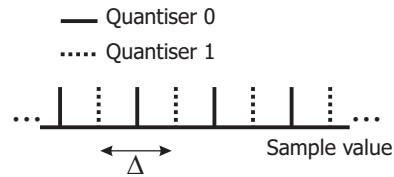


Figure 6.5: Quantisation Index Modulation (QIM) in 1-D. The cover signal samples are quantised individually using a uniform quantiser. If a bit 0 is to be embedded, the sample is quantised to the nearest solid line, otherwise it is quantised to the nearest dashed line. The quantiser step size Δ controls the maximum distortion introduced.

6.2.3.2 Scalar Costa Scheme

Eggers proposed a scalar version of Costa's solution by replacing the random codebook \mathcal{U}^N by the product of N scalar codebooks, each of the form (6.2). This is shown in figure 6.6. In the case of binary signalling, the scalar codebook is given by:

$$\mathcal{U}^1 = \left\{ u = k\alpha\Delta + i\frac{\alpha\Delta}{2} \mid i \in \{0, 1\}, k \in \mathbb{Z} \right\}. \quad (6.11)$$

Additionally a dither αd_i can be added to the term $i\frac{\alpha\Delta}{2}$, otherwise (6.11) is the same as (6.10) except for the appearance of α . The encoding process is similar to that of QIM. The process of searching for a typical sequence $(\mathbf{u}_0, \mathbf{x})$ is simplified to samplewise quantisation using the codebook \mathcal{U}^1/α , which is the codebook in (6.11)

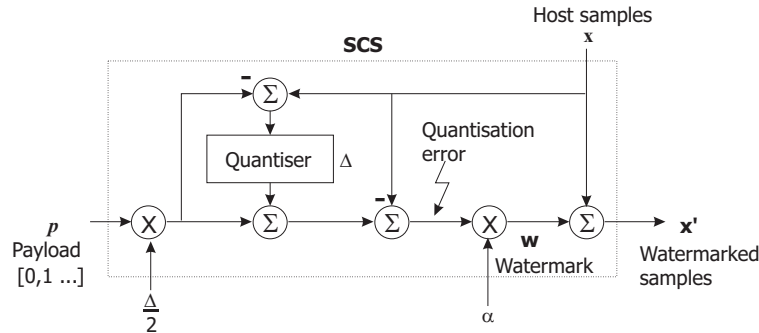


Figure 6.6: Scalar Costa Scheme (SCS) as proposed by Eggers [52, 54], which can be considered as a generalised version of Quantisation Index Modulation with an extra parameter α , which is fixed to 1 in the case of QIM. α can be optimised if the level of interference is known in advance. Using $\alpha < 1$ allows the enlargement of the quantiser step size while keeping the watermark power the same. SCS is a direct scalar implementation of the scheme proposed by Costa in [38].

with all entries scaled by $1/\alpha$. The transmitted signal is given by: $\mathbf{w} = \mathbf{u}_0 - \alpha \mathbf{x}$. The encoding process is summarised in the equation below:

$$x' = x + \alpha \left(Q_{\Delta} \left(x - \frac{i\Delta}{2} - d_i \right) + \frac{i\Delta}{2} + d_i - x \right) \quad i \in \{0, 1\}. \quad (6.12)$$

In other words, only a fraction ($\alpha < 1$) of the quantisation error is added back to the host signal to form the watermarked signal. In order to keep the watermark power the same as in the previous case, the quantiser step size has to be increased such that:

$$\Delta_{SCS} = \frac{\Delta_{QIM}}{\alpha}. \quad (6.13)$$

The decoding process is identical to that of QIM. The received samples are quantised with the same quantiser as the encoder, and the bin indices which the samples fall into give the estimated message. Eggers derived a slightly different value of α to that suggested by Costa by numerical maximisation of capacity in a Gaussian channel and his simulation results showed that the capacity of SCS is superior to that of QIM. Chen later patched QIM by introducing the same codebook as SCS and called it *Distortion Compensated QIM* (DC QIM) [31], but Chen just used the value of α suggested by Costa.

6.2.4 Extension to multiple samples quantisation

The two quantisation based watermarking schemes described earlier assume *one* watermark bit is embedded into *each* host sample. Typically the theoretical capacity

of the image is less than one bit per sample and one would use multiple samples for embedding one payload bit to improve the robustness of the watermark. One simple way to do this is to use the same quantiser repeatedly over each sample corresponding to a particular bit, which is equivalent to using a repetition code. Alternatively one can design a vector quantiser for each symbol. However, the codebook size increases exponentially as the number of host samples increases and we have the same problem as Costa's original scheme. There is yet a third way to embed one payload bit into multiple host samples, called spread transform coding, which will be described next.

6.3 Spread spectrum and quantisation watermarking

6.3.1 Comparing spread spectrum and quantisation based watermarking

In this thesis, we have described two completely different approaches to watermarking. One is based on the addition of a pseudorandom sequence to the host image, while the other quantises the host image samples so they fall into bins corresponding to the embedded data. The major difference between the two systems is the exploitation of the knowledge of the cover in quantisation based systems but not in spread spectrum based systems. The use of a visual mask in spread spectrum systems does not count as exploiting the cover because the watermark is *additive* and constructed *independently* with respect to the cover image. We compare and contrast the different features of spread spectrum and quantisation based watermarking in table 6.1. The advantages and the disadvantages of the two approaches complement each other. Fortunately, there is a way to combine spread spectrum and quantisation watermarking in order to get the advantages of both.

6.3.2 Spread transform watermarking

Quantisation based watermarking allows a relatively large amount of data to be hidden in the image, which is suitable for information hiding applications, where robustness is not a primary requirement. On the other hand, watermarking applications normally only require a much lower capacity than quantisation based methods can offer. Spread transform (ST) watermarking is a hybrid combination between spread spectrum and quantisation watermarking systems, where quantisation occurs in a

Feature	Spread spectrum	Quantisation
Exploiting the HVS	The PN sequence can be shaped by a mask computed from the cover image, making the watermark image adaptive.	Since a single scalar uniform quantiser is usually used for the whole image, it is difficult to adapt the watermark to the cover image.
Host interference at the watermark receiver	The embedder simply adds the watermark to the cover image, so it acts as interference at the receiver.	Since the encoder uses the host to construct the watermark, the interference from the host can be suppressed at the receiver.
Robustness to compression	Image adaptive watermarks can resist compression.	Non-image adaptive watermarks are less robust to image adaptive attacks such as compression.
Robustness to estimation based attacks	Since the watermark is additive with respect to the cover image, it can be estimated from the watermarked image and be removed.	The watermark is dependent on the cover image as it is the quantisation error of the quantiser in the embedder. Therefore it cannot be estimated from the watermarked image.
Capacity	The capacity is usually limited by the interference from the host.	The capacity is normally higher than in spread spectrum systems and is only limited by external interference on the watermarked image.
Embedding multiple watermarks	Multiple watermarks can be embedded by using orthogonal PN sequences.	Additional watermarks will overwrite existing ones unless non-overlapping subsets of pixels are used for the watermarks.

Table 6.1: Comparing various features of spread spectrum and quantisation based watermarks. The features of the two systems are complement of each other.

reduced dimension space. The robustness of a quantisation based watermark is improved at the expense of lowering the capacity of the image. Spread transform was briefly mentioned by Chen in [30] as a method for combining QIM and image adaptive spread spectrum watermarking systems. The host samples vector is projected onto a random vector² and the resulting projection is quantised to embed one bit, as described in the previous section. Finally, a multiple of the random vector is added to the host vector such that the projection of the watermarked samples vector onto the random vector will produce the correct projection value. The embedding process is illustrated in figure 6.7. The encoding and decoding processes are the same as in QIM (or SCS), except that x and x' in (6.6), (6.12), etc. are replaced by projection p of the image samples onto a key dependent random vector, which is defined as:

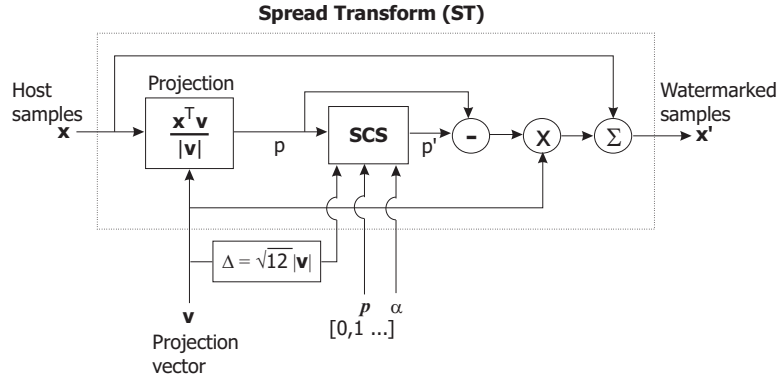


Figure 6.7: Spread Transform is a hybrid combination between spread spectrum and quantisation based watermarking techniques. The basic scheme is quantisation operating in a reduced dimensional space. Image samples are projected onto random vectors to generate the reduced dimensional data for embedding. The box marked ‘SCS’ is the Scalar Costa Scheme as depicted in figure 6.6.

$$p = \frac{\mathbf{x}^T \mathbf{v}}{|\mathbf{v}|}, \quad (6.14)$$

where \mathbf{x} is a column vector of image samples and \mathbf{v} is the random projection vector. The overall embedding process can therefore be summarised as:

$$\mathbf{x}' = \mathbf{x} + (p' - p)\mathbf{v}, \quad (6.15)$$

with,

$$p' = p + \alpha \left(Q_{\Delta} \left(p - \frac{i\Delta}{2} - d_i \right) + \frac{i\Delta}{2} + d_i - p \right) \quad i \in \{0, 1\}. \quad (6.16)$$

Various symbols have the same meaning as in (6.12).

By using the spread transform technique, one can combine the advantages of spread spectrum and quantisation watermarking systems. The host image no longer acts as interference at the decoder because the decoder computes the projection of the watermarked image onto the random vector instead of the correlation between the watermarked image and the random vector. The watermark can be made adaptive to the host image by scaling the random vector according to the host image. Additional watermarks can be embedded in the image by using orthogonal random vectors. In order to reduce the computational load, the host image is divided into blocks and random images the same size as one block are used as projection vectors. A block size

²The length of the projection vector depends on the number of bits to be embedded. The host vector is divided into blocks of the same length as the projection vector, and 1 bit can be embedded into each of the resulting projections.

of 32×32 is used as in our spread spectrum system. We also apply orthogonalisation *before* scaling the random images, because we found that it gives better performance than orthogonalisation *after* scaling, just like the spread spectrum system we proposed earlier in chapter 4 (see page 48).

6.4 Spread transform watermarking with complex wavelets

In this section, we describe a practical implementation of spread transform watermarking in the CWT domain.

6.4.1 Embedding algorithm

In the original algorithm described by Chen [30], only one projection vector is used for encoding each bit. Since the denominator of (6.14) is non-linear in \mathbf{v} , we cannot compute the projection *separately* in each subband and add the results as in the case of correlation based decoding (section 4.4.4). However, the high frequency components of the image become very noisy under compression and one should ideally assign less weight to the contributions from these components. This can be achieved in spread transform watermarking by using *multiple* projection vectors per information bit. If the projection vector is a bandpass signal, then projection onto the random vector is almost equivalent to bandpass filtering the host image, and only part of the image frequency spectrum will contribute to the projection. We define a frequency partitioned signal (or vector) as a signal formed by reconstruction using only some of the subbands of the frequency transform of a white signal. In our implementation, three levels of the CWT are used and we construct *three* frequency partitioned vectors from each random vector, each using only one level of CWT coefficients. Ideally one would use 18 partitioned vectors from one random vector, each reconstructed from one CWT subband, in order to obtain maximum channel adaptability. However, this greatly increases the computational costs both in embedding and decoding of the watermark. Besides, most attacks are not orientation specific, hence it is not justified to weight different orientation subbands in a given frequency range differently. Alternatively, one can also not use frequency partitioning at all and just shape the random sequence in the CWT domain, followed by inverse transform to form the projection vector. We will refer to this as CWT spread transform without frequency partitioning. The details of the embedding algorithm of our CWT spread transform

scheme (with frequency partitioning) are as follows:

1. The host image is divided into 32×32 blocks and the payload is divided into equal-sized portions, with each portion being embedded into a separate block.
2. A pseudo random vector of ± 1 (same size as one block) is generated for each bit of the payload to be embedded.
3. The forward CWT of this vector is computed, and scaled according to local image activity.
4. The scaled coefficients of this vector are frequency partitioned into 3 vectors, with each one reconstructed back into the spatial domain using only one level of CWT coefficients. From this point onwards, the watermarking process takes place in the spatial domain, the CWT domain is only used for adapting the random vector to the image.
5. Compute the projections of the host image samples onto each of these partitioned vectors:

$$p_i = \frac{\mathbf{x}^T \mathbf{v}_i}{|\mathbf{v}_i|} \quad i = 1, \dots, 3, \quad (6.17)$$

where \mathbf{v}_i is the vector in the spatial domain reconstructed using level i (scaled) CWT coefficients of the original random vector and p_i is the projection between the image samples \mathbf{x} and \mathbf{v}_i . If multiple bits are to be embedded, orthogonal random vectors are first generated before they are perceptually adapted to the image and partitioned.

6. Quantise the projections using (6.12), forming p'_i .
7. The watermarked image samples are given by:

$$\mathbf{x}' = \mathbf{x} + \sum_j \sum_{i=1}^3 (p'_{ij} - p_{ij}) \mathbf{v}_{ij}, \quad (6.18)$$

where j is the subscript for an individual bit in that block; \mathbf{v}_{ij} is \mathbf{v}_i for bit j and so on.

The spread transform watermark encoder is illustrated in figure 6.8. Examples of spread transform watermarks look similar to those of spread spectrum (figures 4.7b, 4.8b, 4.9b) and are not shown here.

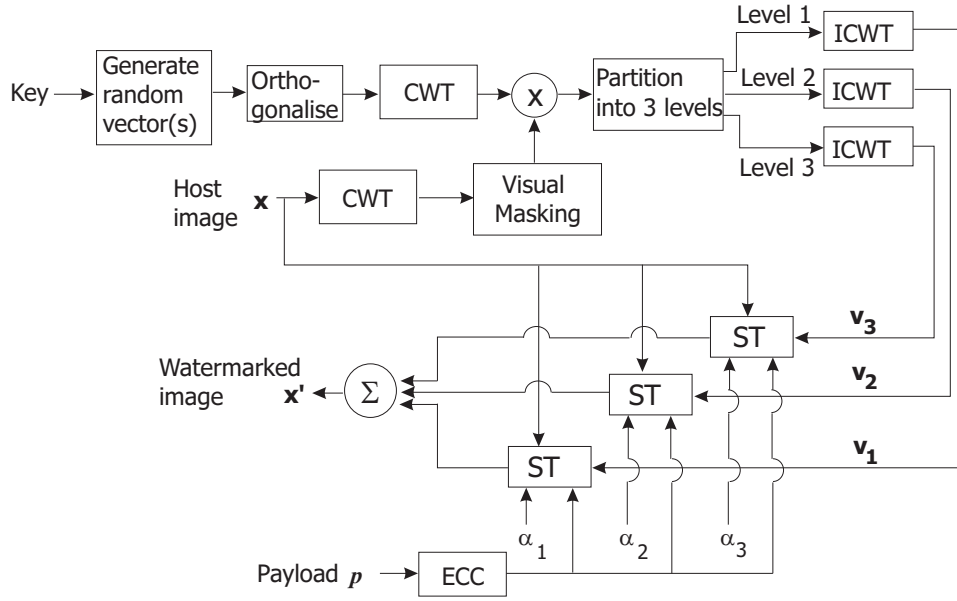


Figure 6.8: We use the CWT to adapt the random vectors to the host image prior to projection in order to make the spread transform watermark image adaptive. We can also partition the random vectors into different frequency components so that only some of the image frequency components contribute to a particular projection, but otherwise our proposed spread transform scheme is very similar to the basic spread transform algorithm. The boxes marked ‘ST’ correspond to figure 6.7.

