LOW-COMPLEXITY METHODS FOR IMAGE AND VIDEO WATERMARKING

by

Lino Evgueni Coria Mendoza

B.Eng., Instituto Tecnológico de Morelia, 1996
M.Eng., McMaster University, 1998

A THESIS SUBMITTED IN PARTIAL FULFILLMENT OF
THE REQUIREMENTS FOR THE DEGREE OF

DOCTOR OF PHILOSOPHY

in

The Faculty of Graduate Studies

(Electrical and Computer Engineering)

THE UNIVERSITY OF BRITISH COLUMBIA
(Vancouver)

March 2008

ABSTRACT

For digital media, the risk of piracy is aggravated by the ease to copy and distribute the content. Watermarking has become the technology of choice for discouraging people from creating illegal copies of digital content. Watermarking is the practice of imperceptibly altering the media content by embedding a message, which can be used to identify the owner of that content. A watermark message can also be a set of instructions for the display equipment, providing information about the content's usage restrictions. Several applications are considered and three watermarking solutions are provided.

First, applications such as owner identification, proof of ownership, and digital fingerprinting are considered and a fast content-dependent image watermarking method is proposed. The scheme offers a high degree of robustness against distortions, mainly additive noise, scaling, low-pass filtering, and lossy compression. This method also requires a small amount of computations. The method generates a set of evenly distributed codewords that are constructed via an iterative algorithm. Every message bit is represented by one of these codewords and is then embedded in one of the image's 8×8 pixel blocks. The information in that particular block is used in the embedding so as to ensure robustness and image fidelity.

Two watermarking schemes designed to prevent theatre camcorder piracy are also presented. In these methods, the video is watermarked so that its display is not permitted if a compliant video player detects the watermark. A watermark that is robust to

geometric distortions (rotation, scaling, cropping) and lossy compression is required in order to block access to media content that has been recorded with a camera inside a movie theatre. The proposed algorithms take advantage of the properties of the dual-tree complex wavelet transform (DT CWT). This transform offers the advantages of both the regular and the complex wavelets (perfect reconstruction, approximate shift invariance and good directional selectivity). Our methods use these characteristics to create watermarks that are robust to geometric distortions and lossy compression. The proposed schemes are simple to implement and outperform comparable methods when tested against geometric distortions.

Keywords: complex wavelets, computational complexity, copyright protection, discrete cosine transform, high capacity, informed coding, informed embedding, lossy compression, piracy, spherical codes, watermarking.

TABLE OF CONTENTS

Abstract ..... **ii**

Table of Contents ..... **iv**

List of Tables ..... **vi**

List of Figures ..... **viii**

List of Acronyms ..... **x**

Acknowledgements ..... **xi**

Dedication ..... **xiv**

Co-Authorship Statement ..... **xv**

Chapter 1: Introduction and Overview **1**

1.1 Introduction 1
1.2 Watermarking Overview 4
1.2.1 Basic watermarking scheme 4
1.2.2 Content-dependent watermarking 6
1.2.3 Some applications and requirements 7
1.3 Literature Review 9
1.3.1 Compressed domain watermarking 9
1.3.2 Uncompressed domain watermarking 10
1.3.3 Informed coding watermarking 12
1.3.4 Specific challenges for video watermarking 14
1.4 Objectives 20
1.5 References 26

Chapter 2: A Fast High-Capacity Informed Coding Scheme for Image Watermarking **30**

2.1 Introduction 30
2.2 Overview of Content-Dependent Watermarking 33
2.3 Proposed Watermarking Scheme 38
2.3.1 Generating the codewords 40
2.3.2 Embedding the watermark 47
2.3.3 Decoding the watermark 50
2.4 Performance Evaluation 51
2.4.1 Parameter setting 51
2.4.2 Computational time 52
2.4.3 Robustness to common attacks 52
2.5 Conclusion 57

2.6 References 63

Chapter 3: A Complex-Wavelet Based Video Watermarking Scheme for Playback Control 65

3.1 Introduction 65
3.2 Proposed Method 67
3.2.1 Brief introduction to the Dual-Tree Complex Wavelet Transform 67
3.2.2 Creating the watermark 70
3.2.3 Embedding the watermark 72
3.2.4 Detecting the watermark 73
3.3 Experimental Results 74
3.3.1 Frame scaling and cropping 77
3.3.2 Frame rotation 78
3.3.3 Compression 78
3.3.4 Joint attack 78
3.4 Conclusion 79
3.5 References 82

Chapter 4: A Video Watermarking Scheme Based on the Dual-Tree Complex Wavelet Transform 83

4.1 Introduction 83
4.2 Proposed Method 88
4.2.1 Creating the watermark 88
4.2.2 Embedding the watermark 90
4.2.3 Decoding the watermark 95
4.3 Experimental Results 97
4.3.1 Frame scaling and cropping 100
4.3.2 Frame rotation 102
4.3.3 Additive noise 104
4.3.4 Compression 105
4.3.5 Joint attack 105
4.4 Conclusion 106
4.5 References 109

Chapter 5: Conclusions 111

5.1 Overall Significance of the Research 111
5.2 Potential Applications of the Research Findings 112
5.3 Discussion and Conclusions 114
5.3.1 Summary of contributions 114
5.3.2 Three common advantages of the proposed schemes 116
5.4 Comments on Future Research 118
5.4.1 Error-correcting codes 118
5.4.2 Better perceptual models 119
5.4.3 Temporal synchronization 119
5.5 References 120

LIST OF TABLES

Table 2.1 Image fidelity after low-pass filtering: filter width vs. average PSNR. 54

Table 2.2 Image fidelity after intensity scaling: scaling factor vs. average PSNR. 55

Table 2.3 Image fidelity after Gaussian noise: standard deviation vs. average PSNR..... 56

Table 2.4 Image fidelity after JPEG compression: quality factor vs. average PSNR. 56

Table 2.5 Robustness to common attacks. For every test, the method with the best performance in terms of BER and MER is indicated. 57

Table 3.1 Strength of the decoded watermark values obtained when using the DT CWT method. The tests involve non-watermarked (NW) videos, as well as watermarked (W) sequences that go through several attacks such as scaling and cropping (S+C), rotation (R), lossy H.264 compression (H264) and, finally, a joint attack (Joint). 80

Table 3.2 Strength of the decoded watermark values obtained when using the DWT method. The tests involve non-watermarked (NW) videos, as well as watermarked (W) sequences that go through several attacks such as scaling and cropping (S+C), rotation (R), lossy H.264 compression (H264) and, finally, a joint attack (Joint). 80

Table 4.1 Comparison of normalized correlation values obtained by the three watermarking methods: DT CWT, DWT1 and DWT2. 100

Table 4.2 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to scaling (by 5%, 10% and 15%) and cropping. 101

Table 4.3 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to rotation (by 3^o, 6^o and 9^o) and cropping. 103

Table 4.4 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2.

Watermarked and Non-Watermarked sequences are subjected to lossy compression (H.264 with a QF of 15) and also to a joint attack (H.264 compression with a QF of 15, scaling up by 5%, rotating by 5º, and cropping back to QCIF)..... 107

Table 4.5 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to a joint attack (additive noise with a standard deviation of 5, rotating by 5º, scaling up by 5%, cropping back to QCIF, and H.264 compression). 108

LIST OF FIGURES

Figure 1.1 A blind watermarking system. 5
Figure 1.2 A content-dependent watermarking system. 8
Figure 1.3 Video piracy points 10
Figure 2.1 A blind watermarking system. 35
Figure 2.2 A content-dependent watermarking system. 37
Figure 2.3 The DCT is applied to an 8 × 8 pixel block. Shaded coefficients indicate the AC terms used for constructing the coefficient vector of length L (in this case, $L = 16$) 39
Figure 2.4 (a) A desirable outcome of our codeword generation algorithm. (b) An undesirable outcome of our codeword generation algorithm: there are clusters with codewords that belong to the same subset. 42
Figure 2.5 Block diagram of our codeword generation algorithm. 44
Figure 2.6 Step 1: Initializing codewords from subset A. 46
Figure 2.7 Step 3: Creating subset B: case when (a) $L = 2$; (b) $L = 3$. 47
Figure 2.8 Step 4: Even distribution of all the codewords ($L = 3$). 48
Figure 2.9 Robustness to low-pass filtering: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired. 59
Figure 2.10 Robustness to intensity scaling: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired. 60
Figure 2.11 Robustness to Gaussian noise: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired. 61
Figure 2.12 Robustness to JPEG compression: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired. 62

Figure 3.1 Basic configuration of the dual-tree filtering approach used to obtain the DT CWT coefficients (for a real one-dimensional signal $x[n]$). 69

Figure 3.2 Typical impulse responses of the high pass decimation filters for each of the filter trees. 69

Figure 3.3 (a) Two dimensional impulse responses of the reconstruction filters in a DT CWT; (b) Structure of the DT CWT coefficients for a four-level decomposition. 71

Figure 3.4 The watermark embedding process. 76

Figure 3.5 The watermark detection process. 77

Figure 3.6 A QCIF video frame of the sequence *News*: (a) watermarked with the DT CWT method, (b) after scaling and cropping, (c) after rotation and (d) after a joint attack. 81

Figure 4.1 Structure of the DT CWT coefficients for a four level decomposition: (a) For each level there are six subbands that correspond to the output of six directional filters oriented at angles of $\pm 15^{o}$, $\pm 45^{o}$, and $\pm 75^{o}$; (b) The notation employed for the proposed method. 89

Figure 4.2 The level-1 DT CWT transform is applied to the watermark w. The coefficients of W_{H1}, ... W_{H6} are the data to be embedded in the video frames. 90

Figure 4.3 Construction of the masks for coefficients from levels 3 and 4 are obtained from level 2 subbands. 93

Figure 4.4 After 300 frames, the decoder detects a high watermark strength value for the watermarked Suzie sequence while the strength decoded for the non-watermarked Suzie video is close to zero. 99

Figure 4.5 A watermarked frame of the sequence *Suzie* is scaled and then cropped: (a) 5%, (b) 10% and (c) 15% scaling. 102

Figure 4.6 A watermarked frame of the sequence *Suzie* is rotated and then cropped: (a) 3^{o}, (b) 6^{o} and (c) 9^{o} rotation. 104

Figure 4.7 (a) A watermarked frame of the sequence *Suzie*. (b) The same frame is subjected to a joint attack (rotating by 5^{o}, scaling up by 5%, cropping, adding noise and using H.264 compression). 106

LIST OF ACRONYMS

BER	Bit Error Rate
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DRM	Digital Rights Management
DT CWT	Dual-Tree Complex Wavelet Transform
DVD	Digital Versatile Disc
DWT	Discrete Wavelet Transform
HVS	Human Visual System
IDCT	Inverse Discrete Cosine Transform
MER	Message Error Rate
NTSC	National Television System Committee
PAL	Phase Alternating Line
PSNR	Peak Signal to Noise Ratio
QCIF	Quarter Common Intermediate Format
QF	Quality Factor
QIM	Quantization Index Modulation

ACKNOWLEDGEMENTS

Working on a PhD feels sometimes like a very lonely task. However, this project would not have been possible without the assistance and support from a large number of people. I want to express my gratitude to them in the following lines.

First, I would like to thank the people from CONACYT, Mexico's National Council for Science and Technology. CONACYT provided me with a scholarship that covered my living expenses and healthcare needs. The staff members of this institution always provided prompt and polite answers to all my questions.

I want to show my most sincere appreciation to everybody at ITESO University in Tlaquepaque, Jalisco, Mexico. Their support has been tremendous and I am in debt with them. Mtra. Gabriela Ortiz Michel was Head of the Department of Electronics, Systems and Informatics (DESI) in 2002 when I first expressed my interest in pursuing a doctorate degree. She was always enthusiastic and supportive regarding this idea. Two years later, Mtro. Jorge Arturo Pardiñas Mir became Head of DESI. He made life easier for me at all times. His support was particularly crucial on the last stages of my PhD program. I have to thank Jorge on a more personal level since he was always available as a friend. I also wish to thank Mtro. Carlos Eduardo Luna Cortés, ITESO's Academic Director-General until 2005 and his successor, Dr. Francisco Morfín Otero. I got nothing but support and encouragement from them. My heartfelt gratitude goes to Ing. Héctor Manuel Acuña Nogueira, S.J., Rector of ITESO. None of this would have been possible without his backing. In addition, I thank Mtro. Francisco Javier Haro del Real and Mrs. Rosa María

Álvarez Bernal from human resources. Thank you as well to all my colleagues from DESI.

I thank Dr. José Luis Naredo Villagrán (CINVESTAV), Dr. José Ernesto Rayas Sánchez (ITESO), and Dr. James P. Reilly (McMaster University) for their kind reference letters.

I cannot stress enough how thankful I am to my supervisors, Dr. Rabab K. Ward and Dr. Panos Nasiopoulos. Besides being a talented and well-respected researcher, Dr. Ward is a kind and caring person. I am amazed at her great ability to see 'the forest and the trees.' Her comments and suggestions were always relevant to my work. Moreover, I will never forget her advice: that I should quit engineering and become a comedian. Dr. Panos Nasiopoulos was always involved with my work and understood every detail of it. In addition, he was available at all times. We had long and memorable conversations while drinking coffee. He showed me that there is more to a PhD than just doing research: the way you present your ideas to the world is very important. He also shared with me his philosophy about many aspects of life and, finally, disproved the myth that engineers cannot dress properly.