6.4.2 Decoding algorithm

The decoding algorithm is as follows:

1. The received image is divided into 32×32 blocks.
2. The same random vectors as used in the encoder are generated and forward transformed into the CWT domain.
3. The CWT coefficients of the random vectors are scaled using a visual mask estimated from the received image.
4. The scaled CWT coefficients are frequency partitioned into 3 vectors as in the encoder.
5. Compute the projection of the watermarked image onto each of these vectors using (6.14). Thus we have 3 projections for each information bit.
6. Each of the three projections is converted into a soft output between -1 and 1 as follows:

$$output_{soft,i} = 1 - \frac{4}{\Delta} |Q_{\Delta}(p_i - \Delta d) + \Delta d - p_i|, \quad (6.19)$$

where $output_{soft,i}$ is the soft output corresponding to projection p_i ; Q_Δ is the uniform quantiser of step size Δ used for quantising the projections and d is the optional dither.

7. The three soft outputs are weighted and combined to give the final soft output for a particular payload bit.

$$output_{soft} = \sum_{i=1}^3 \nu_i \cdot output_{soft,i}. \quad (6.20)$$

If an error control code is used, the final soft output is passed to the ECC decoder, otherwise a bit 0 is returned if $output_{soft}$ is positive and bit 1 otherwise. How to choose the weights ν_i is discussed later.

The aforementioned embedding and decoding algorithms also apply to the non-frequency-partitioning version of our implementation, except in this case the frequency partitioning part is omitted and we have only one projection per data bit.

6.4.3 Parameters selection

6.4.3.1 Choosing α and Δ for the quantiser

In [38], Costa suggests an optimal value of α of $\sigma_w^2 / (\sigma_w^2 + \sigma_e^2)$, whereas Eggers in [52, 54] suggests a slightly different value of α , but both are based on maximising the capacity of an AWGN channel. In the case of spread transform watermarking, the noise depends on the projection of the distortion onto the random vectors and may not be Gaussian. The appropriate values of α for both the frequency-partitioning and non-frequency-partitioning version of our scheme were determined empirically using JPEG compression as the attack. Watermarks are embedded using different values of α and the bit error rates at different compression levels are measured. The value of α which gives the lowest BER on average is used in the quantiser for that level. In the version of our algorithm with frequency-partitioning, only one level of CWT coefficients is used each time so we can obtain the best value of alpha for each level. Figures 6.9 (a-d) show the effects of α for both the frequency-partitioning and the non-frequency-partitioning version of the algorithm. Unlike the AWGN scenario considered by Costa and Eggers (figure 6.9f), where the optimal value of α depends on the noise level, the optimal value of α does not depend on the severity of compression. In particular, using $\alpha < 1$ gives us no advantage over simple QIM for level 1 of the CWT as well as non-frequency-partitioning spread transform (figure 6.9d). On the other hand, the

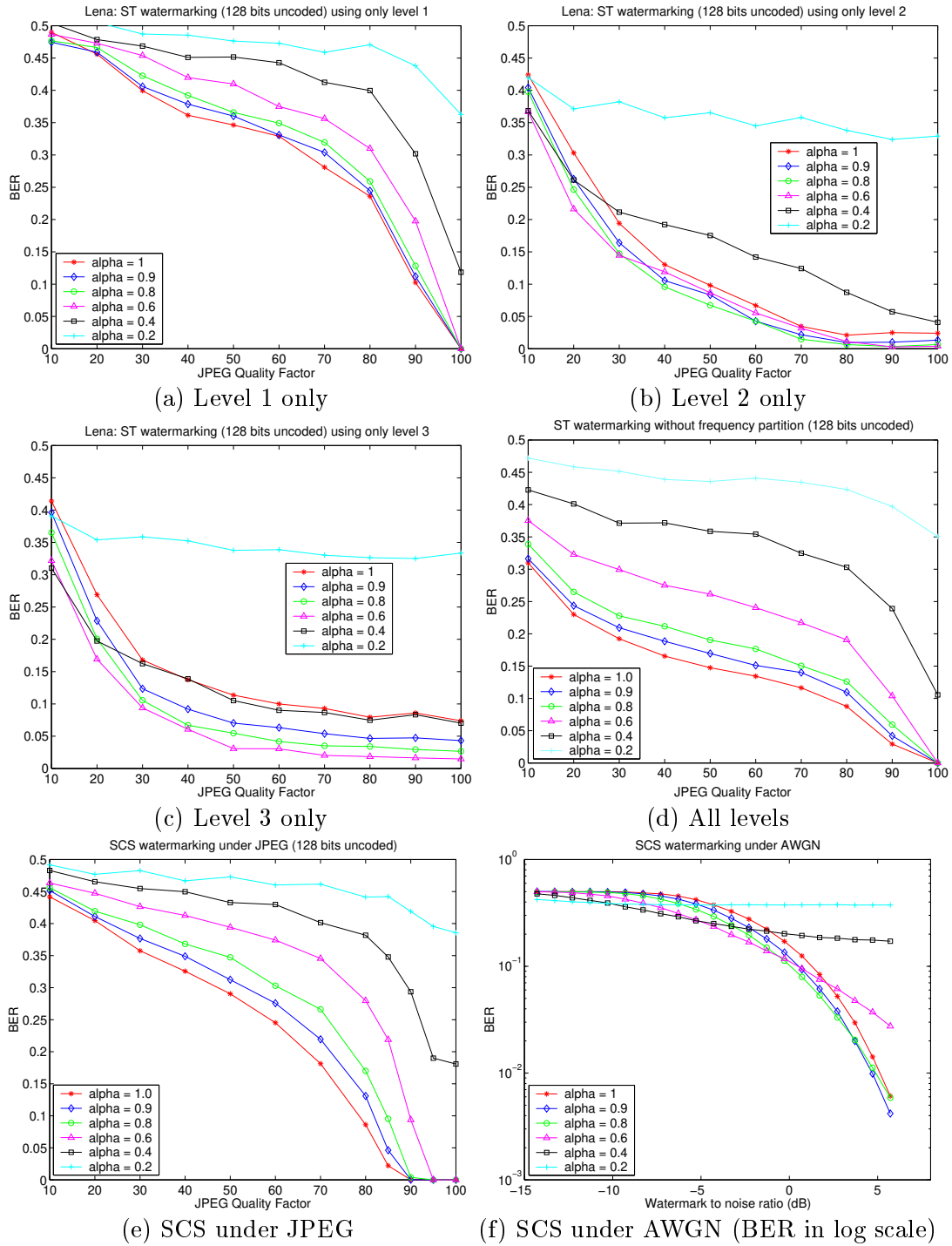


Figure 6.9: Effects of α (see equation 6.12) on spread transform watermark decoding performance. (a) to (c) show the variation of BER with α under JPEG compression when only 1 level of CWT coefficients are used in the embedding algorithm. (d) shows the same result when all the levels are used together without partitioning them into difference sequences. (e) and (f) correspond to the results using the original SCS algorithm proposed in [52, 54], which is *non-image-adaptive*, under JPEG and AWGN. In all cases, the spreading vector is of length 1024 and a 128-bit watermark is embedded. The watermark to noise ratio in (f) is defined as $10 \log(\sigma_w/\sigma_n)$ where the noise variance has taken into account the effect of the spreading vector. Please refer to section 6.4.3.1 for discussions of the results.

optimal value of α for level 2 and 3 of the CWT is around 0.8 and 0.6 respectively, regardless of the level of compression. The observed results can be partly explained as follows. Both Eggers and Costa assume the interference is *independent* with respect to the watermark, which is only true under a low level of compression (see section 5.2.1.1). Using $\alpha < 1$ moves the quantised projections further away from the bin centres and introduces additional interference. The distortion due to compression is less correlated with the host signal for lower frequency components and using $\alpha < 1$ for these components gives us a slight advantage over simple QIM.

The variance of the watermark is given by $\Delta^2/12$, assuming the projection p has a uniform distribution over a quantisation bin. Thus we choose Δ to be $|\mathbf{v}|\sqrt{12}/\alpha$, so that the variance of the watermark will be the same as in a spread spectrum system. Since the quantiser step size is watermark dependent, the decoder has to estimate the Δ used in the encoder. This does not cause a problem because the decoder can estimate the visual mask quite accurately using the watermarked image, provided the watermarked image is not too severely distorted.

6.4.3.2 Assigning channel weights ν_i

In section 4.4.4, we described the use of a reference watermark for estimating channel statistics for combining multiple correlator outputs. Unfortunately, the reference watermark consumes some of the watermark energy. In section 5.2.1.1, we proposed the use of fixed weights, when a pessimistic level of attack is assumed, which allows us to allocate all the watermark energy to the payload. We use the latter strategy in our spread transform scheme. This is because any error correction code will be able to correct errors up to a certain level, so it does not matter if the weights are not optimised under mild attacks. We assume a worst case compression level equivalent to JPEG compression at quality factor of 50 and choose the weights using the binary symmetric model suggested by Kundur in [89] and the measured BER at JPEG quality factor 50 in the 3 levels of CWT. Readers are referred to [89] for details of deriving channel weights from measured bit error rate. The weights are calculated to be 0.1, 0.4, 0.5 for level 1, 2 and 3. We also discovered that it does not matter if level 1 is not used at all because it is assigned such a small weight.

6.4.4 Performance of spread transform watermarking

There are three sources of error at the watermark decoder:

1. The projection of the distortion onto the projection vector.

2. Error in the projection vector due to the slightly different visual mask used at the decoder compared with the encoder.
3. Error in the estimation of quantisation step size.

As discussed in the last section, the quantisation step size Δ depends on the magnitude of the projection vector. Under reasonable levels of attack, the visual mask estimated by the decoder will not be too different from the mask used in the encoder. Hence the magnitude of the projection vectors in the encoder and the decoder will be approximately the same and the last source of error is typically negligible compared with the other two. If we denote the received image vector as $\mathbf{x}' + \delta\mathbf{x}$ and the estimated projection vector as $\mathbf{v} + \delta\mathbf{v}$, then the error in projection is:

$$\begin{aligned}
 p_{receive} &= \frac{(\mathbf{x}' + \delta\mathbf{x})^T (\mathbf{v} + \delta\mathbf{v})}{|\mathbf{v} + \delta\mathbf{v}|}, \\
 &\approx \frac{(\mathbf{x}' + \delta\mathbf{x})^T (\mathbf{v} + \delta\mathbf{v})}{|\mathbf{v}|}, \\
 &= p' + \frac{\delta\mathbf{x}^T \mathbf{v}}{|\mathbf{v}|} + \frac{\mathbf{x}'^T \delta\mathbf{v}}{|\mathbf{v}|} + \frac{\delta\mathbf{x}^T \delta\mathbf{v}}{|\mathbf{v}|}.
 \end{aligned} \tag{6.21}$$

Let the second, the third and the last term of (6.21) be p_{err1} , p_{err2} and p_{err3} respectively. We observed that p_{err1} and p_{err2} have the same order of magnitude for level 2 and level 3 of the CWT under low to medium level of compression, whereas p_{err1} dominates the error in the finest level under medium level of compression. p_{err3} is in general small compared with the other two errors. The distributions of p_{err1} and p_{err2} are observed to be more uniform than the Gaussian distribution. Nevertheless, assuming them to be Gaussian gives us the worst case scenario and hence an upper bound on error. Let the total variance due to p_{err1} and p_{err2} be σ_n^2 . Projecting the image vector onto a random vector will reduce the power of the distortion by a factor of N , the length of the projection vector, on average. Using the fact that the quantisation step size is related to the watermark power by $\Delta = \sigma_w \sqrt{12}$ and that a decoding error will occur if the projection falls outside the quantisation bin boundary, the bit error rate is bounded by:

$$P_{err} \approx \leq 2Q \left(\sqrt{\frac{3N}{4}} \frac{\sigma_w}{\sigma_n} \right), \tag{6.22}$$

where Q is the Gaussian integral (4.15) as before. Note that the region in which decoding error will occur is periodic and so (6.22) is not exact even if the distribution

of the distortion is Gaussian. The BER of an equivalent spread spectrum based scheme is approximately $Q(\sqrt{N}\frac{\sigma_w}{\sigma_x})$. However, the interference associated with the spread spectrum decoder σ_x is typically larger than σ_n because the host image is part of the interference for a spread spectrum scheme. Therefore we expect spread transform to perform better than spread spectrum under mild to medium attacks. When the attack distortion dominates the interference ($\sigma_x \ll \sigma_n$), however, a spread spectrum scheme should perform better because there is only one decision boundary in a spread spectrum scheme, instead of two in spread transform.

Watermark detection in quantisation based (including spread transform) schemes is different from the spread spectrum case discussed in section 4.6. Since we are using a quantiser for decoding, we cannot correlate the image samples with the known sequence to detect a watermark. Fortunately, we can detect a watermark using error control codes, which will be discussed shortly.

6.5 Comparing CWT spread transform watermarks with other quantisation based schemes

In this section we compare the performance of our proposed spread transform scheme with an existing quantisation based watermarking scheme in the wavelet domain. The scheme proposed by Kundur in [89] was chosen to compare with our proposed scheme. In [89], the median of wavelet coefficients at the same location in different subbands is quantised into the appropriate bin to encode one bit of data. The watermark payload is embedded repeatedly for each wavelet level and a reference watermark the same length as the payload is embedded alongside each copy of the payload. The decoder estimates each copy of the payload and weights them according to the BER of the corresponding reference mark. The weighted copies are added together to give the overall estimated watermark.

Figures 6.10 (a and b) show the BER of the frequency-partitioning version of our scheme, the scheme proposed by Kundur in [89], as well as direct extension of SCS using spread transform. One can see that our algorithm has superior performance under medium to severe compression scenario, whereas the scheme in [89] performs better at low compression scenario. However, as argued in chapter 4, when an error control code is used, only the point at which the BER reaches a critical value (usually between 0.1 to 0.2) matters. Therefore we expect our system can tolerate a more severe level of compression than the one in [89]. Other results for Baboon and

Pills images under JPEG and JPEG2000 are similar to figure 6.10 and are thus not shown here. Direct extension of Scalar Costa Scheme (section 6.2.3.2) using spread transform, which is not image adaptive, has poor performance under compression in general. The results highlight the importance of adapting the random vectors to the cover image, if the watermark is to be resistant to compression.