I want to express my genuine gratitude to Dr. Mark Pickering from the University of New South Wales. He brought complex wavelets to my attention and introduced the idea of using them for watermarking purposes. He is an extremely creative person and his involvement in our research proved particularly fruitful.

Thank you also to the members of my lab. I was very lucky to be surrounded by nice and bright people. Thank you Hassan Mansour, Adarsh Golikeri, Victor Sánchez, Sergio Infante, Matthias Von Dem Knesebeck, Zicong Mai, Mahsa Pourazad, Colin

Doutre, Di Xu, Ashfiqua Tahseen Connie, and Dr. Yaser P. Fallah. A very special thank you to Dr. Mehrdad Fatourechi and Qiang Tang for their help and friendship.

In addition, I want to thank my parents for their love and support. Thank you also to my sister Talía for her encouragement and friendship. I am also grateful to my grandma Mary for those nice conversations over the phone.

Finally, I would like to thank the two most important people in my life. One of them is my wife, Marcela: the sweetest girl in the entire universe (and I am not exaggerating). The second person is someone I have yet to meet: a tiny creature that is growing inside Marcela's belly. Marcela and *bebé*: I love you.

To Marcela and our child

CO-AUTHORSHIP STATEMENT

This thesis presents research conducted by Lino Evgueni Coria Mendoza, in collaboration with Dr. Panos Nasiopoulos, Dr. Rabab Ward, and Dr. Mark Pickering.

Manuscript 1: *A Fast High-Capacity Informed Coding Scheme for Image Watermarking.* This manuscript was the work of Lino Evgueni Coria Mendoza, who received suggestions and feedback from his supervisors, Dr. Ward and Dr. Nasiopoulos.

Manuscript 2: *A Complex-Wavelet Based Video Watermarking Scheme for Playback Control.* The algorithm was proposed by Dr. Mark Pickering. Lino Evgueni Coria Mendoza was responsible for the performance evaluations, interpretation of the results and writing the manuscript. Dr. Nasiopoulos and Dr. Ward provided guidance and editorial input into the creation of the manuscript.

Manuscript 3: *A Video Watermarking Scheme Based on the Dual-Tree Complex Wavelet Transform.* The primary investigator and author of the manuscript was Lino Evgueni Coria Mendoza, who conducted the research with guidance from Drs. Rabab Ward, Panos Nasiopoulos, and Mark Pickering.

The first and last chapters of the thesis were written by Lino Evgueni Coria Mendoza, with editing assistance and consultation from Dr. Rabab Ward and Dr. Panos Nasiopoulos.

CHAPTER 1: INTRODUCTION AND OVERVIEW

1.1 Introduction

Many aspects of our lives have been significantly transformed by the ease to create and distribute digital content. From the way we communicate and do business to the way we learn and have fun, digital data have altered the means to interact with the world. We can, for instance, watch an online video describing an important incident that took place only a few minutes ago in a remote part of the world. Or we can automatically download podcasts from the Internet and into our portable mp3 player so we can listen to the content whenever we choose to. Moreover, emerging artists can share their work with anyone interested in it without the need of a publisher, a theatre or a manager. We can also buy a DVD that features a restored movie from the 1920's that, thanks to digital technology, looks better than ever.

Unlike analogue media, digital content does not degrade over time. Furthermore, digital data are easier to manipulate. There are many inexpensive tools available which allow non-experts to, among other things, copy data, alter images, edit video clips and mix music tracks. In addition, the Internet allows users to upload the content and, potentially, share it with millions of people. This new possibility brings many benefits to a large number of people since it provides them with new ways to communicate and express their ideas. Unfortunately, there is a major drawback to this great flexibility offered by digital information: copyright owners lose control of their content after distributing it as it can then be easily copied and circulated without their consent. This

problem is usually referred to as piracy and has a negative impact on the people involved, either creatively or financially, in the production of the digital data. First of all, it affects the creators of the content (filmmakers, musicians, photographers) since their work can be modified without their approval. Secondly, there is no financial revenue for producers of content when it becomes freely distributed over the World Wide Web. As a third issue, piracy affects the global economy. Legitimate jobs might be lost since the distribution of illegal copies of movies and songs makes it more difficult to profit from the creation, distribution and exhibition of licit digital content.

Some legal measures that deter piracy have been considered, although not without controversy [1, 2]. The piracy dilemma is mainly a technology related issue. Digital Rights Management (DRM) is a series of systems designed to protect digital assets and control the use and distribution of this content [3, 4]. A successful DRM implementation involves, among other things, persistent content protection. In other words, protection has to stay within each copy of the content. For instance, a movie can be purchased over the Internet and delivered in a secure manner using cryptographic mechanisms. But once the movie has been downloaded, the recipient can create copies of the content and upload an unrestricted version back to the Internet. Users from around the world can then freely download the video file without losing any picture quality. To avoid this situation, restrictions of the content usage rights have to be maintained after the content has been delivered to the end user. The technology to accomplish this task is known as digital watermarking.

Digital watermarking is the practice of hiding a message about an image, audio clip, video clip, or other media content within the content itself [5]. Watermarking can be

also viewed as the practice of imperceptibly altering the content in order to embed a message about that content. This hidden message can be, for instance, some text that provides information about the copyright owner of the work in question. It can also be an image (e.g., a logo from a company). A watermark message can also be a set of instructions for the image or video decoder or display equipment, providing information about the content's usage restrictions. Ideally, the equipment in charge of displaying the content should be able to decode and interpret the hidden information correctly.

Watermarks need to be imperceptible (invisible in the case of images, inaudible in the case of audio). The user should not be able to notice that the movie she is watching or that the song that he is listening to has been altered with a watermark. However, the main challenge is to design watermarking methods that are capable of retrieving the hidden message even after the watermarked content has been distorted in some way. Sometimes, these changes to the content are necessary. For instance, lossy compression is unavoidable for many applications that involve large amounts of data. Occasionally, the content is altered accidentally, such when it is distorted by noise. On the other hand, some changes to the content are a result of a hostile attack, i.e. an intentional purpose of destroying the watermark while keeping the content's significant information.

Each type of digital content (audio, images and video) represents different challenges for the watermarking systems. For instance compression algorithms, although having the same general principles, are different for every particular type of media. A watermarking algorithm that performs adequately on an image might not be successful for audio files. And even though video is a succession of still images, there are a number of challenges that are particular to this type of data. Therefore, both the application and

the nature of the content must be carefully considered when designing watermarking algorithms. In the rest of this thesis we will focus on watermarking of visual data, providing a particular emphasis on video watermarking.

1.2 Watermarking Overview

1.2.1 Basic watermarking scheme

The basic frequency-domain watermarking system is presented in Fig. 1.1. The binary input watermark message *m* (which can be an image, text or other data) is encoded via a watermarking process. The watermark is inserted in selected frequency coefficients of the original content. Two codewords, one representing the bit 0 of the message and the other representing the bit 1 are generated. A pseudorandom number generator using a seed (starting point), which is a number that is only known to the content owner, is employed. This secret number is referred to as a *watermark key*. Each codeword is a sequence of pseudorandom numbers that can only be later recovered if the watermark key is provided. Each watermark message bit (0 or 1) is replaced by a codeword, forming an array called **w**. The original content $\mathbf{c_o}$ goes through a frequency transformation and **w** is added to selected frequency coefficients. The inverse frequency transformation converts the coefficients to pixel values, resulting in a watermarked image $\mathbf{c_w}$. This watermark embedding process is referred to as *blind embedding* because the watermark is added to the image without using any information about the content. In order to retrieve the watermark, the decoder computes the linear correlation between the watermarked content and the codewords representing bits 0 and 1. Based on the highest correlation values, the array **w**' can be constructed and thus, the decoded watermark message *m*' can be

obtained. The decoding process is also blind since the original (unwatermarked) content $\mathbf{c_o}$ is not taken into account when retrieving the watermark message.

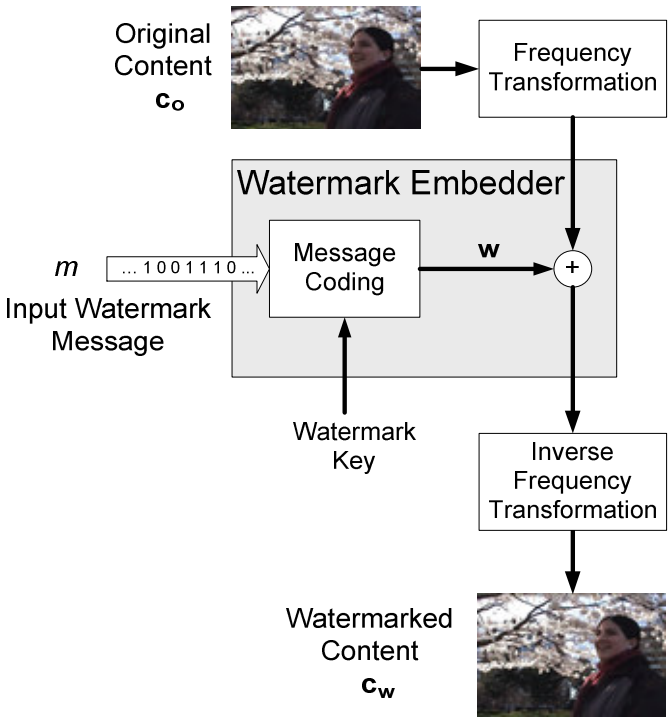


Figure 1.1 A blind watermarking system.

A blind decoder is desirable for most watermarking applications since distributing the original content to the detector (or detectors) would defeat the purpose of the watermarking process. However, ignoring the original content at the embedder may not produce the best benefits. The use of information derived from the original content could be exploited to build watermark embedding algorithms that result in higher robustness or image fidelity. It also enables control over the strength with which the watermark is

embedded. The watermark correlation value could, for instance, be monitored and modified at the embedder in order to ensure a more effective watermarking method.

1.2.2 Content-dependent watermarking

More efficient watermarking algorithms can be constructed if information from the original content is used *before* the codewords are added. For instance, the information derived from the original content can be utilized to improve the choice of codewords. Instead of relying on a one to one mapping (i.e., every message bit is represented by only one codeword), a single message bit could correspond to one of several codewords [5]. Based on the image content, the codeword that results in the least distortion to the content is then chosen amongst these codewords to represent the bit to be embedded. This strategy is known as *informed coding* and has been inspired by the theoretical results obtained by Costa in [6].

Information from the host image can also be employed to control the strength with which the watermark is added to the content. The watermark can be adjusted so that the embedding algorithm offers a predetermined level of robustness to common distortions. By taking the image content into consideration, the embedder can ensure that the correct message is later extracted by the decoder from the watermarked image (assuming no attacks). This process is known as *informed embedding* and, by combining it with informed coding, it is possible to improve the performance of watermarking schemes [5].

Fig. 1.2 shows a watermarking system that takes advantage of both informed embedding and informed coding. In this figure, the input watermark message m is

represented as a sequence of 0's and 1's that are to be embedded in the original content $\mathbf{c_o}$. To embed the message, the content $\mathbf{c_o}$ is divided into non-overlapping 8×8 pixel blocks and the Discrete Cosine Transform (DCT) is applied to each one of these blocks to construct the array $\mathbf{C_o}$. Every message bit is then embedded in a different 8×8 block by modifying some of the block's frequency coefficients. However, instead of having one codeword corresponding to all the 0 bits and another codeword corresponding to all the 1 bits as in the traditional scheme, the watermark key is used to generate two subsets of pseudorandom codewords, each corresponding to each of these bits. As in the traditional scheme, these codewords can be later recovered by the decoder, if the key is known. To embed one bit, the selected coefficients from an 8×8 block are compared against the whole subset of codewords corresponding to this bit. The codeword that is most similar to the block's coefficients (where the bit is to be hidden) is chosen. This codeword is not added to the content's coefficients. Instead, the coefficients are adjusted until they reach a predetermined correlation value with the codeword chosen to represent the embedded bit. This is repeated for every message bit and the result of this process is the array of coefficients $\mathbf{C_w}$, which after an inverse frequency transformation (IDCT) becomes the watermarked image $\mathbf{c_w}$.

1.2.3 Some applications and requirements

The inclusion of informed coding in a watermarking scheme makes it possible to attain a large watermark capacity. Capacity refers to the number of bits a watermarking scheme can encode within the content or within a unit of time. Large capacity is necessary for some purposes. However, other applications such as *copy control* [7] may only require a small message. A 1-bit watermark message, for instance, is enough to

embed one out of two possible messages into a video file: 'copy,' and 'do not copy.' In contrast, an application such as *television broadcast monitoring* [8] might require a few dozen message bits to identify the different commercials that are transmitted by a TV station [5]. The idea behind this application is to watermark television commercials so that advertisers can corroborate that the airtime they purchased from television broadcasters is actually employed to transmit their advertisements.

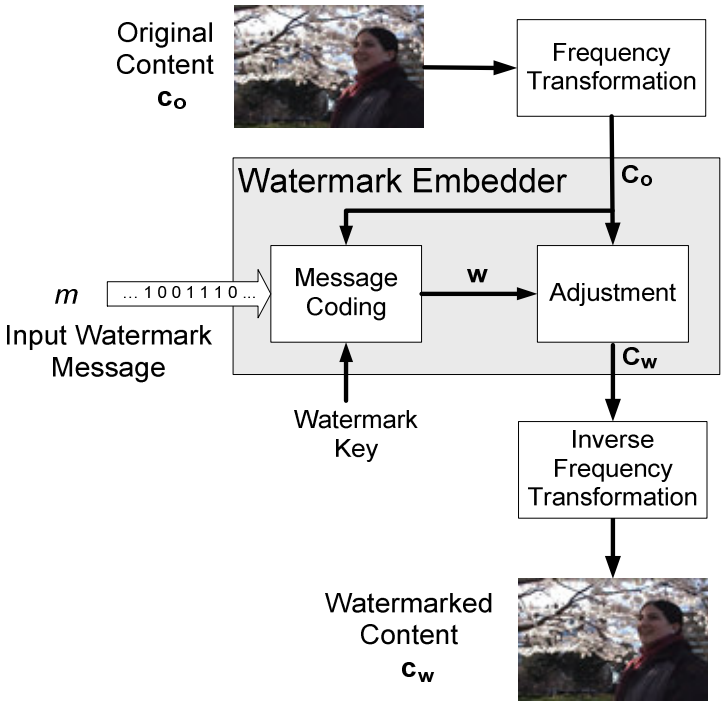


Figure 1.2 A content-dependent watermarking system.

The type of application determines the capacity requirements of the watermark; it also indicates what types of distortion are more likely to alter the watermarked content. Watermarking algorithms must then be designed with these types of distortion in mind. For instance, if a video file is available on a DVD, people might try to upload the content

to the Internet. In this case, distortion resulting from compression will take place and the watermarking scheme should be able to protect this content from lossy compression attacks. As another example, when video is displayed in a movie theatre, camcorder piracy becomes a concern. In this case, the watermark embedded in the content must be able to survive geometric distortions; this is because the movie recorded from the cinema's screen will be a rotated and cropped version of the original video.

1.3 Literature Review

Digital video watermarking is a challenging topic and a substantial amount of research has been devoted to it. Video piracy can take place at the different stages of the video distribution process [9]. As illustrated in Fig. 1.3, pirates may tamper with the hardware or software to capture decrypted data files. Another possibility is to use the screen capture software to obtain the decompressed data from the video buffers. Finally, less sophisticated pirates might just use camcorders to capture exhibited data from the screens.

1.3.1 Compressed domain watermarking

In order to protect the content at the first stage of the distribution process, embedding a watermark in the compressed domain seems to be the best option since most video files are stored in a compressed form. One of the earliest and most influential schemes of this kind is presented in [10]. In this work, as in the majority of the publications on this topic, the MPEG-2 video coding standard is employed. The watermark is embedded in the non-zero DCT coefficients of the encoded video. Another scheme [11] embeds the watermark only in the I-frames of a video sequence. This makes

the method resilient to P or B frame skipping. A similar algorithm where the message is embedded in the luminance component of the I-frames is presented in [12]. The watermark only affects the magnitude of the DCT coefficients. Each bit of the watermark message is embedded in a different macroblock. The watermarking schemes in [13-15] embed a message by modifying the higher magnitude motion vectors in MPEG video bitstreams.

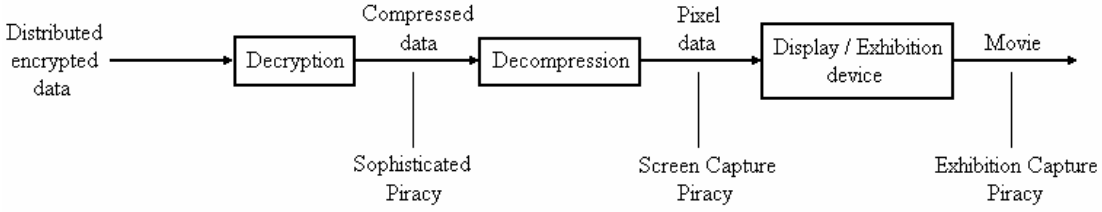


Figure 1.3 Video piracy points.

All these algorithms form effective piracy deterrents provided that the illegal copying of a video file takes place at the first stage of distribution. Also, these schemes are only effective when the video files are in the compressed domain. The success of retrieving the watermark is completely dependable on the video encoding method that has been used. If an attacker decodes the video and then encodes it again with a different coding standard, the watermark might be lost and illegal copies can be made without the possibility of tracking the original work [16].

1.3.2 Uncompressed domain watermarking

In order to avoid the re-encoding attack, the watermark can be embedded in the uncompressed domain. For uncompressed video, some techniques modify the spatial data

samples and spread the watermark energy over all the pixels in each frame [5, 17, 18]. Unfortunately, methods that hide information by directly modifying the pixel values are very vulnerable to attacks and common signal distortions.

Watermarks are more robust to distortions when they are embedded in the frequency domain. In this type of schemes, a frequency transformation is applied to the frames of the video sequence and the watermark is embedded by modifying a selected group of transform coefficients. This is commonly referred to as Spread *Spectrum* watermarking [5]. For watermarking applications, the transforms that are commonly used are the Discrete Cosine Transform (DCT) [9, 19] and the Discrete Wavelet Transform (DWT) [20-22]. Other frequency transformations have also been explored [23-26].

The best approach for embedding the watermark in the transform domain, is to hide the message in the lowest frequency coefficients. This has been demonstrated to improve the robustness of the watermark [9]. In [21] a DWT-based scheme is presented where the watermark is also embedded in the low frequency subband so as to obtain less perceptual distortion. However, the robustness of this method can be considerably improved by constructing a scheme that embeds a watermark using information from the host data.

A content-dependent watermark is presented in [27]. It is shown that robustness against statistical collusion can be obtained when a watermark is both content-dependent and spatially localized. Information from the original content is used to fix the level of robustness of the watermark, which is embedded in a set of sub-frames within each frame.

Another content-dependent approach that encodes the watermark in the lowest frequencies is presented in [9]. Although the essential requirements for a successful forensic watermark are described at great length, the proposed scheme is only mentioned in very general terms. The watermark is embedded in the spatiotemporal regions of the video sequence where the content's fidelity is not affected in a significant way by the embedding. The decoding process however is not blind, i.e., the original content is required in order to decode the watermark. Using the original content to recover a watermark involves a prohibitive number of computations. It also increases the risk of piracy since the people in charge of decoding the watermark, normally a third party, will have access to the unwatermarked work.

An adaptive embedding mechanism is presented in [18]. The watermark is embedded in the spatial domain in order to reduce the number of computations. By using an adaptive scheme, however, the algorithm becomes complex and a large number of computations are nevertheless required. The watermark is embedded in the middle bit planes of the luminance components of the uncompressed video frames, which results in a lack of robustness against valumetric scaling.

1.3.3 Informed coding watermarking

It has been shown [19, 28-32] that the capacity (length) of watermark messages can be increased if the information derived from the content is utilized during the embedding process. In addition, the fidelity and robustness of these content-dependent schemes have been successfully demonstrated. In these methods the message's bit 0 is represented by a codeword chosen from a predefined set and the bit 1 by a codeword chosen from another predetermined set. For each case, the codeword is chosen such that

it causes the least distortion to the content (i.e., the one that is most similar to the content). This strategy is known as informed coding.

Quantization Index Modulation (QIM) watermarking is a scheme that relies on informed coding [28]. In QIM schemes, the amplitude of a vector whose entries are pixels or frequency coefficients is quantized using a quantization lattice. While this approach provides a gain in the watermark capacity over other content-dependent schemes, QIM watermarking offers no robustness to valumetric scaling. A variation of this method has been proposed to deal with this lack of robustness in [30]. The trade-off, in this case, is to reduce the capacity of the watermark.

Another option for informed coding algorithms is to use spherical codewords along with correlation decoding. If all the codewords are located on the surface of a unit sphere, they will have the same energy level and, therefore, robustness to valumetric scaling can be achieved. One example of a scheme that relies on this approach is presented in [31]. This algorithm uses orthogonal and quasi-orthogonal codewords and yields some good theoretical results although the tests are not performed with real data.

An efficient watermarking scheme that also uses spherical codes is presented in [19]. This algorithm modifies a trellis code in order to produce a dirty paper code (an idea borrowed from Costa's work on channel capacity [6]) in which multiple codewords are obtained for each message. The watermark is embedded via an iterative method that seeks to ensure the chosen codewords are not confused with others, even after the addition of noise. However, the embedding process offered by this method is extremely expensive in terms of computations.

1.3.4 Specific challenges for video watermarking

Although many watermarking algorithms that are originally designed for still images can be employed to watermark raw video sequences, there are some key challenges in video watermarking that need to be addressed and that are very specific to this type of content. Several non-hostile distortions as well as deliberate attacks are of particular interest to the video watermarking community.

1.3.4.1 Video compression

In order to reduce the storage needs, content owners often re-encode the video files with a different compression ratio or to a different compression format (i.e., transcoding [33]). Extensive research has been devoted so that watermarking algorithms intended for uncompressed video offer robustness to lossy compression. Some examples can be found in [19, 21, 24, 29, 32, 32, 34].

1.3.4.2 Spatial synchronization attacks

Unfortunately, there are several non-hostile processes that introduce spatial desynchronization to the video content. This may result in a drastic loss of performance for some watermarking schemes. Examples of spatial synchronization attacks include changing of display formats (e.g. from 16:9 to 4:3) and changing the spatial resolution (e.g. from NTSC to PAL or vice versa) [16].

In the digital cinema context, a recording made by a handheld camera is an especially serious attack even though it can be considered as non-hostile since the purpose of the camera is not to explicitly remove the embedded watermark. Theatrical camcorder piracy is one of the most common ways of illegally copying a movie [35].

This method consists of someone taking a camcorder into a poorly supervised theatre and creating a copy of the movie being played. Access to digital content can be controlled with a watermark. In the case of a playback control application, the watermark embedded in the video sequence is designed to provide information on whether video players are authorized to display the content or not [5]. Compliant players or devices detect the watermark and obey the encoded usage restrictions. Controlling access to media content that was recorded with a camera inside a movie theatre is a challenging problem. To begin with, the recorded video might be a slightly resized, rotated and cropped version of the original content. Furthermore, these copies are then subjected to video compression. Since the original content is not available during the decoding process (i.e., it is a blind procedure), extracting the watermark is not a straightforward task. The decoding process must, to a certain extent, be robust to some geometric distortions (rotation and scaling), as well as cropping and lossy compression.

Several watermarking methods that are robust to common geometric distortions have been presented. For example, in [22], two watermarks are employed. The first one is used to embed the message while the second one, a 1-bit watermark, is employed as a geometric reference. This reference watermark is embedded in the spatial domain, which results in low robustness. Information hidden in the space domain can be easily lost to quantization, which makes the watermarking scheme vulnerable to lossy compression and other attacks. Once the reference watermark has been changed, the decoder assumes there is no watermark embedded in the content and, therefore, does not search for the hidden message.

A content-based image watermarking method, where robustness to geometric attacks is achieved using feature points from the image, is offered in [36]. This scheme is shown to be successful to certain attacks, but it is computationally intensive and, therefore, may not be practical for real-time video applications.

Multiresolution analysis can be an important tool for designing watermarks that can withstand geometric distortions. A method for image watermarking in the wavelet domain is presented in [37]. The watermark is applied to the discrete wavelet transform (DWT) coefficients of a sub-image. This sub-image is constructed from the original content using small blocks that are chosen via a chaotic map. Although the scheme is extremely robust to cropping, it does not provide an adequate solution for rotation attacks.

A video watermarking method that also relies on wavelets is presented in [34]. In this case, the watermark is embedded in every video frame by applying DWT to the frames and replacing certain coefficients with the maximum or minimum value of their neighbouring coefficients. This scheme is proven to be robust to mild geometric attacks and high compression. However, the amount of distortion introduced in the frames cannot be controlled.

Complex wavelets have also been employed to create watermarks that are robust to geometric distortions. The complex wavelet transform is an overcomplete transform and therefore creates redundant coefficients, but it also offers some advantages over the regular wavelet transform. Two of the main features of complex wavelets are approximate shift invariance and good directional selectivity [38]. These properties can be employed to produce a watermark that can be decoded even after the original content

has undergone extensive geometric distortions. When dealing with signals that have more than one dimension, the Dual-Tree Complex Wavelet Transform (DT CWT) [38] is a particularly valuable tool, since it adds perfect reconstruction to the list of desirable properties the regular complex wavelets have.

Most watermarking methods rely on embedding a pseudorandom pattern in the transform coefficients of the host image or frame. The same, however, cannot be achieved with the Dual-Tree Complex Wavelet Transform. DT CWT is a redundant transformation and, therefore, some components of the watermark might be lost during the inverse transform process [25]. In order to reduce this loss of information, a watermark formed of a pseudorandom sequence of valid CWT transform coefficients is proposed in [25] and [39]. A four-level DT CWT is applied to the original content, and the watermark is added to the coefficients from levels 2 and 3. Although the ideas portrayed in these efforts show some potential, the robustness of such scheme has never been tested.

Another watermarking method that uses DT CWT is presented in [26]. In this method, the content is also subjected to a four-level DT CWT decomposition, and the watermark is added to the two highest levels using the spread spectrum technique. However, the decoding process is not blind and, therefore, the applications of this scheme are very limited.

1.3.4.3 Temporal synchronization attacks

Temporal synchronization attacks pose another challenge to video watermarking systems. When the watermark to be embedded is not the same for all frames of a video sequence, the hidden data can be desynchronized with a simple operation such as frame

dropping. A watermark decoder is then unable to retrieve the appropriate message. The temporal synchronization problem has been, somehow, an overlooked topic in the literature.