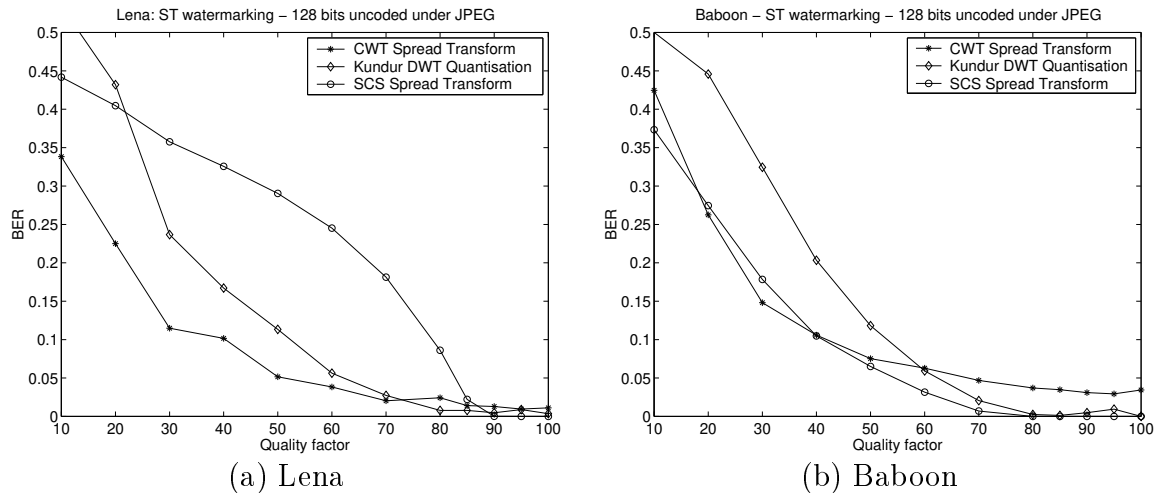


Figure 6.10: Comparing the frequency partitioning version of our proposed spread transform scheme in the CWT domain with Kundur's scheme [89] and spread transform SCS under JPEG compression. The scheme in [89] quantises the median of a tuple of DWT coefficients to embed a bit. In all cases a 128-bit uncoded watermark is embedded with the same RMS (2.8 for Lena and 3.4 for Baboon). Although our proposed scheme has poorer performance than Kundur's scheme at high quality factors (more noticeable in (b)), the BER reaches the critical level (to be correctable by a typical error control code) at more severe compression level. Therefore our scheme will perform better in a practical situation. Simple spread transform SCS watermarking performs worst in Lena and has about the same compression tolerance to our scheme in Baboon.

6.6 The role of error control codes in watermarking

A practical robust watermarking system would employ an error control code to reduce the output error rate at the decoder. As noted in chapter 2, error control codes have the property that, when the SNR is lower than a threshold (or the input BER to the decoder is higher than a certain value), the ECC decoder fails and the output bitstream appears to be random. The rate of degradation depends on the particular code and the nature of the error. We can use this property to detect the presence of a watermark. The detection procedure is as follows:

1. Assume the received image is watermarked, and extract the *estimated* watermark.
2. Pass the extracted watermark to the ECC decoder to obtain the most probable payload.
3. Use the ECC *encoder* to encode the estimated payload. Let us call the resulting data the *feedback* watermark.
4. Compare the *feedback* watermark with the *estimated* watermark, and if the difference between them is *smaller* than the input BER threshold P_e , declare a watermark is detected, otherwise declare no watermark is present. The effects of P_e on the probabilities of false positive and detection and how to choose it are discussed next.

The above procedure can be explained by considering an n -*sphere*, where n is the length of the codeword. This n -*sphere* represents all possible bit patterns of length n . The codebook of an error control code forms a packing of k -*spheres* inside the parent n -*sphere*, where k is the length of the payload (see figure 6.11). Any n -bit patterns within the region of a k -*sphere* will be decoded to a k -bit payload corresponding to the centre of that k -*sphere*. This corresponds to the input BER being within the error correcting capability of that error control code (the radius of a k -*sphere*). An n -bit pattern extracted from an unwatermarked image will most probably fall into the *gaps* between the k -*spheres*. The ECC decoder will decode a k -bit pattern whose k -*sphere* is closest to the n -bit pattern. However, the distance between the extracted n -bit pattern and the centre of that k -*sphere*, which is the codeword after encoding the decoded k -bit pattern, will most probably be greater than the radius of the k -*sphere*. Thus we know the received n -bit pattern is probably not a codeword of the ECC and the target image is probably not watermarked.

We shall now calculate the probabilities of false detection of a non-existent watermark and of failure to detect an existing watermark. Given a threshold BER of P_e , a false positive will occur if a random n -bit sequence has at least $(1 - P_e)n$ bits (assumed to be integer for the moment) in agreement with one of the codewords. The probability that exactly i of the n bits of a random sequence agree with some pattern

is:

$$\begin{aligned} P(i \text{ out of } n) &= \binom{n}{i} \frac{1}{2^i} \frac{1}{2^{n-i}}, \\ &= \binom{n}{i} \frac{1}{2^n}. \end{aligned} \quad (6.23)$$

where $\binom{n}{i}$ is the number of possible ways of choosing i out of n and is given by $\frac{n!}{i!(n-i)!}$. The probability of false positive given a particular codeword is therefore:

$$P(\text{FP} \mid \text{some codeword}) = \sum_{i=(1-P_e)n}^n \binom{n}{i} \frac{1}{2^n}. \quad (6.24)$$

Since the codeword closest to the extracted n -bit pattern is always chosen and the k -spheres do not overlap, the unconditional false positive probability is just (6.24) multiplied by the number of codewords.

$$\begin{aligned} P_{FP,ECC} &= \sum_{\text{all codewords}} P(\text{FP} \mid \text{some codeword}), \\ &= \sum_{i=(1-P_e)n}^n \binom{n}{i} \frac{1}{2^n} \cdot 2^k, \\ &= \sum_{i=(1-P_e)n}^n \binom{n}{i} \frac{1}{2^{n-k}}. \end{aligned} \quad (6.25)$$

On the other hand, if at least $P_e n$ bits of the extracted watermark codeword are wrong, given a watermark is present, then we will miss the watermark. The probability that exactly i bits are wrong is $\binom{n}{i} P_b^i (1 - P_b)^{n-i}$, where P_b is the probability any given bit is decoded wrongly. Assuming all codewords are equally likely, the conditional and unconditional miss probability will be the same and are given by:

$$P_{Miss,ECC} = \sum_{i=P_e n}^n \binom{n}{i} P_b^i (1 - P_b)^{n-i}. \quad (6.26)$$

The probability of detection is therefore:

$$\begin{aligned}
 P_{Detect,ECC} &= 1 - P_{Miss,ECC}, \\
 &= \sum_{i=0}^{P_e n - 1} \binom{n}{i} P_b^i (1 - P_b)^{n-i}.
 \end{aligned}
 \tag{6.27}$$

When the length of the codeword n is fixed, the false positive probability depends on k and P_e , whereas the detection probability depends on the input bit error rate P_b and P_e . The user first chooses the desired maximum allowable output BER of the decoder, the corresponding input BER gives an upper limit of P_e . The user then chooses P_e as a trade off between probabilities of false positive and detection. Figure 6.12a shows the variation of probability of detection of a (384,124) code with input SNR for a few values of P_e , chosen such that $P_e n$ is an integer. The variation of false positive probability with P_e (at values where $P_e n$ is an integer) is shown in figure 6.12b. If we desire a maximum allowable output BER of say 10^{-4} when using the (384,124) turbo code described in section 2.4.2, this requires P_e to be smaller than 0.15 and the false positive probability will be at most 10^{-9} . The curves in figure 6.12a will be steeper, for a given P_e , if n increases while keeping the code rate fixed. The false positive probability will also decrease, because $\binom{n}{i}$ increases slower than 2^{n-k} .

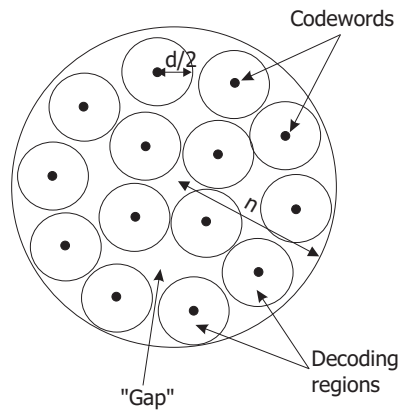


Figure 6.11: An (n, k) error control code can be considered as a packing of k -sphere inside an n -sphere. The radius of each sphere is roughly half the minimum Hamming distance d of the code.

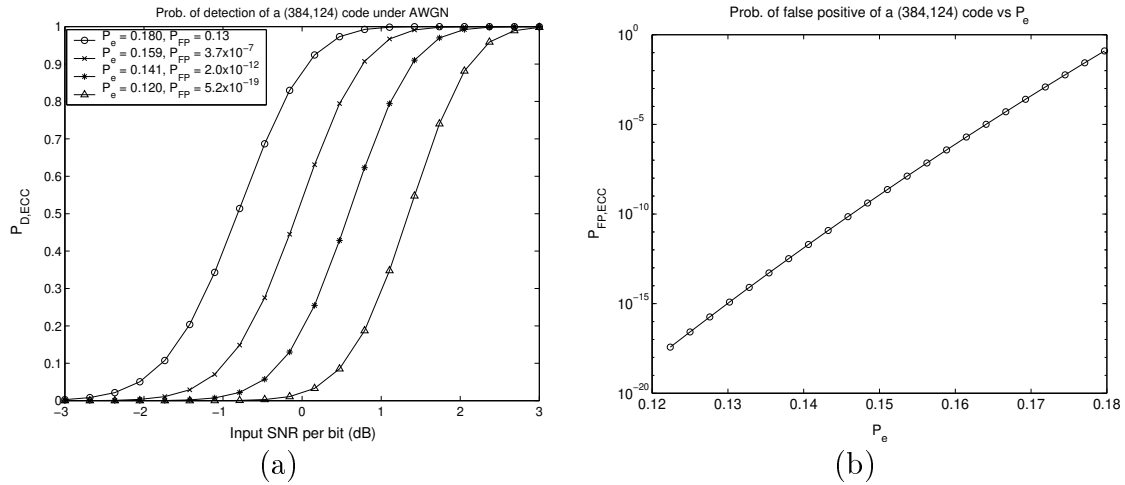


Figure 6.12: (a) shows the detection probability of a (384,124) code under different SNR for a few values of P_e . Assuming the interference is AWGN, P_b is given by $Q(\sqrt{SNR})$. (b) shows the variation of false positive probability with P_e of the same code.

6.7 Comparing spread spectrum and spread transform watermarks

In this section, we compare the performance of spread spectrum (SS) and spread transform (ST) watermarks. We choose JPEG as our attack scenario and the turbo code as our error control code. As discussed in section 2.4.2, the decoding complexity of turbo codes varies exponentially with the constraint length of the constituent encoders and linearly with the reciprocal of the code rate. We use a turbo code with 2 constraint length 5 encoders (and hence a rate of approximately $\frac{1}{3}$), with 10 iterations of decoding as a compromise between computational complexity and performance. The objective here is to compare the robustness of spread spectrum and spread transform watermarks at different payload lengths, where robustness is defined as the maximum level of compression the watermark can tolerate before the system fails, when say the output error rate is $> 10^{-4}$. Figures 6.13 (a and b) show the input BER to the turbo decoder and the output BER of a 124-bit watermark coded with a (384,124) turbo code under different quality factors for the frequency-partitioned version of spread transform scheme proposed earlier as well as the spread spectrum system described in chapter 4. The Lena image was used in all the experiments in this section. Figures 6.13 (c and d) show the probability of detection of the watermark based on error control codes as described in the previous section. The output BER drops below 10^{-4} when the input BER is less than about 0.13. Since the false alarm rate is less than 10^{-15} at this value of P_e , which is good enough for

most applications, we can just set P_e to 0.13 to maximise the detection probability. The threshold BER is lower than that of AWGN (figure 2.4) because the interference due to JPEG compression is not Gaussian. The point where the probability of detection changes sharply from 0 to 1 corresponds to the chosen breaking off point of the decoder, which justifies our approach of using error control codes for watermark detection. Figure 6.14a shows the lowest tolerant JPEG quality factor for our spread spectrum (SS CWT), frequency-partitioned spread transform (ST CWT) as well as straightforward spread transform extension of the Scalar Costa Scheme (ST SCS), which is *non-image-adaptive*, at different capacities. When the lowest tolerant JPEG quality factor reaches 100, it means the system cannot tolerate compression at all. The overall RMS watermark is the same in all three systems. The spread transform system with frequency partitioning is the most robust at low capacities but quickly deteriorates when the length of the payload exceeds 100 bits. Spread spectrum is the best when the payload is between 100 to about 300 bits, whereas at higher capacities (> 300 bits), non-image-adaptive spread transform is more robust. These results can be explained as follows. When a watermark is perceptually adapted to a cover image, it is not possible to generate exactly the same random sequences at the encoder and the decoder, because the visual mask estimated from the received image will *not* be the same as that calculated from the original cover image. In addition, when multiple random sequences are superimposed on top of each other, there will be interference between them, unless they are orthogonal. However, we mentioned in chapter 4 (page 48) that due to the slight inaccuracy in the visual mask at the decoder, if orthogonalisation is applied to the sequences *after* perceptual scaling, this inaccuracy results in errors in the projection vector which propagates to other vectors and leads to degradation in performance of the system. Therefore we decide to orthogonalise the sequences *prior* to scaling and so there remains a little bit of interference between the sequences. This interference builds up and becomes more significant as the length of the payload increases and eventually limits the capacity of the watermark system. In the case of the frequency-partitioned spread transform system, there is another source of interference. The frequency support of the wavelets for different levels in the CWT overlap and so there is interference between the different sequences constructed from the CWT coefficients from different levels. This is why the performance of spread transform with frequency partitioning deteriorates quicker than spread spectrum. On the other hand, quantisation based encoding methods allow us to extract the low frequency components of the watermark more reliably. Therefore frequency-partitioned spread transform is more robust at low capacities, when compression is the dominant