In [40] a watermarking encoder models the construction of a watermark by using a state machine key generator. The decoder uses a queue and a state predictor to perform a search that establishes and maintains temporal synchronization.

Another video watermarking scheme that provides temporal synchronization is presented in [41]. Synchronization is achieved by sending side information along with the watermarked video sequence. Although this method works in real-time, the need for side information at the decoder makes the scheme impractical for most applications.

1.3.4.4 Collusion attacks

For many applications, it is desirable that watermarking schemes are collusion-resistant. Collusion attacks are hostile attempts by more than one attacker to remove watermarks. They can be divided into two types [16]. The first type of collusion attack takes place when the same watermark is embedded into different copies of different data. In the case of video, for instance, the same watermark can be embedded in every frame of a video sequence. An estimate of the watermark can be obtained by examining all the data of the different watermarked copies and producing a refined estimate of the watermark via a linear combination, e.g. computing the average of the individual estimations. Once a good estimate of the watermark is available, unwatermarked data can be easily obtained by subtracting the estimated watermark from the watermarked content. The second kind of collusion attack involves embedding a different watermark into each copy of the same data (this is known as *fingerprinting* [42]). If there is a group of

attackers where every one possesses a copy of the same content but with a different watermark, then they can join forces and produce an unwatermarked version. This collusion attack consists of averaging all the available watermarked copies, in order to produce the unwatermarked data. This is possible because averaging different watermarks usually converges toward zero.

Collusion attacks can sometimes be successful even when only one copy of the video sequence is available. In this case, the first type of collusion can be successful, since different images are obtained from moving scenes but the same watermark is inserted in each frame. For the second type of collusion when a different watermark is embedded in each frame, static scenes become vulnerable, since they produce similar images.

As described in [43], watermarking algorithms can offer some robustness to collusion if there is a dependency between the watermark and the host content (for instance, using informed coding [5]). This means that, unlike the above case, if two frames are very similar, their embedded watermarks should be highly correlated. Also, when two frames are really different, the watermarks inserted into those frames should be very dissimilar.

In [27], a spatially localized watermark is embedded according to the content of each video frame. In this approach, a simple watermark is embedded in some key locations of each frame. The decoder can later detect these locations and look for the presence or the absence of a watermark.

The topic of anti-collusion forensics for fingerprinting applications has been explored in [44]. In this approach, the spread spectrum additive embedding technique is

used. However, a non-blind decoding scenario is assumed, that is, the original content must be available at the decoder end. The collusion attack that averages different copies of the same content is considered and a likelihood-based classifier is employed to estimate the number of colluders.

1.4 Objectives

Based on the previous sections, it can be stated that digital video watermarking is an important yet challenging subject. There are many possible applications, and each one of them has different requirements and poses specific problems. Because of this, watermarking systems need to be designed with a particular application in mind. We need to consider:

1. The types of non-hostile distortion that are likely to take place as well as the malicious attacks that might be attempted to purposely destroy the watermark.
2. Time constraints, i.e., whether or not we are dealing with an application that must be performed in real time This question has a direct impact on the level of computational complexity that can be allowed in our scheme.
3. Another important factor is watermark capacity, i.e., how long must the embedded message be.
4. For decoding purposes, it is essential to know if the original content is going to be available or if the watermark must be extracted blindly.

With these issues in mind, the objective of this thesis is to propose novel watermarking algorithms for different types of applications while requiring a small

amount of computations. By achieving this goal, these algorithms can be employed in real-time video applications.

We consider applications such as proof of ownership and digital fingerprinting. In the case of employing a watermark to prove ownership, the owner of the content might wish to embed the logo of his/her company into the image or video file he/she has created. Digital fingerprinting schemes are technical means to discourage people from illegally redistributing the digital data they have legally obtained. This is done by inserting unique watermarks (fingerprints) into each copy of the content prior to its distribution. For these applications, long messages are required. In addition, the watermarks representing these messages must be robust to common distortions such as additive noise and lossy compression. We address these requirements and propose a new high-capacity watermarking scheme. This method has the same capacity than a leading watermarking scheme [19] while offering better robustness to common distortions. Moreover, the computational time of the proposed scheme is kept low so that the algorithm can be used for real-time video applications.

Next, we consider the case of watermarking for playback control. This is an important application, since most of the illegal copies of current movies available in the black market are a result of camcorder piracy. This is the practice of bringing a camera into a poorly supervised cinema and recording the movie that is been screened. Our goal is to design a watermark that is robust to cropping, rotation, scaling, and lossy compression. If such a watermark is embedded into the content that is shown in a theatre, the embedded message is likely to be present in the pirate video as well. A compliant DVD player can detect the watermark and prevent displaying the content. For this type of

application, a long watermark message is not required. The existence or absence of the watermark is all that is needed by the player to make the decision as to whether or not play the video. We introduce two watermarking schemes that address the requirements of this application. They are robust to geometric distortions and lossy compression. These schemes are also simple to implement and the watermarks are decoded blindly, i.e., without relying on the original content.

These are the ideas that will be addressed in this document. This is a manuscript-based thesis and, for this reason, it follows the specifications required by our university for this format. In addition to this introductory chapter, the thesis includes three chapters which were originally prepared for journal publication and have been slightly modified in order to offer a logical progression in the thesis. The final chapter discusses the conclusions and directions for future work. The remaining chapters are now summarized.

Chapter 2 introduces an image watermarking algorithm that can be used for proof of ownership, digital fingerprinting or any other application that requires the embedding of a long message into the content. The proposed scheme offers the same high capacity than a state-of-the-art paper [19] (i.e., one message bit in every 8×8 pixel block). Moreover, our method achieves better robustness to common signal distortions, in terms of the bit error rate. In addition, this scheme requires a significantly lower amount of computations than [19], making it more suitable for video applications where speed is a main concern. For example, using the same computer, our method embeds a 1,024-bit watermark in a 256×256 pixel image in 7 seconds, while the competing scheme requires 1.83×10^4 seconds to embed the watermark. Furthermore, since every message bit is embedded independently from other bits in the complete message string, parallel

processing can be employed; this makes the proposed method appropriate for real-time applications.

This is achieved by employing an iterative algorithm that constructs a set of evenly distributed codewords prior to the embedding process. The use of this set of codewords diminishes the distortions introduced in the image by watermarking. This is because a strong correlation between an 8×8 pixel block (where the watermark is to be embedded) and one of the codewords exists. Therefore, the codeword can be embedded to the content without altering it in a noticeable way. Although this work focuses on still images, the reduced computational time of our scheme makes it suitable for video applications.

In chapter 3, we introduce a new watermarking scheme designed for video playback control applications. The watermarking scheme is proven to be robust to scaling, cropping, rotation, compression and a combination of all of these attacks. The proposed method is designed to be simple to implement so as not to add to the cost and complexity of DVD players.

Robustness to geometric distortions is achieved by using the Dual-Tree Complex Wavelet Transform (DT CWT) during the watermark embedding process. DT CWT provides important features such as perfect reconstruction, approximate shift invariance and good directional selectivity, i.e., it maintains the advantages of regular wavelets but avoids the shortcomings [38]. These characteristics are employed to create a method that relies on the orientation of edges rather than pixel positions to embed the watermark, which is robust to geometric distortions.

The watermark is a vector of length six whose elements are 1's and -1's. To make the watermark difficult to detect by an attacker, the elements of the watermark are pseudo-randomly reordered before inserting them into a frame. For each frame, a four-level DT CWT is applied and the watermark is added to the magnitudes of some of the level-4 coefficients.

Unfortunately, DT CWT is a redundant transformation and, therefore, some components of the watermark might be lost during the inverse transform process. The lost information corresponds to the part of the watermark that lies in the null space of the inverse DT CWT [25]. Because of this inconvenience, several hundred frames of the video sequence are required in order to decode the watermark successfully.

This problem is addressed in chapter 4. To reduce the information loss, we embed the DT CWT coefficients of the watermark in the host content, *instead of* embedding the actual watermark. Thus, the one-level DT CWT transform is applied to the watermark. This results in a low-pass component and six subbands that contain the details. The coefficients of the six subbands form the data to be embedded in the host video frame. Although the complexity of the scheme is increased when compared to the previous method, the number of frames required to decode the watermark is only half the amount needed for the scheme from chapter 3.

In this new method, the watermark is a random set of 1's and -1's. A one-level DT CWT is applied to this watermark and the coefficients of this transformation become the data that are embedded into the video sequence. Every frame of the original video sequence is transformed with a four-level DT CWT. The content is examined to determine how strongly the watermark embedding should be. Thus, the watermark

coefficients are properly weighted and added to the coefficients of levels 3 and 4. The proposed watermarking method is also robust to lossy compression and some common geometric attacks such as rotation, scaling and cropping.

Finally, chapter 5 offers conclusions about the proposed schemes, relating the manuscripts to each other and, more generally, to the field of digital watermarking. Suggestions for future research are also provided.

1.5 References

[1] Lunney Jr. The death of copyright: Digital technology, private copying, and the digital millennium copyright act. *Virginia law review 87(5),* pp. 813, 2001.

[2] A. Grosso, "Why the digital millennium copyright act is a failure of reason," *Commun ACM,* vol. 45, pp. 19-23, 2002.

[3] Q. Liu, R. Safavi-Naini and N. P. Sheppard, "Digital rights management for content distribution," in *ACSW Frontiers '03: Proceedings of the Australasian Information Security Workshop Conference on ACSW Frontiers 2003*, pp. 49-58.

[4] S. Landau, R. Stratulate and D. Twilleager, "Consumers, fans, and control: What the games industry can teach hollywood about DRM," in *DRM '06: Proceedings of the ACM Workshop on Digital Rights Management,* 2006, pp. 1-8.

[5] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking.* San Francisco, Calif.: Morgan Kaufmann, 2002.

[6] M. Costa, "Writing on dirty paper (Corresp.)," *Information Theory, IEEE Transactions on,* vol. 29, pp. 439-441, 1983.

[7] J. A. Bloom, I. J. Cox, T. Kalker, J. -. M. G. Linnartz, M. L. Miller and C. B. S. Traw, "Copy protection for DVD video," *Proceedings of the IEEE,* vol. 87, pp. 1267-1276, 1999.

[8] T. Kalker, G. Depovere, J. Haitsma and M. J. Maes, "Video watermarking system for broadcast monitoring," in 1999, pp. 103-112.

[9] J. Lubin, J. A. Bloom and H. Cheng, "Robust, content-dependent, high-fidelity watermark for tracking in digital cinema," in *Security and Watermarking of Multimedia Contents V, Proceedings of SPIE,* 2003.

[10] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing,* vol. 66, pp. 283-301, 1998/5/28.

[11] S. Arena, M. Caramma and R. Lancini, "Digital watermarking applied to MPEG-2 coded video sequences exploiting space and frequency masking," in *Image Processing, 2000. Proceedings. 2000 International Conference on,* 2000, pp. 438-441 vol.1.

[12] Chun-Shien Lu, Jan-Ru Chen, Hong-Yuan Mark and Kuo-Chih Fan, "Real-time MPEG2 video watermarking in the VLC domain," in 2002, pp. 20552.

[13] J. Zhang, H. Maitre, J. Li and L. Zhang, "Embedding watermark in MPEG video sequence," in *Multimedia Signal Processing, 2001 IEEE Fourth Workshop on,* 2001, pp. 535-540.

[14] Z. Zhao, N. Yu and X. Li, "A novel video watermarking scheme in compression domain based on fast motion estimation," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on,* 2003, pp. 1878-1882 vol.2.

[15] Y. Dai, L. Zhang and Y. Yang, "A new method of MPEG video watermarking technology," in *Communication Technology Proceedings, 2003. ICCT 2003. International Conference on,* 2003, pp. 1845-1847 vol.2.

[16] G. Doërr and J. Dugelay, "A guide tour of video watermarking," *Signal Processing: Image Communication,* vol. 18, pp. 263-282, 2003/4.

[17] R. Lancini, F. Mapelli and S. Tubaro, "A robust video watermarking technique in the spatial domain," in 2002, pp. 251-256.

[18] Qing-Ming Ge, Zhe-Ming Lu and Xia-Mu Niu, "Oblivious video watermarking scheme with adaptive embedding mechanism," in *Machine Learning and Cybernetics, 2003 International Conference on,* 2003, pp. 2876-2881 Vol.5.

[19] M. L. Miller, G. J. Doerr and I. J. Cox, "Applying informed coding and embedding to design a robust high-capacity watermark," *Image Processing, IEEE Transactions on,* vol. 13, pp. 792-807, 2004.

[20] G. C. Langelaar, I. Setyawan and R. L. Lagendijk, "Watermarking digital image and video data. A state-of-the-art overview," *Signal Processing Magazine, IEEE,* vol. 17, pp. 20-46, 2000.

[21] Hongmei Liu, Nuo Chen, Jiwu Huang, Xialing Huang and Y. Q. Shi, "A robust DWT-based video watermarking algorithm," in 2002, pp. 631-634.

[22] C. V. Serdean, M. A. Ambroze, M. Tomlinson and J. G. Wade, "DWT-based high-capacity blind video watermarking, invariant to geometrical attacks," *IEE Proceedings -- Vision, Image & Signal Processing,* vol. 150, pp. 51-58, Feb 2003. 2003.

[23] A. Herrigel, S. Voloshynovskiy and Y. Rytsar, "The watermark template attack," 2001.

[24] V. Cappellini, F. Bartolini, R. Caldelli, A. De Rosa, A. Piva and A. Bami, "Robust frame-based watermarking for digital video," in *Database and Expert Systems Applications, 2001. Proceedings. 12th International Workshop on,* 2001, pp. 825-829.

[25] P. Loo and N. Kingsburry, "Digital watermarking using complex wavelets," in *International Conference on Image Processing, ICIP,* 2000, pp. 29-32.

[26] N. Terzija and W. Geisselhardt, "Digital image watermarking using complex wavelet transform," in *MM\&Sec '04: Proceedings of the 2004 Workshop on Multimedia and Security,* 2004, pp. 193-198.

[27] K. Su, D. Kundur and D. Hatzinakos, "A content dependent spatially localized video watermark for resistance to collusion and interpolation attacks," in *International Conference on Image Processing*, 2001, pp. 818-821 vol.1.

[28] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *Information Theory, IEEE Transactions on,* vol. 47, pp. 1423-1443, 2001.

[29] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction," *Image Processing, IEEE Transactions on,* vol. 13, pp. 1627-1639, 2004.

[30] F. Ourique, V. Licks, R. Jordan and F. Perez-Gonzalez, "Angle QIM: A novel watermark embedding scheme robust against amplitude scaling distortions," in *Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05). IEEE International Conference on,* 2005, pp. ii/797-ii/800 Vol. 2.

[31] A. Abrardo and M. Barni, "Informed watermarking by means of orthogonal and quasi-orthogonal dirty paper coding," *Signal Processing, IEEE Transactions on [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on],* vol. 53, pp. 824-833, 2005.

[32] M. L. Miller, G. J. Dorr and I. J. Cox, "Dirty-paper trellis codes for watermarking," in *International Conference on Image Processing*, 2002, pp. II-129-132, vol.2.

[33] Q. Tang, R. K. Ward and P. Nasiopoulos, "An efficient MPEG2 to H.264 half-pixel motion compensation transcoding," in *Image Processing, 2006 IEEE International Conference on,* 2006, pp. 865-868.

[34] P. W. Chan, M. R. Lyu and R. T. Chin, "A novel scheme for hybrid digital video watermarking: approach, evaluation and experimentation," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 15, pp. 1638-1649, Dec. 2005. 2005.

[35] Anonymous, Motion Picture Association of America, *2007.* Available: http://www.mpaa.org/piracy.asp

[36] P. Bas, J. M. Chassery and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Transactions on Image Processing,* vol. 11, pp. 1014, 2002.

[37] Z. Dawei, C. Guanrong and L. Wenbo, "A chaos-based robust wavelet-domain watermarking algorithm," *Chaos, Solitons & Fractals,* vol. 22, pp. 47-54, 2004/10.

[38] N. Kingsbury, "Image processing with complex wavelets," *Philosophical Transactions. Mathematical, Physical, and Engineering Sciences,* vol. 357, pp. 2543, 1999.

[39] P. Loo and N. Kingsbury, "Digital watermarking with complex wavelets," in *IEE Seminar on Secure Images and Image Authentication*, 2000, pp. 10/1-10/7.

[40] E. T. Lin and E. J. Delp, "Temporal synchronization in video watermarking," *Signal Processing, IEEE Transactions on* [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on], vol. 52, pp. 3007-3022, 2004.

[41] O. Harmanci, M. Kucukgoz and M. K. Mihcak, "Temporal synchronization of watermarked video using image hashing," *Proceedings of SPIE* Volume 5681, Security, Steganography, and Watermarking of Multimedia Contents VII, Edward J. Delp III, Ping W. Wong, Editors, March 2005, pp. 370-380.

[42] M. Wu, W. Trappe, Z. J. Wang and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," *Signal Processing Magazine, IEEE,* vol. 21, pp. 15-27, 2004.

[43] K. Su, D. Kundur and D. Hatzinakos, "Novel approach to collusion-resistant video watermarking," in *Proc. SPIE* Vol. 4675, Security and Watermarking of Multimedia Contents IV, Edward J. Delp; Ping W. Wong; Eds., 2002, pp. 491-502.

[44] Z. J. Wang, M. Wu, H. V. Zhao, W. Trappe and K. J. R. Liu, "Anti-collusion forensics of multimedia fingerprinting using orthogonal modulation," *Image Processing, IEEE Transactions on,* vol. 14, pp. 804-821, 2005.

CHAPTER 2: A FAST HIGH-CAPACITY INFORMED CODING SCHEME FOR IMAGE WATERMARKING[1]

2.1 Introduction

For digital media, the risk of piracy is aggravated by the abundance of high-capacity digital recording devices. Cryptography is the most common method applied to protect digital content. Unfortunately, cryptography can only protect the content while in transit, but once decrypted, the content has no further protection [1].

Watermarking has become the technology of choice for discouraging users of digital content from creating illegal copies of that information. Watermarking is the practice of imperceptibly altering the media content by embedding a message which can be used to identify either the owner of that content or the device that created the illegal copy. Content fidelity and watermark robustness form two key challenges. Hiding a message within the digital content should not have any significant impact on the content's fidelity. A watermark should also prevail even after the content (where the message is hidden) is altered by common signal processing operations such as additive noise, compression, intensity scaling, and filtering.

The length of the message to be embedded is also a concern for many watermarking systems. For some applications a large message is essential. For instance, if content owners choose to hide a logo image into the work they have created or the

[1] A version of this chapter has been submitted for publication. Authors: L. E. Coria, P. Nasiopoulos, and R. K. Ward. Our gratitude to Dr. Gwenaël Doërr for kindly providing the Dirty Paper Trellis Watermarking Software.

watermark is going to be the serial number of a device, then a long watermark message is required since a significantly large number of bits are needed to represent an image or the large number of devices manufactured. A watermarking scheme that embeds a long message, otherwise known as a high capacity watermarking method, can be considered robust if most of the binary message is decoded correctly. For instance, if at least 90% of the bits are properly retrieved, then the logo can be easily perceived and ownership of the content can be claimed.

Recent work [2-7] has shown that the capacity of watermark messages can be increased if the information derived from the content is used during the embedding process. In addition, the fidelity and robustness of these content-dependent schemes have been successfully demonstrated. This is achieved by encoding the message bit 0 by a codeword chosen from a predefined set and the bit 1 by a codeword chosen from another predetermined set. For each case, the codeword is chosen such that it causes the least distortion to the content (i.e., the one that is most similar to the content). This strategy is known as informed coding.

Quantization Index Modulation (QIM) watermarking is a method that relies on informed coding [2]. In QIM schemes, the amplitude of a vector of pixels or frequency coefficients is quantized using a quantization lattice. While this approach provides a gain in the watermark capacity over other content-dependent schemes, QIM watermarking offers no robustness to valumetric scaling. A variation of this method that addresses this issue has been proposed in [4]. However, the trade-off in this case is a reduction in the watermark capacity.

Another option for informed coding algorithms is to use spherical codewords along with correlation decoding. If all the codewords are located on the surface of a unit sphere, they have the same energy and, therefore, robustness to valumetric scaling can be achieved. One example of a scheme that relies on this approach is presented in [5]. This algorithm uses orthogonal and quasi-orthogonal codewords and yields good theoretical results although no tests are performed with real data.

An efficient watermarking scheme that also uses spherical codes is presented in [7]. This algorithm modifies a trellis code in order to produce a dirty paper code (an idea borrowed from Costa's work on channel capacity [8]) in which multiple codewords are obtained for each multi-bit message. The watermark is embedded using an iterative method so that the chosen codewords will not be confused with others, even after the addition of noise. However, the computational time of the embedding process used in this method is extremely high.

We developed a watermarking method that offers a significantly (three orders of magnitude) lower computational time than [7] while achieving the same high capacity (i.e., 1 message bit in every 8×8 pixel block) and, in terms of the bit error rate, better robustness to common signal distortions. This is achieved by employing an iterative algorithm that constructs a set of evenly distributed codewords before the embedding process. The use of this set of codewords diminishes the distortion of the watermarked images, since it guarantees a strong correlation between one of these codewords and the content where the watermark will be embedded. This results in a robust watermark. The rest of the chapter is structured as follows. Section 2.2 presents an overview of content-dependent watermarking schemes. Section 2.3 describes the proposed image

watermarking method. Performance evaluations are discussed in Section 2.4. Finally, Section 2.5 presents the conclusions.

2.2 Overview of Content-Dependent Watermarking

During the past decade, numerous image watermarking methods have been proposed. Some of these schemes embed the watermark in the space domain [9, 10]. These methods proved to have low robustness when exposed to simple distortions such as additive Gaussian noise. Watermarking schemes that embed the watermark in the frequency domain have been shown to give better performance [1, 11-13]. The transforms that are commonly used are the Discrete Cosine Transform (DCT) [7, 11] and the Discrete Wavelet Transform (DWT) [12, 13]. The image is transformed to the frequency domain and the watermark is embedded in some of the coefficients.

The basic frequency-domain watermarking system is presented in Fig. 2.1. The binary input watermark message *m* (which can be an image, text or other data) is encoded via a watermarking process. The resulting watermark is then inserted in selected frequency coefficients of the original content. Two codewords are generated by a pseudorandom number generator using a seed (starting point), which is a number that is only known to the content owner. This secret number is referred to as a *watermark key*. Each codeword is a sequence of pseudorandom numbers that can only be later recovered if the watermark key is provided. One of the codewords represents bit 0 and the other represents bit 1. Each watermark message bit (0 or 1) is replaced by a codeword, forming an array called **w**. The original content $\mathbf{c_o}$ goes through a frequency transformation and **w** is added to selected frequency coefficients. The inverse frequency transformation converts the coefficients to pixel values, resulting in a watermarked image (content) $\mathbf{c_w}$.

This watermark embedding process is referred to as *blind embedding* because the watermark is added to the image without using any information about the content. In order to retrieve the watermark, the decoder computes the linear correlation between the watermarked content $\mathbf{c_w}$ and the codewords representing bits 0 and 1. Based on the highest correlation values, the array **w**' can be constructed and thus, the decoded watermark message *m*' can be obtained. This decoding process is also blind since the original (unwatermarked) content $\mathbf{c_o}$ is not taken into account when retrieving the watermark message.

A blind decoder is desirable for most watermarking applications since, in some cases, distributing the original content to the detector (or detectors) would defeat the purpose of the watermarking system. In other instances, it may not be possible to have access to the original content at the receiver end. However, using a blind decoder does not necessarily imply the use of a blind embedding process at the encoder side. It has been shown that blind embedding is a poor scheme for information hiding, particularly in terms of robustness and fidelity, since there is no control over the strength with which the watermark is embedded [9]. The use of information derived from the content could be exploited to build more reliable watermark embedding algorithms. The detection watermark value could, for instance, be monitored and modified at the embedder in order to ensure a more effective watermarking method.

More efficient watermarking algorithms can be constructed using informed coding. Here, the original content is examined *before* the codewords are added. For instance, the information derived from the original content can be used to improve the choice of codewords. Instead of relying on a one to one mapping (i.e., one message bit is

represented by only one codeword), several codewords can be employed to correspond to a single message bit. Based on the image, the codeword that results in the least distortion to the content is then chosen to represent the bit to be embedded.

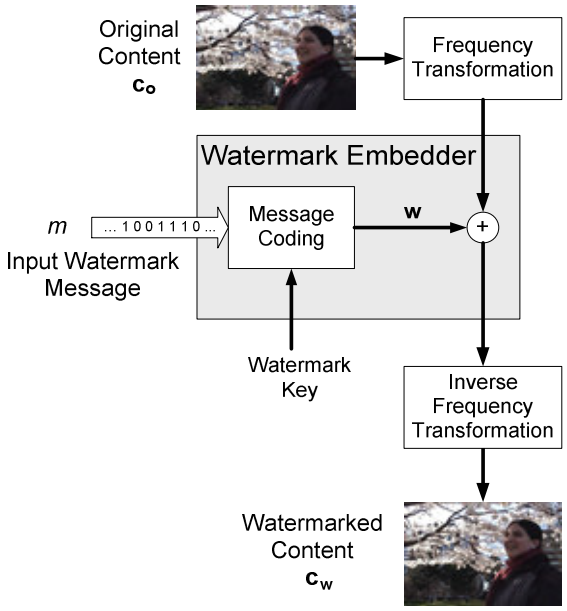


Figure 2.1 A blind watermarking system.

Information from the host image can also be employed to control the strength with which the watermark is added to the content. The watermark can be adjusted so that the embedding algorithm offers a predetermined level of robustness to common distortions. By taking the image into consideration, the embedder can ensure that the correct message is always extracted from the watermarked content (assuming no attacks). This process is known as *informed embedding* and, by combining it with informed coding, it is possible to improve the performance of any watermarking scheme [9].

Fig. 2.2 shows a watermarking system that takes advantage of both informed embedding and informed coding. In this figure, the input watermark message m is represented as a sequence of 0's and 1's that are to be embedded in the original content $\mathbf{c_o}$. To embed the message, the content $\mathbf{c_o}$ is divided into non-overlapping 8 × 8 pixel blocks and the Discrete Cosine Transform (DCT) is applied to each one of these blocks to construct the array $\mathbf{C_o}$. Every message bit is then embedded in a different 8 × 8 block by modifying some of the block's frequency coefficients. However, instead of having one codeword corresponding to all the 0 bits and another codeword corresponding to all the 1 bits as in the traditional case, the watermark key is used to generate one subset of pseudorandom codewords to bit 0 and another subset to correspond to bit 1. These codewords can be recovered later by the decoder, if the key is known. To embed a bit, the selected coefficients from an 8 × 8 block are compared against the whole subset of codewords corresponding to this bit. The codeword that is most similar to these coefficients is chosen. To hide this codeword into these coefficients, the watermark is not added to the coefficients. Instead, the coefficients are adjusted until they reach a predetermined correlation value with the codeword. This is repeated for every message bit and the result of this process is the array of altered coefficients $\mathbf{C_w}$, which after an inverse frequency transformation (IDCT) becomes the watermarked image $\mathbf{c_w}$.