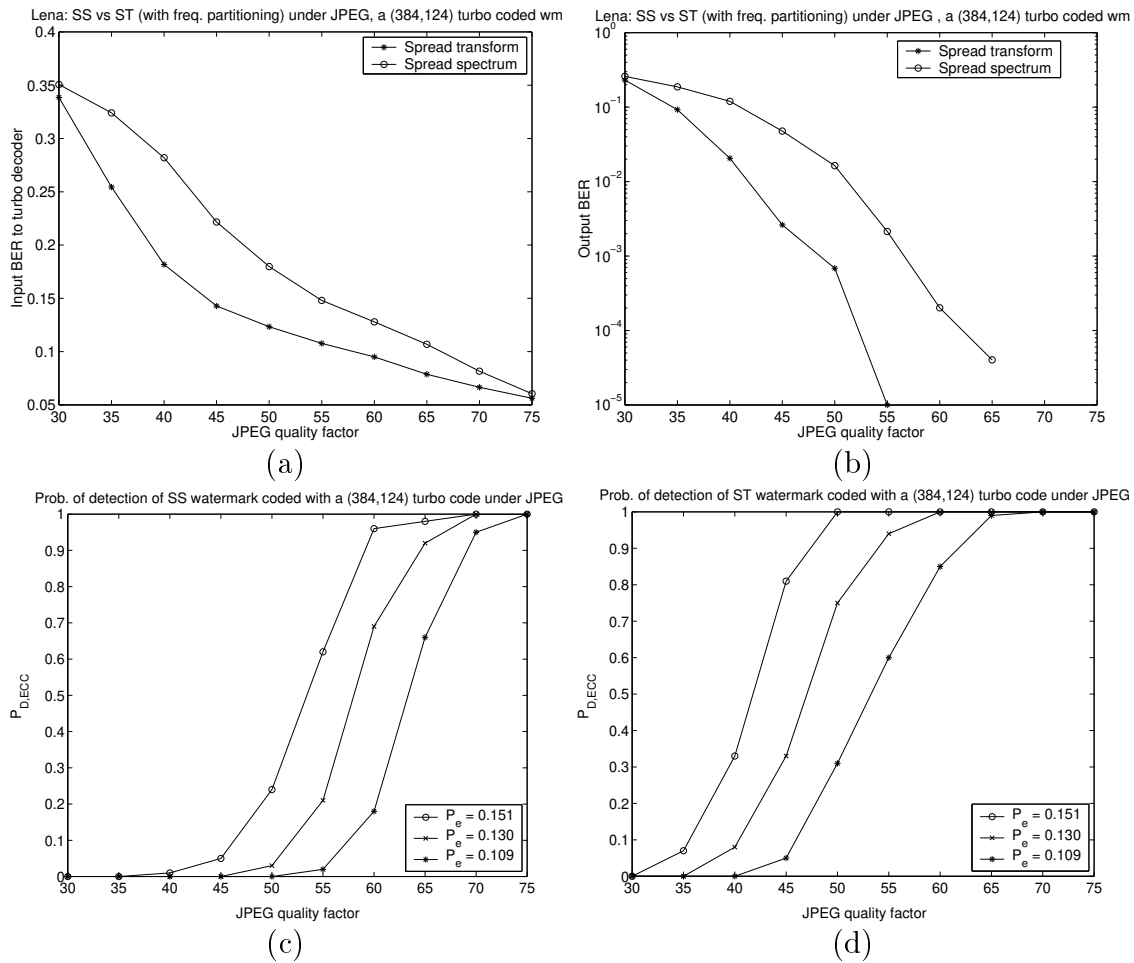


Figure 6.13: Comparison between our spread spectrum watermarking system described in chapter 4 and the spread transform system with frequency partitioning proposed in this chapter. A (384,124) turbo code is used to code the watermark. (a) shows the input BER to the turbo decoder. (b) shows the output BER. (c) shows the corresponding probability of detection for the spread spectrum watermark at different values of P_e . (d) shows the probability of detection for the spread transform watermark.

interference. Non-image-adaptive spread transform suffers from neither the interference due to the host nor the interference between different sequences and so allows a high capacity payload to be embedded, but since the watermark is not image adaptive, it is not very robust against compression. In addition, such watermarks will be more perceptible in smooth regions of the image.

An alternative to the current implementation of the frequency partitioning version of our spread transform is to only use 1 level of CWT coefficients, instead of using all 3 levels *separately*. Although the watermark energy will be greatly reduced as a result, we do not have the interference due to the vectors from different bands interfering with each other. Figure 6.14b shows the alternative implementation of spread transform

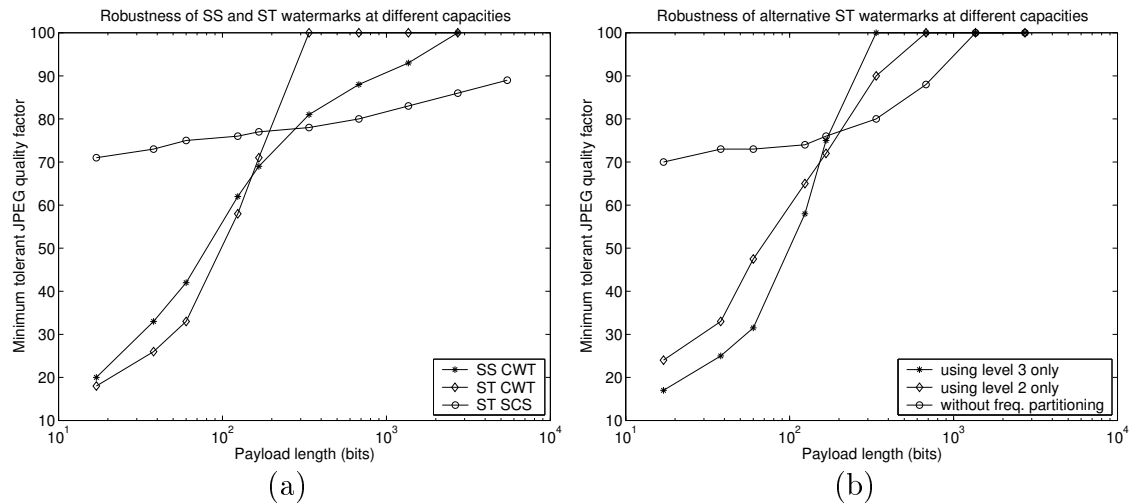


Figure 6.14: (a) shows the lowest tolerant JPEG quality factor (when the output BER is $< 10^{-4}$) for both spread spectrum, spread transform with frequency partitioning and non-image-adaptive spread transform. (b) shows the same thing when only *one* level of CWT coefficients are used in spread transform. The robustness of the non-frequency-partitioned version of our spread transform algorithm is also shown. In general, image adaptive spread transform is the best at low capacities; spread spectrum (also image adaptive) is the best at medium capacities and non-image-adaptive spread transform is the best at high capacities. When the lowest tolerant JPEG quality factor reaches 100, it means that the system cannot tolerate compression at all.

with frequency partitioning using only level 2 or level 3. If only level 3 is used, one can achieve slightly better performance than using all three levels at very low capacities. However, since the watermark energy is very small, the robustness of the spread transform system using only level 2 or 3 coefficients is quickly limited by the energy of the watermark. The results of non-frequency-partitioned spread transform under JPEG are also shown and we can see that it performs poorly at low capacities compared with SS CWT and ST CWT and has comparable performance to SS CWT between 100 to 300 bits. Non-frequency-partitioned spread transform treats the high and low frequency components of the watermark with equal weights, so its performance is limited by compression at low capacities. In a nutshell, one should use spread transform with frequency partitioning at low capacities (up to 100 bits) and use spread spectrum at medium capacities (100 to 300 bits) and use non-frequency-adaptive spread transform when higher capacities are required.

6.8 Chapter summary

In this chapter, we introduced the concept of informed watermarking, where the knowledge of the cover image is exploited at the encoder in order to reduce the interference due to the cover image at the decoder. Two practical implementations of quantisation based watermarking were described. The pros and cons of spread spectrum and quantisation based watermarking were compared and contrasted, and spread transform watermarking was introduced to combine the advantages of both spread spectrum and quantisation based watermarking. We proposed a spread transform watermark algorithm in the CWT domain which uses a frequency partitioning technique to increase the watermark's robustness to compression. Simulation results showed that spread transform with frequency partitioning is better than spread spectrum at low payload capacities, and spread spectrum is better at medium capacities, because the interference between the partitioned vectors in a spread transform system then becomes significant. Simple spread transform watermarking in the spatial domain *without* image adaptation is not very robust to compression, but it allows a large payload to be embedded. A watermark detection algorithm based on error control codes was also introduced, which removes the need for a reference watermark, and allows one to put all the energy into the payload to increase its robustness.

Chapter 7

Conclusions and Future Research

7.1 Thesis review

The work described in this thesis is concerned with the design of robust watermarking algorithms using complex wavelets. Various applications of watermarks were introduced and an overview of existing watermarking techniques and attacks was given. Complex wavelets were introduced as an alternative to conventional real wavelets and they overcome the two drawbacks of real wavelets: lack of shift invariance and directional selectivity. We demonstrated that the complex wavelets transform (CWT) closely resembles the human visual system and a visual model in the CWT domain was developed and shown to allow better image adaptation than real wavelets or the Discrete Cosine Transform (DCT).

The first watermarking algorithm developed was based on spread spectrum, just like many of the existing algorithms. However, since the CWT is a redundant transform, it is necessary to generate the random sequences in the spatial domain, otherwise significant amount of energy of the watermark will be lost upon inverse transform. A decoder based on a modified form of the matched filter was proposed and shown to achieve better performance than the ordinary matched filter when the underlying signal is non-stationary, which is typical of wavelet coefficients where most of the energy of the signal tends to be concentrated in a few large coefficients. We also compared our generic algorithm in three domains: the CWT, the DWT and the DCT. Experimental results showed that the CWT watermarks are the most robust under compression, additive noise, median and mean filtering. The issue of watermark detection was also addressed and we argued that using a separate reference watermark is better than fixing part of the payload to some known pattern, if no error control codes are used. However, in a practical system, one would encode the payload with

an error control code prior to embedding and we proposed a detection mechanism using the properties of error control codes. This removes the need for a reference watermark and allows one to put all the watermark energy into the payload and increase the robustness of the watermark.

We then discussed how the performance of the watermark decoder can be improved when the watermarked image is attacked. Three scenarios: compression, geometric distortion and denoising were considered. A new decoder based on the generalised Gaussian distribution was introduced but it was shown to perform no better than our modified matched filter under compression, hence we can just use the modified matched filter for decoding under compression. An image registration algorithm based on motion estimation in the CWT domain was developed which allows accurate image registration to combat geometric distortion attacks. Unfortunately, a reference image is still required for the registration process. Denoising attacks have similar effects to compression but are less effective. On the other hand, remodulation attacks proved to be effective against our algorithm, but they resulted in more distorted images. However, better visual models should make remodulation attacks more effective.

Although spread spectrum based watermarking systems are robust, the interference due to the cover image limits the capacity. The knowledge of the host can be exploited at the encoder, which can reduce the host interference at the decoder. This results in a new class of embedding algorithms based on quantisation. The pros and cons of the spread spectrum and quantisation based systems complement each other and spread transform was introduced as a hybrid combination of the two in order to harness the advantages of the two systems. A frequency partitioning technique was developed which increases the robustness of the basic spread transform system under compression. The robustness of our spread transform and spread spectrum algorithms were compared under compression. Experimental results showed that one should use image adaptive spread transform watermark if the payload length is relatively short, and resort to spread spectrum when the payload length is medium and only use the non-image-adaptive form of the spread transform algorithm when a high data rate is required.

7.2 Future research directions

In this section, we summarise a few unresolved issues and give some possible research directions.

7.2.1 Blind image registration

An image registration algorithm based on motion estimation was developed in chapter 5, but it requires a reference image. The problem with many template based blind registration schemes is that they assume the transformation is globally affine. Geometric distortion, though not globally affine, can be well approximated by affine transform on a blockwise basis. Thus a block-by-block template based registration scheme may allow blind registration against geometric distortion. At the time of writing, *Voloshynovskiy et al.* [48, 150] propose just such a scheme to combat geometric distortion.

7.2.2 Remodulation attack and second generation watermarks

The remodulation attack, which is a removal based attack similar to denoising, but with part of the estimated noise added back to the denoised image, is shown to be effective against additive watermarks which use correlation based decoders. Unfortunately, even our spread transform based scheme is not very robust against it. A possible counter measure against the remodulation attack is to embed the watermark in a non-additive way¹. An example is proposed in [94], where the positions of feature points (for examples, corners) are modified to embed a watermark. The authors refer to this as *second generation watermarks*. The available capacity for embedding watermarks in this manner is rather low and much work is still required to make the scheme more practical. Another research area related to second generation watermarks is known as object watermarking, where an image is segmented into different regions and only the regions of interest are watermarked.

7.2.3 Partially and iteratively image adaptive watermarking

Watermarks should be perceptually similar to the cover image if they are to be resistant against image adaptive attacks (such as compression and other estimation based attacks). However, one cannot generate identical watermark sequences in the embedder and the decoder in these systems because the visual mask estimated from the candidate image will always be slightly different from that used in the embedder. This results in a small interference between the random sequences corresponding to

¹Both spread spectrum and quantisation based watermarks are additive. Spread spectrum systems construct watermarks independently of the host image, whereas quantisation based watermarks are image dependent.

different watermark symbols, which builds up as the length of the payload increases. At the other extreme, non-image-adaptive watermarks do not suffer from this interference, but these systems are not very robust under compression. One way to make the watermark sequences at the embedder and the decoder more similar is to make them less image adaptive, for example, by first lowpass filtering the image before calculating the perceptual mask. In this way, only the lower frequency components of the image, which tend to be better preserved under compression than the higher frequency components, contribute to the visual mask calculation, whereas the higher frequency components of the mask are made constant. Since the watermark typically has less energy in the lower frequency components, the perceptual mask calculated at the decoder will be more similar to the one used at the embedder. By changing the cutoff frequency of the lowpass filter, one can make the watermark progressively less image adaptive as the length of the payload is increased. In addition, if we use a transform with less overlap between the frequency support of different bands, the interference between sequences in different subbands in a frequency-partitioned spread transform scheme will also be reduced.

Another way of making the random images constructed in the watermark embedder and decoder more similar is to first use the cover image to generate the visual mask, and then iteratively generate the visual mask in the embedder using the watermarked image. The mask should be scaled before being used to multiply the CWT coefficients of the random images, so that its values, when calculated from the watermarked coefficients, will not be affected by the addition of the watermark.

7.2.4 Extension to video watermarks

The watermarking systems proposed in this thesis can be extended to video. However, computing the CWT on a frame by frame basis is too computationally intensive to make watermarking each frame *independently* practical. Besides, watermarking each frame differently is susceptible to the collusion attack. One should embed similar watermarks in similar frames and different watermarks in different frames (scene based watermarks). For example, we can use scene change detection algorithms to segment the video sequence into scenes, with one CWT computed from the first frame of each scene for calculating the basic visual mask for that scene. The watermark is then scaled with this mask and wrapped according to the overall (slow) motion of a particular frame with respect to the first frame. This also avoids the ‘dirty glass’ artifacts when the *same* watermark is embedded in all the frames of a scene.