Although informed coding reduces the distortion to the content (when compared to the blind approach), it has the risk of lowering the watermark's robustness to certain distortions such as additive noise and compression. For the blind approach, it is very likely that the codeword representing bit 0 is significantly different from the codeword that represents bit 1. For the informed coding approach however, the probability of two

codewords (representing different bits 0 and 1) being similar, becomes higher as the number of codewords that represent a single bit increases. If the watermarked content is then distorted by some additive noise, the decoder might retrieve the wrong codeword and, consequently, the wrong bit. Therefore, some measures have to be taken to provide an encoding process that has several codeword choices and at the same time guarantees robustness.

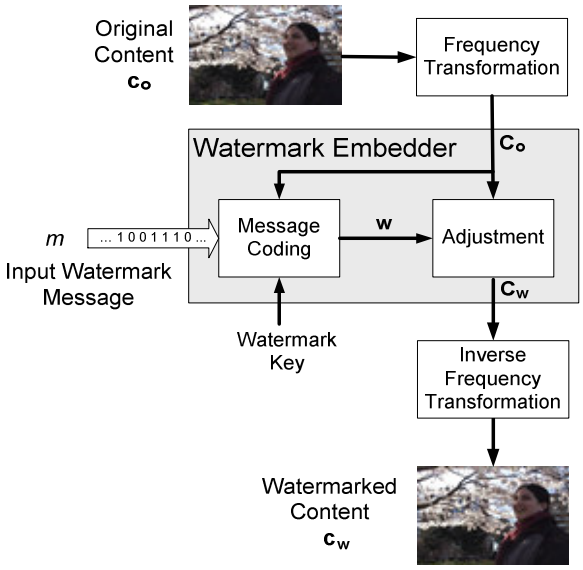


Figure 2.2 A content-dependent watermarking system.

An effective watermarking scheme that combines both informed coding and high robustness is presented in [7]. In this scheme, each bit of the message is embedded in an 8×8 pixel block. A key is used to generate a set of pseudorandom codewords and the entire image is considered when determining the codewords that cause the least distortion. All the possible codewords are arranged in a modified trellis and a

computationally expensive process selects the string of codewords that creates the least overall distortion.

The modified trellis structure (and the heavy computations associated with it) can be avoided if the codewords are carefully constructed instead of being generated in an arbitrary fashion. In the following section we propose an alternative method where the codewords are spread evenly in space. This results in a watermark that is robust to common image processing operations. In addition, the computational time of the embedding process is significantly reduced by few orders of magnitude.

2.3 Proposed Watermarking Scheme

The proposed scheme embeds a watermark in the frequency domain by slightly modifying a small number of coefficients that correspond to the lower frequencies. It is important to note that, when embedding the watermark in the frequency domain, some algorithms choose to modify the highest AC coefficients, as the resultant changes in the image are not easily noticed by the human visual system (HVS). However, since the high frequency information is not crucial for perceiving images of acceptable fidelity, coarse quantization of these coefficients may unfortunately remove the watermark while still producing images of adequate quality. On the other hand, hiding the information by modifying the lower AC coefficients offers a more robust watermark. Altering these coefficients however increases the chance of degrading the image quality. We use the knowledge about the original content and adjust this content so that the distortion caused by the embedded watermark is within a predetermined acceptable range.

Our method incorporates informed coding and embedding techniques as in [7]. But instead of generating each codeword to be embedded using knowledge from the whole image we select the most appropriate codeword for each 8 × 8 block by only considering the information from that particular block. We first design the codewords so that they are evenly distributed in space. Then, considering features from each 8 × 8 pixel block, the embedder selects each codeword so that the least distortion is introduced in the content. The resulting embedded message is binary and each of its bits is hidden in one of the 8 × 8 image blocks. For this reason, the length of the whole watermark message must not exceed the number of blocks in the image. The Discrete Cosine Transform (DCT) is applied to every 8 × 8 block and its L lowest AC coefficients are used for coding the embedded information (see Fig. 2.3). These L coefficients form a one-dimensional array of length L. We call this array of coefficients the coefficient vector, $\mathbf{v_o}$.

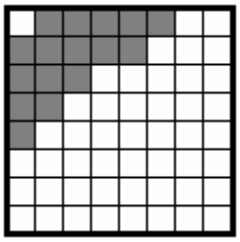


Figure 2.3 The DCT is applied to an 8 × 8 pixel block. Shaded coefficients indicate the AC terms used for constructing the coefficient vector of length L (in this case, L = 16).

The objective now is to embed one of the message bits in vector $\mathbf{v_o}$. For example, to embed bit 0 in $\mathbf{v_o}$, we find in the subset representing bit 0, the codeword that best correlates with vector $\mathbf{v_o}$. As will be shown in subsection 2.3.2, the resulting correlation value must be larger than the highest correlation value between $\mathbf{v_o}$ and the codewords

representing bit 1. If that is not the case, $\mathbf{v_o}$ is then modified. The resulting modified vector is the watermarked vector $\mathbf{v_w}$.

2.3.1 Generating the codewords

We use P pseudorandom vectors, each of length L, as the codewords to represent the 0 and the 1 bits to be embedded. Half of these codewords (P/2) will represent bit 0 and the other half will represent bit 1. These subsets are called A and B, respectively, and are described in (2.1) and (2.2) below.

$$A = \{\mathbf{a}_1, \mathbf{a}_2, \cdots, \mathbf{a}_{P/2}\} \text{ corresponds to bit 0} \tag{2.1}$$

$$B = \{\mathbf{b}_1, \mathbf{b}_2, \cdots, \mathbf{b}_{P/2}\} \text{ corresponds to bit 1} \tag{2.2}$$

where $\mathbf{a}_i$ and $\mathbf{b}_i$ represent the codewords.

The choice of the codewords in these subsets has a big impact on the robustness of the watermark to common attacks and on the fidelity of the watermarked image. If a vector from subset B, for instance, is too close to the chosen vector from subset A, a very small amount of change in the watermarked vector $\mathbf{v_w}$ may cause the decoder to choose the wrong codeword and consequently result in an incorrect binary message. Thus each vector of subset A must be far enough from those in subset B and vice versa. On the other hand, there might be significant image distortion when all the codewords from subset A are too close to each other but far away from all the codewords of subset B. Assume for instance that this is the case and that we wish to embed a codeword from subset A in an 8 × 8 block. Then assume that during encoding, the coefficient vector $\mathbf{v_o}$ obtained from the 8 × 8 block is found to correlate better with a codeword in subset B. In this case, we have to significantly modify $\mathbf{v_o}$. Although the resulting image might offer a robust watermark,

the resulting 8 × 8 block will be very different from the original one. Consequently, it is crucial to accomplish a fine distribution of the codewords in the L-dimensional space, keeping in mind that the random nature of the codeword generating process is essential for any secure watermarking system.

Based on the requirements described above, we introduce an algorithm that creates a set of codewords that are evenly distributed in space. Codewords can be regarded as points on the surface of a unitary sphere of dimension L, where L is the length of any of these codewords. This type of spatial arrangement is known in the literature as a *spherical code* [14]. A significant amount of research has been devoted to the design of spherical codes. However, in most cases, the structure of these codes is very complex [5]. For our application, a simpler yet effective embedding strategy is needed. This condition is reflected on the scheme proposed below.

Our algorithm distributes the P codewords evenly on an L-dimensional sphere (P is the number of codewords used to represent bits 0 and 1), and creates "diversity" among the codewords by avoiding the generation of clusters of codewords that belong to the same subset. Fig. 2.4(a) offers an example of a desirable outcome of this method when $L = 2$ and $P = 14$. In this case, codewords from the two subsets alternate. Fig. 2.4(b) illustrates an undesirable case for the same scenario, where clusters with codewords that belong to the same subset are present, particularly in the bottom half of the plot where three codewords from the same subset are found in a row. The proposed codeword generation algorithm is an iterative one and consists of four steps which are shown in Fig. 2.5 and are described in detail in the following subsections.

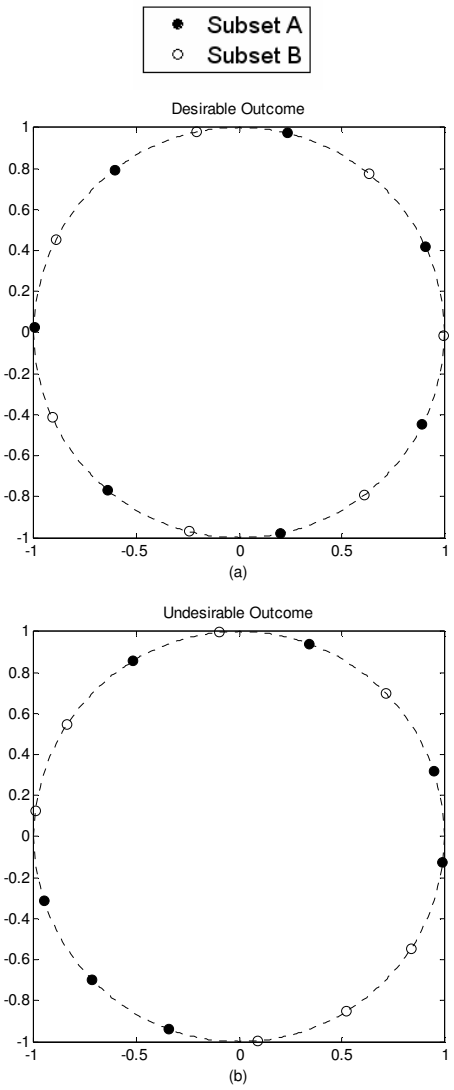


Figure 2.4 (a) A desirable outcome of our codeword generation algorithm. (b) An undesirable outcome of our codeword generation algorithm: there are clusters with codewords that belong to the same subset.

2.3.1.1 Step 1: Initial codewords for subset A

Initial values for the codewords $\mathbf{a}_i$'s could be chosen completely at random. However, the number of computations can be considerably reduced if the codewords provided at the initial stage are spread out through the L-dimensional sphere. These initial values are obtained using the method presented in [15]. The watermark key K is used as

the seed to generate the codeword $\mathbf{a}_1$. This codeword is normalized so that its length is equal to one. Then, we establish the distance between two consecutive codewords, d_0. The second codeword of subset A is generated by adding a vector of magnitude d_0 to the first codeword $\mathbf{a}_1$. The direction of this vector is obtained pseudorandomly by using a multiple of the original key K as its seed. Since our objective is to spread the codewords along the L-dimensional sphere, it is recommended that the parameter d_0 be any value greater than one. The remaining codewords of subset A are created in a similar way, always obeying the following restriction:

$$\left|\mathbf{a}_{i-1} - \mathbf{a}_i\right| = d_0 \quad \text{for } i = 2, 3, ..., P/2. \tag{2.3}$$

Every time a new codeword is obtained, it is normalized so that its length is equal to 1. An example of the codewords obtained after step 1 can be seen in Fig. 2.6 where, for illustration purposes, the simple case $L = 2$ is shown. In this case a unitary circle is used in lieu of a sphere. Although an even distribution has not been achieved yet, the codewords are mildly uncorrelated on the surface of the circle, providing a good initial distribution. This initial distribution allows us to significantly reduce the number of computations in the following step of our algorithm.

2.3.1.2 Step 2: Even distribution of the initial codewords

To achieve an even distribution of the codewords in subset A over the sphere, we modify the iterative algorithm in [16], for the case where $L > 3$. This algorithm is motivated by a century-old physics challenge known as Thomson's problem [17] that involves finding the configuration of N unit point charges on the surface of a unit conducting sphere which minimizes the Coulombic energy. This is an optimization

problem and has been investigated in [17-19] for $L = 3$ and different ranges of N. In order to achieve an optimum solution, the initiatives to solve Thomson's problem are rigorous and elaborate. However, such complexity is not necessary for our application. Instead, we opt for a fast and straightforward algorithm that successfully approximates an even distribution of points on a sphere with $L > 3$.

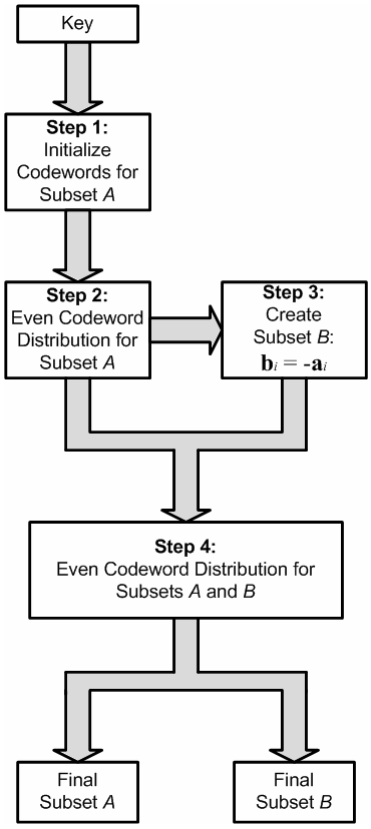


Figure 2.5 Block diagram of our codeword generation algorithm.

For every iteration n, the two points in the unitary L-dimensional sphere that are closest to each other ($\mathbf{p}_i$ and $\mathbf{p}_j$) are found. Each of these two points is moved apart by

some small amount δ in the direction of the difference between them ($\mathbf{p}_i$ - $\mathbf{p}_j$). These points are then normalized to one:

$$\mathbf{p}'_i = \frac{\mathbf{p}_i + \delta(\mathbf{p}_i - \mathbf{p}_j)}{\left|\mathbf{p}_i + \delta(\mathbf{p}_i - \mathbf{p}_j)\right|}. \tag{2.4}$$

$$\mathbf{p}'_j = \frac{\mathbf{p}_j - \delta(\mathbf{p}_i - \mathbf{p}_j)}{\left|\mathbf{p}_j - \delta(\mathbf{p}_i - \mathbf{p}_j)\right|}. \tag{2.5}$$

The smaller δ is, the larger the number of iterations required for the method to approximate an even codeword distribution. On the other hand, if δ has a large magnitude, the method may never accomplish its goal of generating a set of evenly distributed points.

The number of iterations is determined empirically once the values for *P* and *L* have been set (see subsection 2.4.1). When the iterative process is complete, the codewords from subset *A* are approximately evenly distributed.

2.3.1.3 Step3: Initial codewords for subset B

With regards to subset *B*, we take advantage of the fact that the codewords from subset *A* are already evenly distributed. We make

$$\mathbf{b}_i = -\mathbf{a}_i \quad \text{for } i = 1, 2, ..., P/2. \tag{2.6}$$

For the two-dimensional case (*L* = 2), the two subsets are shown in Fig. 2.7(a). In this case, the two subsets form a single group of codewords that are evenly distributed. The codewords alternate when they are tracked along the unitary circle (i.e., a codeword from subset *A* is followed by a codeword from subset *B* and vice versa.). However, this

will not always be the case for a sphere of higher dimensions as is illustrated in Fig. 2.7(b) for the case of $L = 3$. Therefore, a final adjustment is needed.

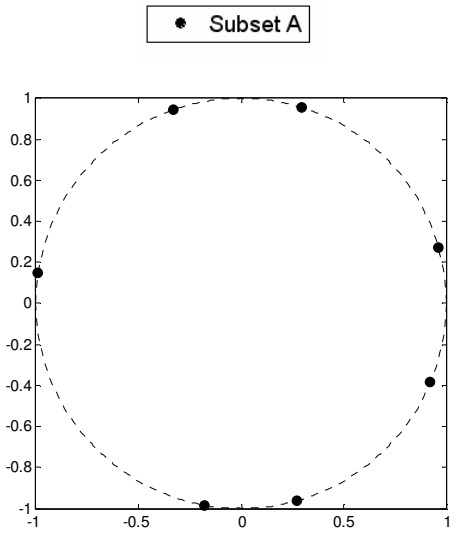


Figure 2.6 Step 1: Initializing codewords from subset *A*.

2.3.1.4 Step 4: Even distribution of all the codewords

At this stage, when each subset is considered independently from the other, all its codewords are distributed evenly on the L-dimensional sphere. However, the codewords of both subsets A and B should be evenly distributed. The codewords of both subsets are considered as codewords in one set and the even distribution process of step 2 above is applied, i.e. at every iteration the two points that are closest to each other are found and then equations (2.4) and (2.5) are applied. This process generates the final codewords for subsets A and B. The distribution of these codewords is shown in Fig. 2.8 for the case when $L = 3$.

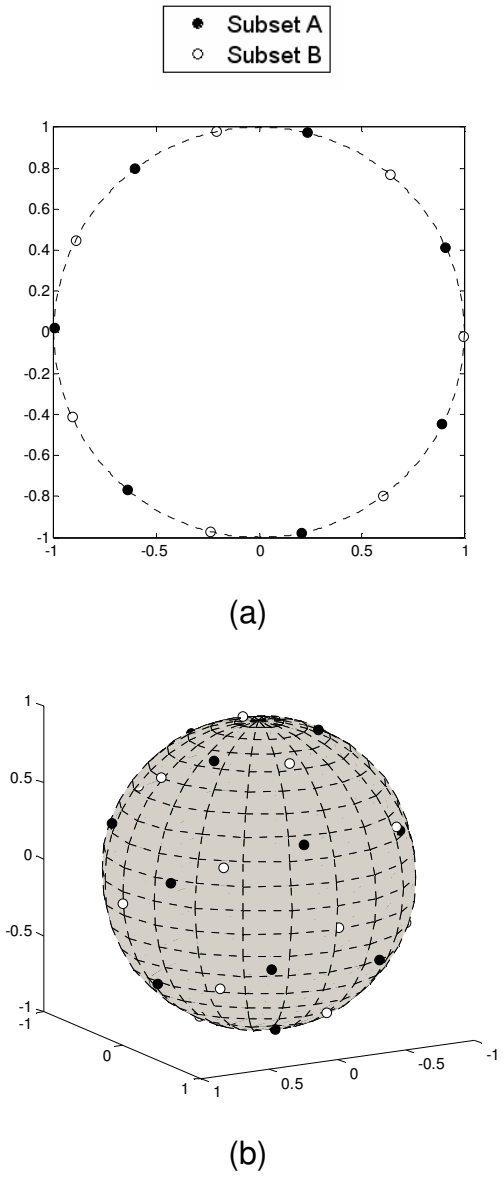


Figure 2.7 Step 3: Creating subset B: case when (a) $L = 2$; (b) $L = 3$.

2.3.2 Embedding the watermark

Once the codewords are constructed, we proceed to embed the message in the image. The embedding process presented in this subsection is similar to the one

introduced in [7]. In [7], however, the entire message is embedded at once, using information from the whole content. In our approach the watermark is embedded on a bit by bit basis, dealing with each 8 × 8 block at a time. By constructing a better set of codewords, we can reduce the computational time of the embedding process and achieve comparable results.

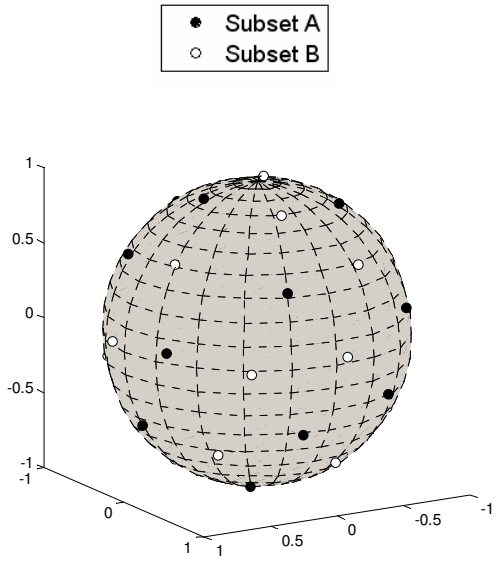


Figure 2.8 Step 4: Even distribution of all the codewords (L = 3).

For every 8 × 8 block, the correlation between the coefficient vector $\mathbf{v_o}$ and each of the $P/2$ codewords corresponding to the desired bit (0 or 1) to be embedded in that particular block is computed. It should be noted that although all the codewords are normalized, $\mathbf{v_o}$ is not. This is because changing its magnitude will not affect the performance of the proposed method.

Assume that we wish to embed bit 0, which is represented by subset *A*. Let the codeword with the highest correlation with $\mathbf{v_o}$ be $\mathbf{a}_{\max}$:

$$\mathbf{a}_{\max} = \max_{\mathbf{a}_i}(\mathbf{a}_i \bullet \mathbf{v_o}) \quad \text{for } i = 1, 2, \ldots P/2. \tag{2.7}$$

Let us also assume that the correlation between $\mathbf{v_o}$ and $\mathbf{a}_{\max}$ is larger than the correlation between $\mathbf{v_o}$ and every $\mathbf{b}_i$ ($\forall i = 1, ..., P/2$). Thus

$$R_{0i} = (\mathbf{a_{max}} - \mathbf{b}_i) \bullet \mathbf{v_o} \quad \forall i = 1, 2, ..., P/2$$

and

$$R_{0i} > 0 \quad \forall i = 1, 2, ..., P/2. \tag{2.8}$$

If an image did not pass through any attacks after it was watermarked, then this condition is enough for the decoder to retrieve the correct watermark. However, if an image distortion attack has happened, then condition (2.8) is not enough to guarantee a robust watermark. Therefore, it must be ensured that R_{0i} has the largest possible value. This is achieved by introducing a robustness threshold R_t so that

$$R_{0i} > R_t \quad \forall i = 1,2,..., P/2. \tag{2.9}$$

This implies that there is a large enough distance between the vector $\mathbf{v_o}$ and (in this case) the unwanted codewords of subset *B*.

If condition (2.9) is not satisfied, then $\mathbf{v_o}$ is modified in an iterative fashion until (2.9) is satisfied. Let at any iteration the modified vector be denoted by $\mathbf{v_w}$. The modification is performed by first finding the minimum R_{0i} over all the codewords from the unwanted subset *B*. That is

$$R_{0\min} = \min_{i=1}^{P/2} R_{0i}. \tag{2.10}$$

To minimize the changes in the image fidelity, $R_{0\min}$ and the vector $\mathbf{b}_{\min}$ associated with it are used to modify the watermarked vector $\mathbf{v_w}$. $\mathbf{v_w}$ is modified so that the next time $R_{0\min}$ is computed, $R_{0\min}$ yields a value exactly equal to R_t, while having a minimum Euclidian distance from the previous value of $\mathbf{v_w}$. This modification is achieved as follows:

$$\mathbf{v_w} \leftarrow \mathbf{v_w} + (R_t - R_{0\min}) \frac{(\mathbf{a_{max}} - \mathbf{b}_{\min})}{|\mathbf{a_{max}} - \mathbf{b}_{\min}|}. \tag{2.11}$$

This procedure is iterated by finding the new value for $R_{0\min}$ (whose associated vector $\mathbf{b}_{\min}$ will be different from the one obtained in the previous iteration) and re-modifying the vector $\mathbf{v_w}$ until $R_{0\min}$ is equal or larger than R_t. The resulting $\mathbf{v_w}$ becomes the watermarked vector, i.e., the modified coefficients of the 8 × 8 block. Finally, the inverse DCT is applied to this block to obtain the spatial values. The entire process is repeated for all blocks of the image.

2.3.3 Decoding the watermark

When decoding the message, the image is again partitioned into 8 × 8 blocks and the DCT is applied. The low frequency coefficients are used to obtain the watermarked vector $\mathbf{v_{wn}}$, which could be a distorted version of the encoded $\mathbf{v_w}$. Next, the nearest neighbour is found by computing the correlation between $\mathbf{v_{wn}}$ and all P codewords. The codeword with the largest correlation value is the one the decoder chooses as the codeword $\mathbf{a_{max}}$ and the bit corresponding to the subset $\mathbf{a_{max}}$ belongs to is chosen as the original message bit.

2.4 Performance Evaluation

2.4.1 Parameter setting

One hundred 256 × 256 pixel images (depicting people, animals, objects and landscapes) are watermarked using four different keys and a message of 1,024 bits length. We call our scheme the *Even Codeword Distribution* (ECD) method. In order to evaluate the performance of our algorithm, the images are also watermarked using two other methods. The first is the one presented in [7]. In [7], the codewords are constructed randomly and the embedding process relies on a *Dirty Paper Trellis* code (DPT). The second method has the same embedding process as our ECD method but with the codewords generated randomly. The latter is denoted as the *Random Codeword Distribution* (RCD) algorithm and is included in these tests with the purpose of illustrating the positive impact that a careful distribution of codewords has on the robustness of a watermarking scheme.

For the case of DPT, the modified trellis has 64 arcs and 64 states. $P = 64$ codewords are also used in both our method (ECD) and RCD. Subset A consists of 32 codewords that represent bit 0 and subset B consists of the other 32 codewords that represent bit 1. As for L, the number of coefficients used to construct vector $\mathbf{v_o}$, it is recommended to keep this parameter lower than 20 as these coefficients are usually altered by the embedded watermark. Furthermore, in order to ensure robustness to compression and other image distortions, the watermark must only be embedded in the lower AC frequency coefficients. For our tests, the length of the codewords was set to $L = 16$. The same average picture fidelity value (PSNR = 35 dB) was kept for every method.

The number of iterations n needed to distribute the codewords in an even fashion was set to 1,000. A larger number of iterations did not significantly improve the results and made the algorithm more time consuming. Various values were tested for the step δ and the best results were found when $\delta = 0.01$. This also happens to be the value recommended in [16] for the three-dimensional case.

2.4.2 Computational time

When compared to DPT, the embedding process of our proposed ECD method is extremely fast. For instance, using a PC that features a Pentium 4 processor with a speed of 2.4 GHz and 512 MB of RAM, our proposed method, ECD, generates codewords in 9 seconds (when 1,000 iterations are computed) and requires just 7 seconds to embed a 1,024-bit watermark in a 256×256 pixel image, for a total of 16 seconds. In the case of DPT on the other hand, 1.83×10^4 seconds are needed for the same process. Although the informed coding process offered by DPT is fast and simple, it is its iterative informed embedding procedure that is very computational intensive.