Appendices

Appendix A Summary of visual models

We summarise here the visual models which are used in this thesis, namely the CWT model, the DWT model and the DCT model due to Watson [110, 158]. These models are used to calculate the perceptual mask used in our proposed spread spectrum based watermark embedding scheme. The three visual models below define the default local gain for the watermark in the respective domain. In all cases, the final watermark is scaled globally by a constant so that the energy of the watermark satisfies some constraint.

A.1 The CWT visual model

The CWT visual model in section 3.3.3 is repeated here for sake of completeness. In a particular resolution l and orientation θ , the gain for the watermark at location (u, v) is given by:

$$g(u, v) = \beta \sqrt{k^2 \overline{|x|^2} + C_{T_0}^2}. \quad (\text{A.1})$$

$\overline{|x|^2}$ is the result of lowpass filtering of the squared amplitude of the CWT coefficients in a 3×3 neighbourhood centered at (u, v) in a particular subband. β accounts for the change in gain due to the variation in background luminance and is approximated as:

$$\beta = 4.46(|x_{dc}| - 0.56)^2 + 1.02, \quad (\text{A.2})$$

where $|x_{dc}|$ is the normalised amplitude of the level 4 lowpass CWT coefficient. k and C_{T_0} are subband dependent constants and are listed in the table below: k_0 depends on the actual image and typically has range between 0.5 and 1. The average value of k_0 for the set of test images we used is about 0.9.

Level / Subband	k	C_{T_0}
Level 1 $\pm 75^\circ, \pm 15^\circ$	$1.00k_0$	1.5
Level 1 $\pm 45^\circ$	$0.55k_0$	3.1
Level 2 $\pm 75^\circ, \pm 15^\circ$	$1.05k_0$	0.75
Level 2 $\pm 45^\circ$	$1.05k_0$	0.9
Level 3 $\pm 75^\circ, \pm 15^\circ$	$1.10k_0$	1.05
Level 3 $\pm 45^\circ$	$1.35k_0$	1.0

Table A.1: Table of constants used in the CWT visual model

A.2 The DWT visual model

Due to the lack of a generic visual model for the DWT domain, and that all proposed models so far depend on a specific pair of wavelets (most commonly Daubechies' ones) being used, we decide to adopt our CWT visual model to the DWT domain. We use the same form (A.1) for the gain, but now θ corresponds to 90° , 0° and $\pm 45^\circ$. β in this case is given by:

$$\beta = 3.62(|x_{dc}| - 0.59)^2 + 0.68. \quad (\text{A.3})$$

The table A.2 gives the corresponding k and C_{T_0} . The average value of k_0 for the set of test images in appendix C is 2.1.

Level / Subband	k	C_{T_0}
Level 1 $90^\circ, 0^\circ$	$0.17k_0$	2.8
Level 1 $\pm 45^\circ$	$0.15k_0$	4.9
Level 2 $90^\circ, 0^\circ$	$0.29k_0$	1.45
Level 2 $\pm 45^\circ$	$0.30k_0$	1.9
Level 3 $90^\circ, 0^\circ$	$0.35k_0$	2.05
Level 3 $\pm 45^\circ$	$0.33k_0$	1.9

Table A.2: Table of constants used in the DWT visual model

A.3 The DCT visual model

The model due to Watson [110, 158] is originally used to design a quantisation matrix for JPEG compression, which is optimised for a given image. The model is divided in the three factors: contrast sensitivity, luminance masking, and contrast masking.

A.3.1 Contrast sensitivity

The detection threshold t_{ij} ($0 \leq i, j \leq 7$) for each DCT basis function has been measured and Ahumada [14] has extended this to a formula which gives the thresholds

under any luminance condition:

$$\log_{10} t_{ij} = \log_{10} \frac{T_{min}}{r_{ij}} + k(\log_{10} f_{ij} - \log_{10} f_{min})^2, \quad (\text{A.4})$$

$$r_{ij} = r + (1 - r) \cos^2 \theta_{ij}. \quad (\text{A.5})$$

T_{min} , k and f_{min} are given by:

$$T_{min} = \begin{cases} \frac{L}{S_0}, & \text{if } L > L_T, \\ \frac{L}{S_0} \left(\frac{L}{L_T}\right)^{1-a_t}, & \text{if } L \leq L_T, \end{cases} \quad (\text{A.6})$$

where $L_T = 13.45 \text{ cd/m}^2$, $S_0 = 94.7$ and $a_t = 0.65$. L is the average luminance of the image.

$$k = \begin{cases} k_0, & \text{if } L > L_k, \\ k_0 \left(\frac{L}{L_k}\right)^{a_k}, & \text{if } L \leq L_k, \end{cases} \quad (\text{A.7})$$

where $L_k = 300 \text{ cd/m}^2$, $k_0 = 3.125$ and $a_k = 0.0706$.

$$f_{min} = \begin{cases} f_0, & \text{if } L > L_f, \\ f_0 \left(\frac{L}{L_f}\right)^{a_f}, & \text{if } L \leq L_f, \end{cases} \quad (\text{A.8})$$

where $L_f = 300 \text{ cd/m}^2$, $f_0 = 6.78 \text{ cycles/deg}$ and $a_f = 0.182$. The spatial frequency f_{ij} of the DCT basis function (i, j) is:

$$f_{ij} = \frac{1}{16} \sqrt{(i/W_x)^2 + (j/W_y)^2}, \quad (\text{A.9})$$

where W_x and W_y are the horizontal and vertical size of a pixel in degree of visual angle. $0 < r < 1$ takes into account of the imperfect summation of the two Fourier components in the basis functions and is recommended to have a value of 0.7. Finally, the angular parameter θ_{ij} is given by:

$$\theta_{ij} = \arcsin \frac{2f_{i0}f_{0j}}{f_{ij}^2}. \quad (\text{A.10})$$

A.3.2 Luminance masking

The basic threshold t_{ij} from the last section is modified to take into account of the local variation of luminance as follows:

$$t_{ijk} = t_{ij} \left(\frac{c_{00k}}{\bar{c}_{00}} \right)^{a_t}, \quad (\text{A.11})$$

where t_{ijk} is the adjusted threshold t_{ij} for block k , c_{00k} is the DC coefficient of block k and $\bar{c}_{00} = 1024$, $a_t = 0.65$.

A.3.3 Contrast masking

The visibility of a basis function is reduced in the presence of an image component in the same frequency band. The final masking value m_{ijk} (gain of the watermark at frequency (i, j) in block k) is given by:

$$m_{ijk} = \max(t_{ijk}, |c_{ijk}|^{w_{ij}} t_{ijk}^{1-w_{ij}}), \quad (\text{A.12})$$

where $|c_{ijk}|$ is the magnitude of the DCT coefficient at that location and $w_{00} = 0$ and $w_{ij} = 0.7$ for all other frequencies. In practice, the user further scales m_{ijk} by a global factor (like α for the CWT visual model discussed in section 3.3.3) so that the watermark satisfies some energy constraint.

Appendix B Analysis of blind spread spectrum watermark decoding and detection using correlation

B.1 Performance of correlation based decoder

In this section we analyse the performance of the three types of correlation based decoder given in section 4.4. For the sake of completeness, we repeat our assumptions about the coefficient distributions as well as the mean/variance of the three types of decoder. We assume the host and watermark coefficients in a particular channel (if CWT is used as the watermark domain, the real parts and the imaginary parts are concatenated together) are distributed as follows:

$$\begin{aligned} \text{host : } \mathbf{x} &\sim \mathcal{N}(0, h_i^2 \sigma_x^2) \quad 0 \leq i \leq N - 1, \\ \text{watermark : } \mathbf{s} &\sim \mathcal{N}(0, g_i^2 \sigma_w^2) \quad 0 \leq i \leq N - 1 \quad \text{and } \sigma_x^2 \gg \sigma_w^2, \end{aligned} \quad (\text{B.13})$$

where $\mathcal{N}(\mu, \sigma^2)$ is a Gaussian distribution with mean μ and variance σ^2 and N is the number of coefficients. In other words, each element in the sequence can have different variance from one another, but they will all have zero mean. The expectation and the variance of the three types of decoder are as follows:

Case 1: Simple correlator

$$E(r_1) = \frac{1}{N} \sum_i g_i \sigma_w^2, \quad (\text{B.14})$$

$$\text{var}(r_1) = \frac{1}{N^2} \sum_i (h_i^2 \sigma_x^2 \sigma_w^2 + 2g_i^2 \sigma_w^4). \quad (\text{B.15})$$

Case 2: Matched filter

$$E(r_2) = \frac{1}{N} \sum_i g_i^2 \sigma_w^2, \quad (\text{B.16})$$

$$\text{var}(r_2) = \frac{1}{N^2} \sum_i (h_i^2 g_i^2 \sigma_x^2 \sigma_w^2 + 2g_i^4 \sigma_w^4). \quad (\text{B.17})$$

Case 3: Modified matched filter

$$E(r_3) = \sigma_w^2, \quad (\text{B.18})$$

$$\text{var}(r_3) = \frac{1}{N^2} \sum_i \left(\frac{h_i^2}{g_i^2} \sigma_x^2 \sigma_w^2 + 2\sigma_w^4 \right). \quad (\text{B.19})$$

Now consider the case where there is an element in the sequence with h_k ($0 \leq k \leq N-1$) such that $h_k \gg h_i, \forall i \neq k$. This also means that $g_k \gg g_i, \forall i \neq k$ as the watermark is adapted to the image. The term associated with h_k and g_k will dominate the sums in the equations (B.14) to (B.17)². Let $\chi_i = h_i/h_k$ and $\xi_i = g_i/g_k$ ($i \neq k$) respectively, such that $0 < \chi_i, \xi_i \ll 1$. We can rewrite (B.14) to (B.17) as follows:

$$E(r_1) = \frac{1}{N} g_k \sigma_w^2 \left(1 + \sum_{i \neq k} \xi_i \right), \quad (\text{B.20})$$

$$\text{var}(r_1) = \frac{1}{N^2} \left(h_k^2 \sigma_x^2 \sigma_w^2 \left(1 + \sum_{i \neq k} \chi_i^2 \right) + 2g_k^2 \sigma_w^4 \left(1 + \sum_{i \neq k} \xi_i^2 \right) \right). \quad (\text{B.21})$$

$$E(r_2) = \frac{1}{N} g_k^2 \sigma_w^2 \left(1 + \sum_{i \neq k} \xi_i^2 \right), \quad (\text{B.22})$$

$$\text{var}(r_2) = \frac{1}{N^2} \left(h_k^2 g_k^2 \sigma_x^2 \sigma_w^2 \left(1 + \sum_{i \neq k} \chi_i^2 \xi_i^2 \right) + 2g_k^4 \sigma_w^4 \left(1 + \sum_{i \neq k} \xi_i^4 \right) \right). \quad (\text{B.23})$$

Expressions for the ratio ρ (defined as $E/\sqrt{\text{var}}$) for the three decoders can be derived as follows:

$$\begin{aligned} \rho_1 &= \frac{g_k \sigma_w^2 \left(1 + \sum_{i \neq k} \xi_i \right)}{\sqrt{h_k^2 \sigma_x^2 \sigma_w^2 \left(1 + \sum_{i \neq k} \chi_i^2 \right) + 2g_k^2 \sigma_w^4 \left(1 + \sum_{i \neq k} \xi_i^2 \right)}}, \\ &\approx \underbrace{\frac{g_k \sigma_w}{h_k \sigma_x} \left(\frac{1 + \sum_{i \neq k} \xi_i}{\sqrt{1 + \sum_{i \neq k} \chi_i^2}} \right)}_a \quad \because \sigma_x^2 \gg \sigma_w^2. \end{aligned} \quad (\text{B.24})$$

²There is no loss of generality here as we can extend the argument to the case where there are several big coefficients, by partitioning our sequences into several sub-sequences, each with one big coefficient and treat each sub-sequence as a separate channel. Since the resulting SNR is the sum of SNR of each sequence (equation 4.27), the proof is still valid. This extension works as long as the number of large coefficients is small compared with the total number of coefficients.

$$\begin{aligned}
\rho_2 &= \frac{g_k^2 \sigma_w^2 \left(1 + \sum_{i \neq k} \xi_i^2\right)}{\sqrt{h_k^2 g_k^2 \sigma_x^2 \sigma_w^2 \left(1 + \sum_{i \neq k} \chi_i^2 \xi_i^2\right) + 2g_k^4 \sigma_w^4 \left(1 + \sum_{i \neq k} \xi_i^4\right)}}, \\
&\approx \underbrace{\frac{g_k \sigma_w}{h_k \sigma_x} \left(\frac{1 + \sum_{i \neq k} \xi_i^2}{\sqrt{1 + \sum_{i \neq k} \chi_i^2 \xi_i^2}} \right)}_b. \tag{B.25}
\end{aligned}$$

$$\begin{aligned}
\rho_3 &= \frac{\sigma_w^2}{\frac{1}{N} \sqrt{\sum_i \frac{h_i^2}{g_i^2} \sigma_x^2 \sigma_w^2 + 2\sigma_w^4}}, \\
&\approx \frac{\sqrt{N} g_k \sigma_w}{h_k \sigma_x}, \tag{B.26}
\end{aligned}$$

where the last line of (B.26) follows from the assumption that $h_i \approx g_i$ throughout the sequence. It is not obvious how big the terms marked with (a) and (b) are compared with \sqrt{N} . If we assume $\chi_i = \xi_i \forall i$, and let:

$$\Xi(\xi) = \frac{(1 + \sum \xi_i)^2}{1 + \sum \xi_i^2}, \tag{B.27}$$

$$\Upsilon(\xi, n) = \frac{(1 + \sum \xi_i^n)^2}{1 + \sum \xi_i^{2n}}, \tag{B.28}$$

where we have dropped the index range on the summation signs for clarity and all sums in (B.27, B.28) have $N - 1$ terms, we can show that:

$$\begin{aligned}
N &> \Xi(\xi) = \Upsilon(\xi, 1) > \Upsilon(\xi, 2), \\
\Rightarrow \rho_3 &> \rho_1 > \rho_2. \tag{B.29}
\end{aligned}$$

In order to show $N > \Xi(\xi)$, we first note that if all ξ_i were equal to 1, then $\Xi(\xi) = N$. Next, we consider the sign of the first and the second partial derivatives of $\Xi(\xi)$ with respect to a particular element ξ_j in the summation. The first partial derivative is given by:

$$\frac{\partial \Xi}{\partial \xi_j} = \frac{2(1 + \sum \xi_i)(1 - \xi_j + \sum \xi_i^2 - \xi_j \sum \xi_i)}{(1 + \sum \xi_i^2)^2}. \tag{B.30}$$

The sign of (B.30) is given by the sign of the numerator. $1 - \xi_j > 0$ since $\xi_j < 1 \forall j$,

but the sign of $\sum \xi_i^2 - \xi_j \sum \xi_i$ depends on the actual values of ξ_i . If we assume the sequence of ξ_i are i.i.d., this term will be non-negative on average because:

$$\begin{aligned}
& E\left(\sum \xi_i^2 - \xi_j \sum \xi_i\right), \\
& \approx (N-1)E(\xi^2) - E(\xi)(N-1)E(\xi), \\
& = (N-1)(E(\xi^2) - E^2(\xi)), \\
& \geq 0.
\end{aligned} \tag{B.31}$$

The gradient of (B.30) with respect to ξ will therefore be positive on average. We now calculate the numerator of the second partial derivative to determine its sign, which is given as follows in its expanded form:

$$\begin{aligned}
\text{sign}\left(\frac{\partial^2 \Xi}{\partial \xi_j^2}\right) &= \text{sign}\left(\sum \xi_i^2 + 2\left(\sum \xi_i^2\right)^2 + \left(\sum \xi_i^2\right)^3 - 2\sum \xi_i - 4\sum \xi_i \sum \xi_i^2\right. \\
&\quad - 2\sum \xi_i \left(\sum \xi_i^2\right)^2 - \left(\sum \xi_i\right)^2 - 2\left(\sum \xi_i\right)^2 \sum \xi_i^2 - \left(\sum \xi_i\right)^2 \left(\sum \xi_i^2\right)^2 \\
&\quad - 4\xi_j + 4\xi_j^2 - 4\xi_j \sum \xi_i - 8\xi_j \sum \xi_i^2 + 8\xi_j^2 \sum \xi_i + 4\xi_j^2 \sum \xi_i^2 \\
&\quad - 8\xi_j \sum \xi_i \sum \xi_i^2 + 4\xi_j^2 \left(\sum \xi_i\right)^2 - 4\xi_j \left(\sum \xi_i^2\right)^2 \\
&\quad \left. + 8\xi_j^2 \sum \xi_i \sum \xi_i^2 - 4\xi_j \sum \xi_i \left(\sum \xi_i^2\right)^2 - 4\xi_j \left(\sum \xi_i\right)^2 \sum \xi_i^2\right)
\end{aligned} \tag{B.32}$$

By comparing the magnitudes of the positive and negative terms in (B.32) and using facts such as $\sum \xi_i > \sum \xi_i^2$ and $\xi_j \ll \sum \xi_i$, we can show that (B.32) is always negative. Therefore $\Xi(\xi)$ is a concave function of ξ_i , and $\Xi(\xi) < N$ for $\xi_i < 1$. Since the function Ξ is continuous in ξ , we can argue that the term marked a in (B.24) is smaller than \sqrt{N} in the small neighbourhood around ξ such that $\xi_i \approx \chi_i$, hence $\rho_3 > \rho_1$.

In the second part of our proof, we show $\Upsilon(\xi, 1) > \Upsilon(\xi, 2)$ by considering the partial derivative of Υ with respect to n :

$$\frac{\partial \Upsilon}{\partial n} = \frac{2(1 + \sum \xi_i^n)(\sum \ln(\xi_i) \xi_i^n - \sum \ln(\xi_i) \xi_i^{2n})}{(1 + \sum \xi_i^{2n})^2}, \tag{B.33}$$

which is always negative because $\ln(\xi_i) < 0$ and so $\ln(\xi_i)\xi_i^n < \ln(\xi_i)\xi_i^{2n} \forall i$. Using the continuity principle on Υ with respect to n and ξ respectively, we find that $\Upsilon(\xi, 1) > \Upsilon(\xi, 2)$ and $\rho_1 > \rho_2$, which complete our proof. We can therefore infer that the modified correlator should perform better than both the simple correlator and matched filter under non-stationary noise conditions.

B.2 Blind spread spectrum watermark detection

This section derives the false positive, miss and detection probabilities of the two approaches for watermark detection described in section 4.6, namely fixing L bits of the payload to some known pattern or allocate part of the watermark energy to a separate reference watermark. We assume *no* error control coding is used here to encode the payload. There are two possible situations where a false positive can occur:

1. The watermark detector declares a watermark is present when in fact there is *none*.
2. The watermark detector declares a watermark is present when in fact another watermark marked with the *wrong* key is present.

We can modify our null hypothesis to cope with the two cases as follows:

$$\begin{aligned}
 H_0 &: \mathbf{y} = \mathbf{e} && \text{no watermark,} \\
 H_0^* &: \mathbf{y} = \mathbf{s}^* + \mathbf{e} && \text{wrong key,} \\
 H_1 &: \mathbf{y} = \mathbf{s} + \mathbf{e}. &&
 \end{aligned} \tag{B.34}$$

where \mathbf{e} and \mathbf{s} are the noise (which includes the host signal) and the desired signal respectively. η from (4.30) is dropped for the sake of clarity. \mathbf{s}^* is another watermark with the wrong key such that $E(\mathbf{s} \cdot \mathbf{s}^*) = 0$. The statistics of the correlator under these hypotheses are all Gaussian (statistics under H_0^* can be derived in a similar way as described in section 4.4) and are summarized as follows:

$$\begin{aligned}
 \mu_{H_0} = \mu_{H_0^*} = 0; \quad \mu_{H_1} &\neq 0, \\
 \sigma_{H_0^*}^2 \approx \sigma_{H_1}^2; \quad \sigma_{H_0^*}^2, \sigma_{H_1}^2 &> \sigma_{H_0}^2.
 \end{aligned} \tag{B.35}$$

$\sigma_{H_0^*}^2 \approx \sigma_{H_1}^2$ because the image energy is much larger than the watermark energy and $E(\mathbf{s}^* \cdot \mathbf{e}) = 0$. Since $\sigma_{H_0^*} > \sigma_{H_0}$, it is the worst case scenario and we should use $\sigma_{H_0^*}$ in our calculations. In the following analysis, we assume the energy of the payload is E_b per bit and that the noise power spectral density (PSD) is N_0 (i.e. assuming white noise). We also assume the length of the payload is M bits. Therefore the SNR *per bit* is given by E_b/N_0 and the total energy E of the watermark is $M \cdot E_b$.

B.2.1 L fixed bits

It can easily be seen that the decoder will decode 1 and 0 with equal probability in an unwatermarked image ($\because \mu_{H_0^*} = 0$), the probability of getting L bits correctly at random (and hence false alarm) is simply:

$$P_{FP} = 2^{-L}. \quad (\text{B.36})$$

If we require a false alarm rate of say 10^{-6} , then we need $L \geq 20$. In order to keep the distortion level in the watermarked image the same, the energy of the payload has to be reduced by a factor of $\frac{M}{L+M}$. The effective energy per bit and the effective SNR per bit are:

$$\begin{aligned} E'_b &= \frac{M}{L+M} E_b, \\ SNR' &= \frac{M}{L+M} SNR. \end{aligned} \quad (\text{B.37})$$

The new bit error rate can be derived from (4.14) using the new SNR as:

$$P_{error} = Q \left(\sqrt{\frac{M}{L+M} \cdot SNR} \right), \quad (\text{B.38})$$

where SNR is the original SNR *per bit* at the correlator in the absence of the fix pattern. Assuming the decoder decodes *1 bit at a time*, we will miss a watermark if *any* of these L bits is decoded wrongly. Hence probability of detection is:

$$\begin{aligned} P_D &= P(\text{all } L \text{ bits are decoded correctly}), \\ &= \left(1 - Q \left(\sqrt{\frac{M}{L+M} \cdot SNR} \right) \right)^L, \end{aligned} \quad (\text{B.39})$$

and the miss probability is simply:

$$P_{Miss} = 1 - \left(1 - Q \left(\sqrt{\frac{M}{L+M} \cdot SNR} \right) \right)^L. \quad (\text{B.40})$$

B.2.2 Separate reference watermark

If we allocate some of our watermark energy to a separate reference watermark, say, by scaling the reference and the payload by α_1 and α_2 such that $\alpha_1^2 + \alpha_2^2 = 1$, so as

to keep the resulting distortion the same. The payload total energy and the reference watermark energy are $\alpha_2^2 E$ and $\alpha_1^2 E$ respectively. If we equate the total energy of the reference watermark in the two cases, we can see that this is equivalent to setting L to:

$$L = \frac{\alpha_1^2 M}{1 - \alpha_1^2}. \quad (\text{B.41})$$

Since the entire reference watermark is detected as a whole, the effective SNR of the reference watermark is:

$$\begin{aligned} SNR_{ref\ mark} &= \frac{\alpha_1^2 E}{N_0}, \\ &= \alpha_1^2 M \cdot SNR, \\ &= \frac{LM}{L + M} \cdot SNR. \end{aligned} \quad (\text{B.42})$$

SNR is the SNR *per bit* without the reference as before. The user chooses the desired false alarm rate, say p , and computes the threshold λ based on Neyman-Pearson. The relationship between p and λ is:

$$P_{FP} = p, \quad (\text{B.43})$$

$$\lambda = \sigma_{H_0^*} Q^{-1}(p), \quad (\text{B.44})$$

where $Q^{-1}(\cdot)$ is the inverse of the Q function defined in (4.15). The watermark will be missed if the output of the correlator is less than λ . The miss probability can be derived as follows:

$$\begin{aligned} P_{Miss} &= P(\text{output} < \lambda | H_1), \\ &= Q\left(\frac{\mu_{H_1} - \lambda}{\sigma_{H_1}}\right), \\ &= Q\left(\frac{\mu_{H_1} - \sigma_{H_0^*} Q^{-1}(p)}{\sigma_{H_1}}\right), \\ &\approx Q\left(\sqrt{\frac{LM}{L + M}} \cdot SNR - Q^{-1}(p)\right). \end{aligned} \quad (\text{B.45})$$

The last line follows from (B.35) and (B.42). The probability of detection is just:

$$P_D \approx 1 - Q \left(\sqrt{\frac{LM}{L+M}} \cdot SNR - Q^{-1}(p) \right). \quad (\text{B.46})$$

B.2.3 Special case: a *yes/no* watermark

In case of a *yes/no* watermark (i.e. $M = 0$), the above expressions of detection/miss probabilities need to be modified. If we split the watermark up into L bits and use the first approach for detection, the SNR is reduced by a factor of L for each bit and we get:

$$P_{D_{L \text{ bit}}} = \left(1 - Q \left(\sqrt{\frac{SNR}{L}} \right) \right)^L, \quad (\text{B.47})$$

where SNR is the ratio of the total watermark energy to the noise PSD. On the other hand if we detect this *yes/no* watermark in the conventional way, we find:

$$P_{D_{ref}} \approx 1 - Q \left(\sqrt{SNR} - Q^{-1}(p) \right). \quad (\text{B.48})$$

The false alarm probabilities are independent of M and are thus unaffected.

Appendix C Test images

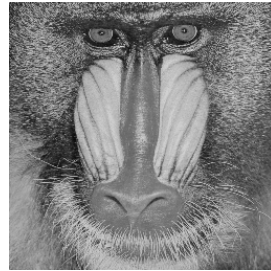
All images are 256 grey scale and are of 256×256 pixels. All images (except Pills, Newyork and Lochness) courtesy of the Signal and Image Processing Institute at the University of Southern California.



Lena



Camera



Baboon



Claire



Couple



Tng2



Pentagon



Peppers



Fishingboat



Pills*



Indust



F16



Bridge



Newyork



Barbara



Lochness

*Pills - copyright photo courtest of Karel de Gendie.

Bibliography

- [1] *Proc. of International Conference on Image Processing*, Sep 1996. (pp. 152, 159, 160, 161)
- [2] *Proc. of International Conference on Image Processing*, Santa Barbara, CA, Oct 1997. (pp. 151, 154, 156, 159, 160)
- [3] *Optical Express*, volume 3, Dec 1998. (pp. 150, 163)
- [4] *Proc. 2nd Workshop on Information Hiding*, volume 1525, Portland, OR, Apr 1998. Springer-Verlag, LNCS. (pp. 153, 154, 157, 158, 159)
- [5] *Proc. of European Signal Processing Conference*, Rhode Island, Greece, Sep 1998. (pp. 155, 157)
- [6] *Proc. of IEEE Workshop on Multimedia Signal Processing*, Dec 1998. (pp. 158, 163)
- [7] *Proc. of the IEEE International Conference on Image Processing*, Chicago, IL, Oct 1998. (pp. 155, 156, 158, 163)
- [8] *Proc. 3rd Workshop on Information Hiding*, volume 1768, Dresden, Germany, Sept 1999. Springer-Verlag, LNCS. (pp. 152, 153, 155, 157, 162)
- [9] *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, Phoenix, AZ, Mar 1999. (pp. 155, 161)
- [10] *Proc. of International Conference on Image Processing*, Kobe, Japan, Oct 1999. (pp. 155, 157, 161)
- [11] *Proc. of European Conference on Signal Processing*, Tampere, Finland, Sept 2000. (pp. 153, 162)
- [12] *4th Information Hiding Workshop*, Pittsburgh, PA, USA, Apr 2001. (pp. 153, 159)

- [13] *Proc. of International Conference on Image Processing*, Thessaloniki, Greece, Oct 2001. (pp. 158, 162)
- [14] A. J. Ahumada and H. A. Peterson. Luminance model based DCT Quantization for color Image. In *Proc. of SPIE Human vision, Visual Processing and Digital Display III*, volume 1666, pages 365–374, 1992. (p. 136)
- [15] P. Anandan. A Computational Framework and an Algorithm for the Measurement of Visual Motion. *International Journal of Computer Vision*, 2:283–310, 1989. (p. 92)
- [16] F. Balado and F. Pérez-González. Coding at the sample level for data hiding: turbo and concatenated codes. In Wong and Delp [163], pages 532–543. (p. 20)
- [17] M. Barni, F. Bartolini, and A. Piva. Improved Wavelet-Based Watermarking Through Pixel-Wise Masking. *IEEE Trans. on Image Processing*, 10(5):783–791, May 2001. (pp. 57, 58, 59, 61, 64, 66, 68, 72)
- [18] W. Bender, D. Gruhl, N. Morimoto, and A. Lu. Techniques for data hiding. *IBM Systems Journal*, 35(3/4):313–336, 1996. (p. 13)
- [19] C. Berrou, A. Glavieux, and P. Thitimajshima. Near Shannon limit error-correcting coding: turbo codes. In *Proc. IEEE International Conference on Communication*, pages 1064–1070, Geneva, Switzerland, 1993. (p. 17)
- [20] K. A. Birney and T. R. Fischer. On the modeling of dct and subband image data for compression. *IEEE Trans. on Image Processing*, 4(2):186–193, Feb 1995. (p. 78)
- [21] J. A. Bloom, I. J. Cox, T. Kalker, J.-P. M. G. Linnartz, M. L. Miller, and B. Traw. Copy Protection for DVD Video. *Proc. of the IEEE Special Issue on Identification and Protection of Multimedia Information*, 87(7):1267–1276, Jul 1999. (p. 6)
- [22] R. S. Blum, R. J. Kozick, and B. M. Sadler. An Adaptive Spatial Diversity Receiver for Non-Gaussian Interference and Noise. *IEEE Trans. on Signal Processing*, 47(8):2100–2111, Aug 1999. (p. 78)
- [23] A. G. Bors and I. Pitas. Image watermarking using block site selection and DCT domain constraints. In *Optical Express* [3], pages 512–523. (pp. 13, 14)

- [24] G. W. Braudaway. Protecting Publicly-Available Images with an Invisible Image Watermark. In *Proc. of International Conference on Image Processing* [2], pages 524–527. (p. 17)
- [25] G. W. Braudaway, K. A. Magerlein, and F. Mintzer. Color Correct Digital Watermarking of Images. US patent No. 5530759, 1996. (p. 3)
- [26] A. Burr. Turbo-codes:the ultimate error control codes? *IEE Electronics and Communications Journal*, 13(4):155–165, Aug 2001. (p. 19)
- [27] G. Caronni. Ermitteln unauthorisierter verteriler von maschinenlesbaren daten. Technical report, ETH Zürich, Switzerland, Aug 1993. (p. 2)
- [28] J. C. Carr, W. R. Fright, and R. K. Beatson. Surface interpolation with radial basis functions for medical imaging. *IEEE Trans. on Medical Imaging*, 16(1):96–107, Feb 1997. (p. 89)
- [29] B. Chen and G. Wornell. Dither modulation: A new approach to digital watermarking and information embedding. In Wong and Delp [161]. (p. 108)
- [30] B. Chen and G. Wornell. Provably Robust Digital Watermarking. In *Proc. of SPIE, Multimedia Systems and Applications II*, volume 3845, Boston, MA, Sep 1999. (pp. 108, 112, 114)
- [31] B. Chen and G. Wornell. Preprocessed and Postprocessed Quantization Index Modulation Methods for Digital Watermarking. In Wong and Delp [162]. (p. 110)
- [32] B. Chen and G. W. Wornell. Quantization Index Modulation: A Class of Provably Good Methods for Digital Watermarking and Information Embedding. *IEEE Trans. on Information Theory*, 47(4):1423–1443, May 2001. (p. 21)
- [33] Q. Cheng and T. S. Huang. Blind digital watermarking for images and videos and performance analysis. In *Proc. ICME*, New York, Aug 2000. (p. 78)
- [34] C. Christopolous, A. Skodras, and T. Ebrahimi. The JPEG2000 still image coding system: an overview. *IEEE Trans. on Consumer Electronics*, 46(4):1103–1127, Nov 2000. (pp. 2, 25)
- [35] M. J. Coates, E. E. Kuruoglu, and W. J. Fitzgerald. Time-frequency based detection in alpha-stable distributed noise environments. In *IEEE. Workshop Higher Order Statistics*, Caesarea, Israel, Jun 1999. (p. 78)

- [36] D. Coltuc and P. Bolon. Watermarking by histogram specification. In Wong and Delp [161], pages 252–263. (p. 16)
- [37] S. Comes and B. Macq. Human Visual Quality Criterion. In *Proc. of SPIE Visual Communication and Image Processing*, volume 1360, pages 2–13, Lausanne, Switzerland, Oct 1990. (pp. 32, 33, 37)
- [38] M. H. M. Costa. Writing on Dirty Paper. *IEEE Trans. on Information Theory*, 29(3):438–441, May 1983. (pp. 103, 105, 106, 109, 110, 117)
- [39] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. John Wiley and Sons, Inc., 1991. (p. 106)
- [40] I. J. Cox. Secure Spread Spectrum Watermarking for Multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687, Dec 1997. (p. 95)
- [41] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon. A Secure Robust Watermark for Multimedia. In *Proc. 1st Workshop on Information Hiding*, volume 1174, pages 183–206, Cambridge, UK, Jun 1996. Springer-Verlag, LNCS. (p. 12)
- [42] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for images, audio and video. In *Proc. of International Conference on Image Processing [1]*, pages 243–246. (p. 12)
- [43] I. J. Cox, M. L. Miller, and A. L. McKellips. Watermarking as Communications with Side Information. *Proc. of IEEE Special Issue on Identification and Protection of Multimedia Information*, 87(7):1127–1141, Jul 1999. (p. 103)
- [44] S. Craver. Zero Knowledge Watermark Detection. In *Proc. 3rd Workshop on Information Hiding [8]*, pages 101–116. (p. 8)
- [45] S. Craver, N. Memon, B.-L. Yeo, and M. Yeung. Can Invisible Watermarks Resolve Rightful Ownerships? *IBM Research Report RC 20509*, July 1996. (p. 8)
- [46] V. Darmstaedter, J.-F. Delaigle, J. J. Quisquater, and B. Macq. Low Cost Spatial Watermarking. *Computer and Graphics*, 22(4):417–424, 1998. (p. 13)
- [47] A. de Rosa, M. Barni, F. Bartolini, V. Cappellini, and A. Piva. Optimal Decoding of Non-additive Full Frame DFT Watermarks. In *Proc. 3rd Workshop on Information Hiding [8]*, pages 159–171. (pp. 14, 21, 84)

- [48] F. Deguillaume, S. Voloshynovskiy, and T. Pun. Method for the Estimation and Recovering from General Affine Transforms in Digital Watermarking Applications. In P. W. Wong and E. J. Delp, editors, *Proc. of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging IV*, volume 4675, San Jose, CA, Jan 2002. (p. 133)
- [49] J. Dittmann, A. Behr, M. Stabenau, P. Schmitt, J. Schwenk, and J. Ueberberg. Combining digital Watermarks and collusion secure Fingerprints for digital Images. In Wong and Delp [161], pages 171–182. (p. 6)
- [50] J. J. Eggers and B. Girod. Watermark Detection after Quantization Attacks. In *Proc. 3rd Workshop on Information Hiding* [8], pages 172–186. (p. 75)
- [51] J. J. Eggers and B. Girod. Quantization Effects on Digital Watermarks. *Signal Processing*, 81(2):239–263, 2001. (p. 75)
- [52] J. J. Eggers, J. K. Su, and B. Girod. A Blind Watermarking Scheme Based on Structured Codebooks. In *Proc. of IEEE Seminar on Secure Images and Image Authentication*, Apr 2000. (pp. 108, 110, 117, 118)
- [53] J. J. Eggers, J. K. Su, and B. Girod. Public Key Watermarking By Eigenvectors of Linear Transforms. In *Proc. of European Conference on Signal Processing* [11]. (p. 8)
- [54] J. J. Eggers, J. K. Su, and B. Girod. Performance of a Practical Blind Watermarking Scheme. In Wong and Delp [163]. (pp. 108, 110, 117, 118)
- [55] C. Fei, D. Kundur, and R. Kwong. The Choice of Watermark Domain in the Presence of Compression. In *Proc. of IEEE Int. Conf. on Information Technology: Coding and Computing*, Las Vegas, Apr 2001. (p. 76)
- [56] J. Fridrich. Methods for Detecting Changes in Digital Images. In *Proc. of the 6th IEEE International Workshop on Intelligent Signal Processing and Communication Systems*, Melbourne, Australia, Nov 1998. (p. 7)
- [57] J. Fridrich, A. C. Baldoza, and R. J. Simard. Robust digital watermarking based on key-dependent basis functions. In *Proc. 2nd Workshop on Information Hiding* [4], pages 143–157. (p. 15)
- [58] J. Fridrich, M. Goljan, and R. Du. Distortion-free Data Embedding. In *4th Information Hiding Workshop* [12]. (p. 7)

- [59] J. Fridrich, M. Goljan, and R. Du. Invertible authentication. In Wong and Delp [163]. (p. 7)
- [60] S. I. Gel'fand and M. S. Pinsker. Coding for channel with random parameters. *Problems of Control and Information Theory*, 9(1):19–31, 1980. (p. 106)
- [61] A. Goshtasby. Registration of Images with Geometric Distortions. In *IEEE Trans. on Geoscience and Remote Sensing*, pages 60–64, Jan 1988. (p. 83)
- [62] J. Hagenauer and L. Papke. Decoding turbo codes with the soft-output Viterbi algorithm (SOVA). In *Proc. of IEEE Int. Symposium on Information Theory*, page 164, Trondheim, Norway, Jun 1994. (p. 19)
- [63] F. Hartung and B. Girod. Fast Public-Key Watermarking of Compressed Video. In *Proc. of International Conference on Image Processing* [2], pages 528–531. (p. 17)
- [64] C. Heegard and A. A. El Gamal. On the Capacity of Computer Memory with Defects. *IEEE Trans. on Information Theory*, 29(5):731–739, Sep 1983. (p. 106)
- [65] E. F. Hembrooke. Identification of sound and like signals. US patent No. 3004104, 1961. (p. 2)
- [66] J. R. Hernández, M. Amado, and F. Pérez-González. DCT-domain watermarking techniques for still images. *IEEE Trans. on Image Processing*, 9:55–68, Jan 2000. (p. 78)
- [67] A. Herrigel, J. Ó Ruanaidh, H. Petersen, S. Pereira, and T. Pun. Secure copyright protection techniques for digital images. In *Proc. 2nd Workshop on Information Hiding* [4], pages 169–190. (p. 7)
- [68] A. Herrigel, S. Voloshynovskiy, and Y. Rytsar. The Watermark Template Attack. In Wong and Delp [163]. (pp. 81, 95)
- [69] J. Huang and Y. Q. Shi. Adaptive image watermarking scheme based on visual masking. In *Electronic Letters*, volume 34, pages 748–750, Apr 1998. (p. 14)
- [70] A. K. Jain. *Fundamentals of Digital Image Processing*. Prentice Hall Information and System Sciences Series. Prentice Hall, 1989. (p. 31)

- [71] JJ2000. An Implementation of the JPEG2000 standard in Java. <http://jj2000.epfl.ch>. (p. 62)
- [72] N. F. Johnson, Z. Duric, and S. Jajodia. Recovery of Watermark from Distorted Images. In *Proc. 3rd Workshop on Information Hiding* [8], pages 318–332. (p. 81)
- [73] T. Kalker, G. Depovere, J. Haitsma, and M. Maes. A Video Watermarking System for Broadcast Monitoring. In Wong and Delp [161]. (p. 16)
- [74] T. Kalker and A. J. E. M. Janssen. Analysis of Watermark Detection Using SMOPF. In *Proc. of International Conference on Image Processing* [10]. (pp. 21, 54)
- [75] T. Kalker, J.-P. M. G. Linnartz, and M. van Dijk. Watermark estimation through detector analysis. In *Proc. of the IEEE International Conference on Image Processing* [7]. (p. 23)
- [76] S. A. Karunasekera and N. G. Kingsbury. A Distortion Measure for Image Artefacts Based on Human Visual Sensitivity. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, volume 5, pages 117–120, Apr 1994. (pp. 35, 39, 64)
- [77] A. Kerckhoffs. La Cryptographie Militaire. *Journal des Sciences Militaires*, 9:5–38, Jan 1883. <http://www.cl.cam.ac.uk/~fapp2/kerckhoffs>. (p. 5)
- [78] M. Kesal, M. K. Mihcak, R. Koetter, and P. Moulin. Iteratively decodable codes for watermarking applications. In *Proc. 2nd Symposium on Turbo Codes and Their Applications*, Brest, France, Sep 2000. (p. 20)
- [79] N. G. Kingsbury. The Dual-Tree Complex Wavelet Transform: A New Efficient Tool for Image Restoration and Enhancement. In *Proc. of European Signal Processing Conference* [5], pages 319–322. (p. 27)
- [80] N. G. Kingsbury. Shift Invariant Properties of the Dual-Tree Complex Wavelet Transform. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)* [9]. (p. 27)
- [81] N. G. Kingsbury. A Dual-Tree Complex Wavelet Transform with Improved Orthogonality and Symmetry Properties. In *Proc. of International Conference on Image Processing*, Vancouver, Canada, Oct 2000. (pp. 27, 28)

- [82] N. G. Kingsbury. Complex wavelets for shift invariant analysis and filtering of signals. submitted to (by invitation) to *Journal of Applied Computation and Harmonic Analysis*, Jun 2000. (p. 27)
- [83] E. Koch, J. Rindfrey, and J. Zhao. Copyright protection for multimedia data. In *Proc. of the International Conference on Digital Media and Electronic Publishing*, 1994. (pp. 12, 14)
- [84] E. Koch and J. Zhao. Towards Robust and Hidden Image. In *Proc. of the IEEE International Workshop on Nonlinear Signal and Image Processing*, pages 452–455, Halkidiki, Marmaras, Greece, Jun 1995. (pp. 12, 14, 95)
- [85] A. C. Kokaram. *Motion Picture Restoration*. PhD thesis, University of Cambridge, Cambridge, UK, May 1993. (p. 83)
- [86] A. C. Kokaram and S. J. Godsill. A system for reconstruction of missing data in image sequences using sampled 3D AR models and MRF motion priors. In *Computer Vision - ECCV' 96*, volume II, pages 613–624. Springer-Verlag, LNCS, Apr 1996. (p. 83)
- [87] D. Kundur and D. Hatzinakos. A Robust Digital Image Watermarking Method using Wavelet-Based Fusion. In *Proc. of International Conference on Image Processing [2]*, pages 544–547. (p. 14)
- [88] D. Kundur and D. Hatzinakos. Digital Watermarking Using Multiresolution Wavelet Decomposition. In *Proc. of International Conference on Acoustics, Speech and Signal Procsssing (ICASSP)*, pages 2969–2972, May 1998. (p. 15)
- [89] D. Kundur and D. Hatzinakos. Improved robust watermarking through attack characterisation. *Optics Express focus issue on Digital Watermarking*, 3(12):485–490, Dec 1998. (pp. 13, 59, 119, 121, 122)
- [90] D. Kundur and D. Hatzinakos. Towards a telltale watermarking technique for tamper-proofing. In *Proc. of the IEEE International Conference on Image Processing [7]*, pages 409–413. (p. 7)
- [91] M. Kutter. Watermarking resisting to translation, rotation and scaling. In *Proc. of SPIE, Multimedia Systems and Applications*, volume 3528, pages 523–531, Nov 1998. (pp. 13, 21, 54, 95)

- [92] M. Kutter. *Digital Image Watermarking: Hiding Information in Images*. PhD thesis, EPFL, Lausanne, Switzerland, 1999. (p. 95)
- [93] M. Kutter. Performance Improvement of Spread Spectrum Based Image Watermarking Schemes Through M-ary Modulation. In *Proc. 3rd Workshop on Information Hiding* [8], pages 237–252. (p. 20)
- [94] M. Kutter, S. K. Bhattacharjee, and T. Erbahimi. Towards Second Generation Watermarking Schemes. In *Proc. of International Conference on Image Processing* [10]. (pp. 96, 133)
- [95] M. Kutter and F. A. P. Petitcolas. A fair benchmark for image watermarking systems. In Wong and Delp [161], pages 226–239. <http://www.cl.cam.ac.uk/~fapp2/watermarking/stirmark/>. (pp. 23, 73, 81, 95)
- [96] M. Kutter, S. Voloshynovskiy, and A. Herrigel. Watermark copy attack. In Wong and Delp [162]. (p. 23)
- [97] G. C. Langelaar, R. L. Lagendijk, and J. Biemond. Removing Spatial Spread Spectrum Watermarks by Non-linear Filtering. In *Proc. of European Signal Processing Conference* [5]. (p. 22)
- [98] J. Lévy Véhel and A. Manoury. Wavelet packet based digital watermarking. In *Proc. of International Conference on Pattern Recognition*, 2000. (p. 15)
- [99] A. S. Lewis and G. Knowles. Image Compression Using the 2-D Wavelet Transform. *IEEE Trans. on Image Processing*, 1(2):244–250, Apr 1992. (p. 57)
- [100] J. S. Lim. *Two-Dimensional Signal and Image Processing*. Prentice Hall Signal Processing Series. Prentice Hall, 1990. (p. 31)
- [101] J.-P. M. G. Linnartz. The ‘Ticket’ Concept for Copy Control Based on Embedded Signalling. In *Computer Security - 5th European Symposium on Research in Computer Security*, volume 1485, pages 257–274. Springer-Verlag, LNCS, 1998. (p. 6)
- [102] J.-P. M. G. Linnartz, T. Kalker, and G. Depovere. Modelling the false alarm and missed detection rate for electronic watermark. In *Proc. 2nd Workshop on Information Hiding* [4]. (p. 68)

- [103] P. Loo and N. G. Kingsbury. Motion Estimation Based Registration of Geometrically Distorted Images for Watermark Recovery. In Wong and Delp [163]. (p. 95)
- [104] N. H. Lu and B. A. Eisenstein. Detection of Weak Signals in Non-Gaussian Noise. *IEEE Trans. on Information Theory*, 27(6):755–771, Nov 1981. (p. 69)
- [105] M. Maes. Twin Peaks: The Histogram Attack on Fixed Depth Image Watermarks. In *Proc. 2nd Workshop on Information Hiding* [4], pages 290–305. (p. 22)
- [106] J. Magarey and A. Dick. Multiresolution Stereo Image Matching Using Complex Wavelets. In *Proc. of 14th International Conference on Pattern Recognition*, volume 1, pages 4–7, Brisbane, Australia, Aug 1998. (p. 92)
- [107] J. F. A. Magarey and N. G. Kingsbury. Motion estimation using a complex-valued wavelet transform. *IEEE Trans. on Signal Processing, special issue on wavelets and filter banks*, 46(4):1069–84, Apr 1998. (pp. 83, 86)
- [108] S. G. Mallat. A theory for multiresolution signal decomposition: The wavelet representation. *IEEE Trans. on Pattern and Machine Intelligence*, 11(7):674–693, Jul 1989. (pp. 25, 78)
- [109] A. B. Martinez, P. F. Swaszek, and J. B. Thomas. Locally Optimal Detection in Multivariate Non-Gaussian Noise. *IEEE Trans. on Information Theory*, 30(6):815–822, Nov 1984. (p. 69)
- [110] A. Mayache, T. Eude, and H. Cherifi. A Comparison of Image Quality Models and Metrics Based on Human Visual Sensitivity. In *Proc. of the IEEE International Conference on Image Processing* [7]. (pp. 37, 39, 135, 136)
- [111] P. Meerwald and A. Uhl. Watermark security via wavelet filter parametrization. In *Proc. of International Conference on Image Processing* [13], pages 1027–1030. (p. 15)
- [112] N. Memon and P. W. Wong. A Buyer-Seller Watermarking Protocol. In *Proc. of IEEE Workshop on Multimedia Signal Processing* [6]. (p. 6)
- [113] F. C. Mintzer, L. E. Boyle, A. N. Cazes, B. S. Christian, S. C. Cox, F. P. Giordano, H. M. Gladney, J. C. Lee, M. L. Kelmanson, A. C. Lirani, K. A. Magerlein, A. M. B. Pavani, and F. Schiattarella. Toward on-line worldwide

- access to Vatican library materials. *IBM Journal of Research & Development*, 40(2), 1995. (p. 3)
- [114] P. Moulin. The role of information theory in watermarking and its application to image watermarking. *Signal Processing*, 81(6):1121–1139, Jun 2001. (p. 2)
- [115] P. Moulin and J. A. O’Sullivan. Information-Theoretic Analysis of Information Hiding. <http://www.ifp.uiuc.edu/~moulin/paper.html>, Jan 2001. (p. 106)
- [116] J. J. K. Ó Ruanaidh, W. J. Dowling, and F. M. Boland. Phase Watermarking of Digital Images. In *Proc. of International Conference on Image Processing* [1], pages 239–242. (pp. 14, 20)
- [117] J. J. K. Ó Ruanaidh and T. Pun. Rotation, Scale and Translation Invariant Digital Image Watermarking. In *Proc. of International Conference on Image Processing* [2]. (p. 15)
- [118] E. Peli. Contrast of Complex Images. *Journal of Optical Society of America A*, 7(10):2032–2040, Oct 1990. (p. 32)
- [119] W. B. Pennebaker and J. L. Mitchell. *JPEG still image data compression standard*. van Nostrand Reinhold, New York, 1993. (p. 2)
- [120] S. Pereira, J. J. K. Ó Ruanaidh, F. Deguillaume, G. Csurka, and T. Pun. Template Based Recovery of Fourier-Based Watermarks Using Log-Polar and Log-Log Maps. In *IEEE International Conference on Multimedia Computing and Systems*, Jun 1999. (pp. 21, 81, 95, 96)
- [121] S. Pereira and T. Pun. An Iterative Template Matching Algorithm Using the Chirp-Z Transform for Digital Image Watermarking. *Pattern Recognition*, 33(1):173–175, Jan 2000. (p. 96)
- [122] S. Pereira, S. Voloshynovskiy, M. Madueño, S. Marchand-Maillet, and T. Pun. Second generation benchmarking and application oriented evaluation. In *4th Information Hiding Workshop* [12]. <http://watermarking.unige.ch/Checkmark/>. (pp. 23, 100)
- [123] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on Copyright Marking Systems. In *Proc. 2nd Workshop on Information Hiding* [4], pages 218–238. (pp. 23, 81)

- [124] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding - a survey. *Proc. of the IEEE Special Issue on Identification and Protection of Multimedia Information*, 87(7):1062–1078, Jul 1999. (p. 2)
- [125] F. A. P. Petitcolas and S. Katzenbeisser, editors. *Information hiding techniques for steganography and digital watermarking*. Artech House, Dec 1999. (p. 2)
- [126] I. Pitas. A Method for Signature Casting on Digital Images. In *Proc. of International Conference on Image Processing* [1], pages 215–218. (p. 13)
- [127] A. Piva, M. Barni, F. Bartolini, and V. Cappellini. DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image. In *Proc. of International Conference on Image Processing* [2], pages 520–523. (p. 14)
- [128] A. Piva, M. Barni, F. Bartolini, and V. Cappellini. Threshold Selection for Correlation-based Watermark Detection. In *Proc. of COST 254 Workshop on Intelligent Communications*, pages 67–72, LAquila, Italy, Jun 1998. (p. 21)
- [129] C. I. Podilchuk and W. Zeng. Watermarking of the JPEG Bitstream. In *Proc. of CISST International Conference*, 1997. (p. 16)
- [130] H. V. Poor. *An Introduction to Signal Detection and Estimation*. Springer Texts in Electrical Engineering. Springer, 2nd edition, 1998. (pp. 21, 69)
- [131] J. Puate and F. Jordan. Using fractal compression scheme to embed a digital signature into an image. In *Proc. of SPIE*, volume 2915, 1996. (p. 15)
- [132] L. Qiao and K. Nahrstedt. Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer’s Rights. *Journal of Visual Communication and Image Representation*, 9(3):194–210, Sep 1998. (p. 7)
- [133] M. Ramkumar and A. N. Akansu. Theoretical capacity measures for data hiding in compressed images. In *Proc. of SPIE Symposium on Voice, Video and Data Communication (VV-06)*, volume 3528, Nov 1998. (p. 76)
- [134] M. Ramkumar, A. N. Akansu, and A. Alatan. On the choice of transforms for data hiding in compressed video. In *Proc. of IEEE Intl. Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3049–3052, Phoenix, AZ, mar 1999. (p. 76)

- [135] M. Ramkumar, A. N. Akansu, and A. A. Alatin. A robust data hiding scheme for images using dft. In *Proc. of International Conference on Image Processing* [10]. (p. 14)
- [136] C. Rey and J.-L. Dugelay. Blind detection of malicious alterations on still images using robust watermarks. In *Proc. of IEE Seminar on Secure Images and Image Authentication*, Apr 2000. (p. 7)
- [137] V. Solachidis and I. Pitas. Circularly Symmetric Water Embedding in 2-D DFT Domain. In *Proc. of International Conference on Acoustics, Speech and Signal Processing (ICASSP)* [9]. (pp. 14, 84)
- [138] J. K. Su, J. J. Eggers, and B. Girod. Analysis of Digital Watermarks Subjected to Optimum Linearizing and Additive Noise. *Signal Processing*, 81:1141–1175, Jun 2001. (pp. 97, 99)
- [139] J. K. Su and B. Girod. Power spectrum condition for energy-efficient watermarking. In *Proc. of International Conference on Image Processing* [10]. (p. 98)
- [140] J. K. Su and B. Girod. Fundamental Performance Limits of Power-Spectrum Condition-Compliant Watermarks. In Wong and Delp [162]. (pp. 97, 98, 99, 100)
- [141] M. D. Swanson, B. Zhu, and A. H. Tewfik. Transparent Robust Image Watermarking. In *Proc. of International Conference on Image Processing* [1], pages 211–214. (p. 14)
- [142] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding Secret Information Into a Dithered Multilevel Image. In *Proc. of the 1990 IEEE Military Communications Conference*, pages 216–220, 1990. (p. 2)
- [143] D. B. H. Tay and N. G. Kingsbury. Flexible design of multidimensional perfect reconstruction FIR-2 band filters using transformations of variables. *IEEE Trans. on Image Processing*, 2(4):466–480, 1993. (p. 26)
- [144] A. Z. Tirkel, G. A. Rankin, R. M. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne. Electronic watermark. In *Proc. DICTA 93*, pages 666–673, Macquarie University, Australia, 1993. (p. 2)
- [145] M. Unser. Splines - a perfect fit for signal and image processing. *IEEE Signal Processing Magazine*, 16(6):22–38, Nov 1999. (p. 94)

- [146] M. C. Valenti. An Introduction to Turbo Codes .
<http://www.csee.wvu.edu/~mvalenti/documents/valenti1996.pdf>, May 1996.
(p. 19)
- [147] C. J. van den Branden Lambrecht. Perceptual Quality Metric for Digitally Coded Color Images. In *Proc. of European Conference on Signal Processing (Eusipco)*, Lausanne, Switzerland, 1996. (pp. 34, 35)
- [148] S. Voloshynovskiy, F. Deguillaume, S. Pereira, and T. Pun. Optimal adaptive diversity watermarking with channel state estimation. In Wong and Delp [163]. (p. 103)
- [149] S. Voloshynovskiy, F. Deguillaume, and T. Pun. Content Adaptive Watermarking Based on a Stochastic Multiresolution Image Modelling. In *Proc. of European Conference on Signal Processing* [11]. (pp. 21, 96)
- [150] S. Voloshynovskiy, F. Deguillaume, and T. Pun. Multibit Digital Watermarking Robust Against Local Nonlinear Geometrical Distortions. In *Proc. of International Conference on Image Processing* [13], pages 999–1002. (p. 133)
- [151] S. Voloshynovskiy, A. Herrigel, N. Baumgaertner, and T. Pun. A stochastic approach to content adaptive digital image watermarking. In *Proc. 3rd Workshop on Information Hiding* [8], pages 211–236. (p. 97)
- [152] S. Voloshynovskiy, S. Pereira, A. Herrigel, N. Baumgaertner, and T. Pun. A generalized watermark attack based on stochastic watermark estimation and perceptual remodulation. In Wong and Delp [162]. (pp. 22, 98)
- [153] S. Voloshynovskiy, S. Pereira, T. Pun, J. J. Eggers, and J. K. Su. Attacks on Digital Watermarks: Classification, Estimation-based Attacks and Benchmarks. *IEEE Communications Magazine (Special Issue on Digital watermarking for copyright protection: a communications perspective)*, 39(8):118–127, 2001. M. Barni, F. Bartolini, I.J. Cox, J. Hernandez, F. Pérez-Gonzalez, Guest Eds. (p. 22)
- [154] G. Voyatzis and I. Pitas. Digital Images Watermarking Using Mixing Systems. In *Computer and Graphics*, volume 22, pages 405–416, 1998. (p. 13)
- [155] H.-J. Wang and C.-C. J. Kuo. High fidelity image compression with multi-threshold wavelet coding (MTWC). In *SPIE's Annual Meeting - Application of Digital Image Processing XX*, Aug 1997. (p. 13)

- [156] H.-J. Wang and C.-C. J. Kuo. Image Protection via Watermarking on Perceptually Significant Wavelet Coefficients. In *Proc. of IEEE Workshop on Multimedia Signal Processing* [6]. (pp. 13, 15)
- [157] A. B. Watson. The Cortex Transform: Rapid Computation of Simulated Neural Images. *Computer Vision, Graphics, and Image Processing*, 39:311–327, 1987. (pp. 31, 32)
- [158] A. B. Watson. DCT quantization metrics visually optimized for individual images. In *Proc. of SPIE Human Vision, Visual Processing and Digital Display IV*, volume 1913, 1993. (pp. 14, 37, 39, 135, 136)
- [159] T. S. Wilson, S. K. Rogers, and L. B. Myers. Perceptual-Based Hyperspectral Image Fusion Using Multiresolution Analysis. *Optical Engineering*, 34:3154–3164, 1995. (p. 14)
- [160] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp. The effect of matching watermark and compression transforms in compressed color images. In *Proc. of the IEEE International Conference on Image Processing* [7]. (p. 74)
- [161] P. W. Wong and E. J. Delp, editors. *Proc. of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging*, volume 3657, San Jose, CA, Jan 1999. (pp. 151, 152, 153, 155, 157)
- [162] P. W. Wong and E. J. Delp, editors. *Proc. of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging II*, volume 3971, San Jose, CA, Jan 2000. (pp. 151, 157, 161, 162)
- [163] P. W. Wong and E. J. Delp, editors. *Proc. of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging III*, volume 4314, San Jose, CA, Jan 2001. (pp. 150, 153, 154, 158, 162)
- [164] M. Xia and B. Liu. Effect of JPEG Compression on Image Watermark Detection. In *Proc. of International Conference on Acoustics, Speech and Signal Processing*, Salt Lake City, UT, May 2001. (p. 76)
- [165] X.-G. Xia, C. G. Boncelet, and G. R. Arce. Wavelet transform based watermark for digital images. In *Optical Express* [3], pages 497–511. (pp. 15, 95)
- [166] L. Xie and G. R. Arce. Joint Wavelet Compression and Authentication Watermarking. In *Proc. of the IEEE International Conference on Image Processing* [7]. (p. 95)