2.4.3 Robustness to common attacks

Digital images are often subjected to different kinds of signal processing operations, which result in numeric changes to the content. While these operations are sometimes applied to the images with the purpose of destroying any hidden information, often the distortion is not malicious, but necessary. The latter, for example, is the case of image compression. Irrespective of what causes the distortion, it is important to be able to retrieve the watermark of an attacked image. A watermark is known as *robust* if it is capable of withstanding common and legitimate distortions of the host image. In this

section we apply the most common signal processing distortions that images may endure: low-pass filtering, intensity scaling, the addition of Gaussian noise, and lossy JPEG compression.

We measured robustness by computing the Bit Error Rate (BER), which is the percentage of message bits that are decoded incorrectly for every image. A low BER indicates a robust watermark. We computed BER for every image subjected to a certain type of distortion, while varying the magnitude of this distortion. We then calculated the mean of these Bit Error Rates in order to provide an appropriate representation of the robustness to attacks of the three tested schemes.

The Message Error Rate (MER) was also computed. MER is the percentage of messages where one or more of the message bits are erroneously retrieved. MER was computed in the same fashion as the average BER. A watermarking scheme is considered to offer good performance when MER is below 20% [7]. However, for completeness in our experiments, we calculated MER up to its highest value of 100%. Values above 20% are shadowed in the Figures below.

It should be noted that, although the proponents of DPT originally chose MER to assess performance of their watermarking scheme, they have recently preferred the use of BER, which is the most popular performance metric for watermarking algorithms [20]. This choice makes sense, since BER provides a better estimate of the amount of information that has been lost due to image distortion.

2.4.3.1 Low-pass filtering

Due to the fact that watermarks in our scheme are only embedded in low-frequency coefficients, it is expected that our proposed method (and the ones used for comparison) will be robust to low-pass filtering. In order to validate this statement, we filtered the watermarked images using Gaussian filters of width σ_g (the smaller the width of the filter the stronger the filtering process). After repeating these experiments for values of σ_g between 0.1 and 1 BER and MER were computed.

We observe in Fig. 2.9(a) that our method (ECD) yields the lowest average BER. For values of σ_g greater than 0.7, ECD is the only method with a 0% MER (see Fig. 2.9(b)). Nevertheless, DPT achieves the best performance for values of σ_g between 0.5 and 0.7. Table 2.1 shows the average PSNR value of the degraded images when compared to the watermarked content.

Table 2.1 Image fidelity after low-pass filtering: filter width vs. average PSNR.

Filter width	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1
PSNR (dB)	27	27	27	28	30	34	41	62	122	∞

2.4.3.2 Intensity scaling

Let ν denote the scaling factor. The scaled image $\mathbf{c_n}$ is obtained form the original image $\mathbf{c}$ as:

$$\mathbf{c_n}(i, j) = \nu\, \mathbf{c}(i, j). \tag{2.12}$$

In our experiments, the scaling factor ν was varied from 0.2 to 2. We observe from Fig. 2.10(a) that our method achieves a better BER overall (except when the scaling factor equals 1.2). As illustrated in Fig. 2.10(b) ECD offers the lowest MER for $\nu \leq 1$. The three schemes are extremely robust to intensity down scaling. However, they all suffer from high error rates as the pixel intensities increase. This is understandable since, for high values of ν, many pixel values will have to be rounded to the highest possible value that a pixel can have (255) and, therefore, there will be a significant loss of information. The fidelity of the distorted images is shown in Table 2.2.

Table 2.2 Image fidelity after intensity scaling: scaling factor vs. average PSNR.

Scaling factor	0.2	0.4	0.6	0.8	1	1.2	1.4	1.6	1.8	2
PSNR (dB)	7	10	14	20	∞	20	14	11	9	8

2.4.3.3 Gaussian noise

Zero-mean Gaussian noise is added to the watermarked images. BER was computed for noisy sequences with different values for the standard deviation σ. From Fig. 2.11(a) we observe that our method (ECD) performs significantly better than the other algorithms. Despite being a system without memory, ECD also shows the best performance in terms of MER (see Fig. 2.11(b)). Table 2.3 illustrates the relationship between the standard deviation of the Gaussian distribution and the fidelity of the noisy images.

Table 2.3 Image fidelity after Gaussian noise: standard deviation vs. average PSNR.

Standard deviation	1	2	3	4	5	6	7	8	9	10
PSNR (dB)	48	42	39	36	34	33	31	30	29	28

2.4.3.4 JPEG compression

One of the most common signal processing operations that an image has to endure is compression. JPEG, a DCT-based scheme, is one of the most widely known standards for lossy image compression [21]. We tested the robustness of the three schemes for a wide range of compression rates. We varied the JPEG quality factor from 10 to 100 (smaller quality factors correspond to higher compression rates) and obtained the average BER.

Figs. 12(a) and 12(b) illustrate the performance in BER and MER, respectively. We observe that our method (ECD) performs slightly better than DPT in terms of BER. Regarding MER, ECD is the last of the schemes to reach the 20% mark (around a quality factor of 50). Table 2.4 shows the fidelity of the images after JPEG compression.

Table 2.4 Image fidelity after JPEG compression: quality factor vs. average PSNR.

Quality factor	10	20	30	40	50	60	70	80	90	100
PSNR (dB)	31	32	32	33	33	34	35	36	41	62

In summary, the proposed Even Codeword Distribution (ECD) method is significantly less computationally demanding than the DPT scheme, while offering higher robustness to common signal processing attacks. The significance of the evenly distributed codewords is demonstrated by comparing the performance of our method with that of the Random Codeword Distribution (RCD) approach. Table 2.5 summarizes the results.

Table 2.5 Robustness to common attacks. For every test, the method with the best performance in terms of BER and MER is indicated.

	BER	**MER ≤ 20%**
Low-pass Filtering	ECD	DPT for $0.5 < \sigma < 0.7$ ECD for $\sigma \geq 0.7$
Valumetric Scaling	ECD	ECD
Gaussian Noise	ECD	ECD
Compression	ECD (slightly better)	ECD

2.5 Conclusion

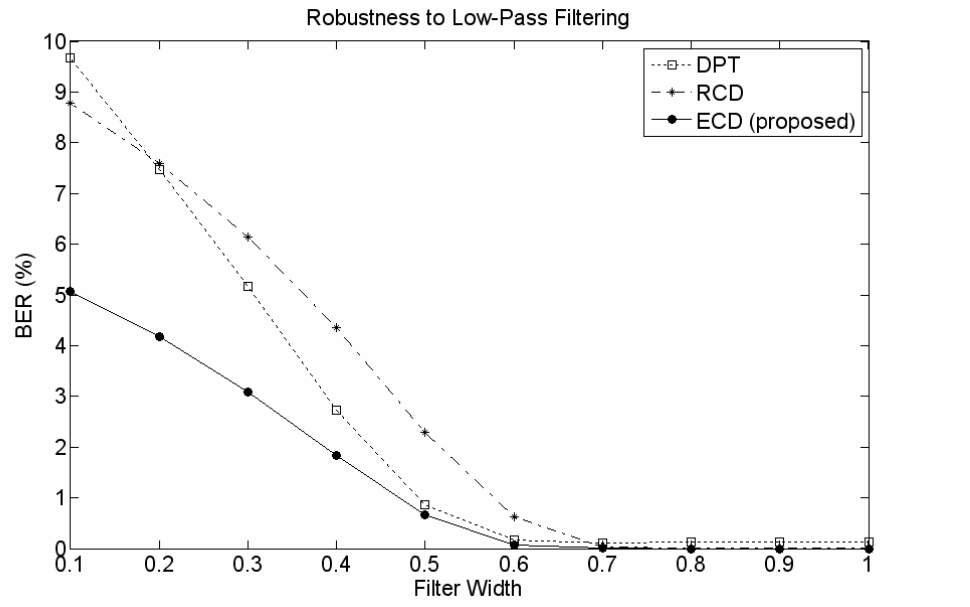
We present an image watermarking method that offers a large data payload and high degree of robustness while maintaining a very low amount of computations. This straightforward and yet effective scheme is feasible because the codewords used for watermarking are carefully constructed (independently of the content and before the embedding process). An iterative algorithm that finds a set of evenly distributed codewords is presented. Once the codewords are chosen, the method embeds every

message bit in one of the image's 8×8 pixel blocks and uses information from only that particular block to ensure robustness and image fidelity.

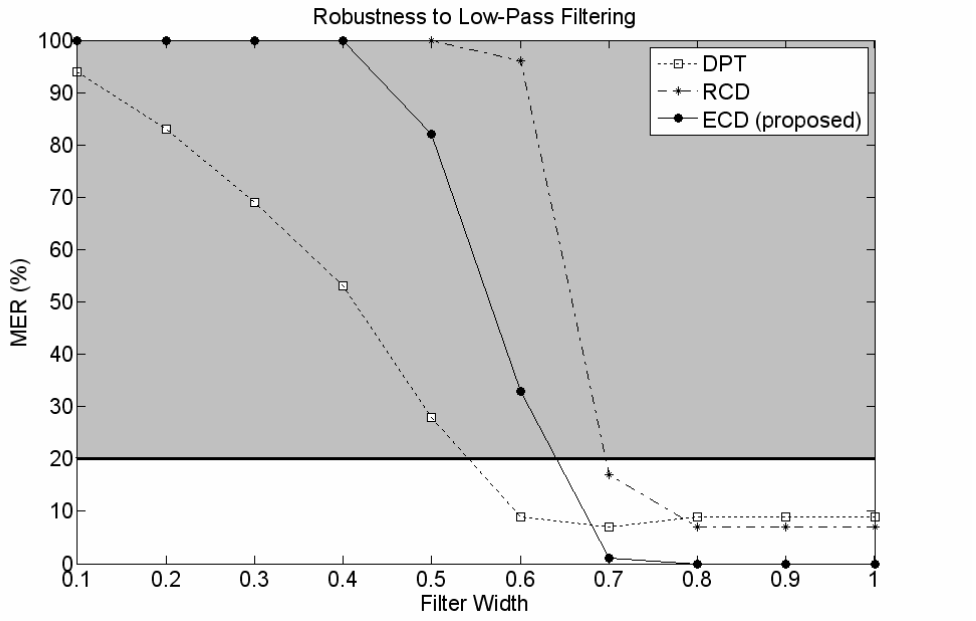
To evaluate the robustness of the watermark to common distortions, experiments were performed on 100 pictures. The proposed method, denoted by ECD, proved to be robust to low-pass filtering, intensity scaling down, additive white Gaussian noise and JPEG compression. Our scheme was compared to a memoryless watermarking method that uses a set of random distributed codewords (RCD). In every test, ECD outperformed RCD, proving the advantage of using evenly distributed codewords over random ones, when designing a robust watermarking algorithm.

Our method was also compared with a leading content-dependent watermarking solution, Dirty Paper Trellis (DPT) [7], which selects the string of random codewords that creates the least overall distortion. It was found that the bit error rate offered by ECD was better than DPT for the four types of distortion.

The method presented in this chapter offers the same capacity and perceptual impact as in [7]. However, in addition to achieving a better performance in terms of the bit error rate, the computational demands of the proposed algorithm are significantly lower, making it more suitable for video applications where speed is a main concern. For example, using the same computer, our method constructs a set of 64 codewords in 9 seconds and embeds a 1,024-bit watermark in a 256×256 pixel image in 7 seconds, while DPT requires 1.83×10^4 seconds to construct the codewords and embed the watermark. Furthermore, since every message bit is embedded independently from other bits in the complete message string, parallel processing can be employed; this makes the proposed method appropriate for real-time applications.

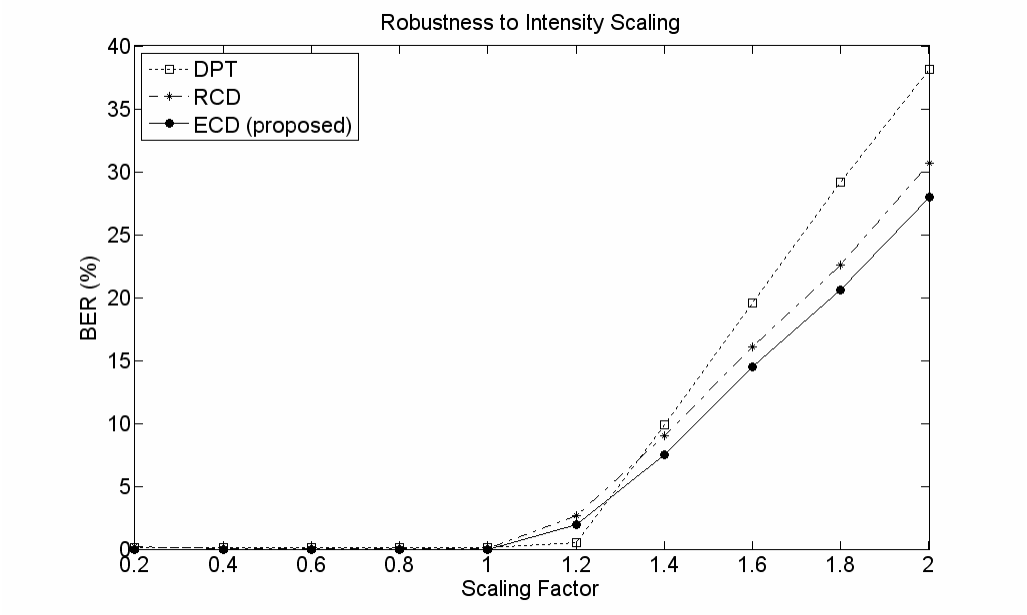


(a)

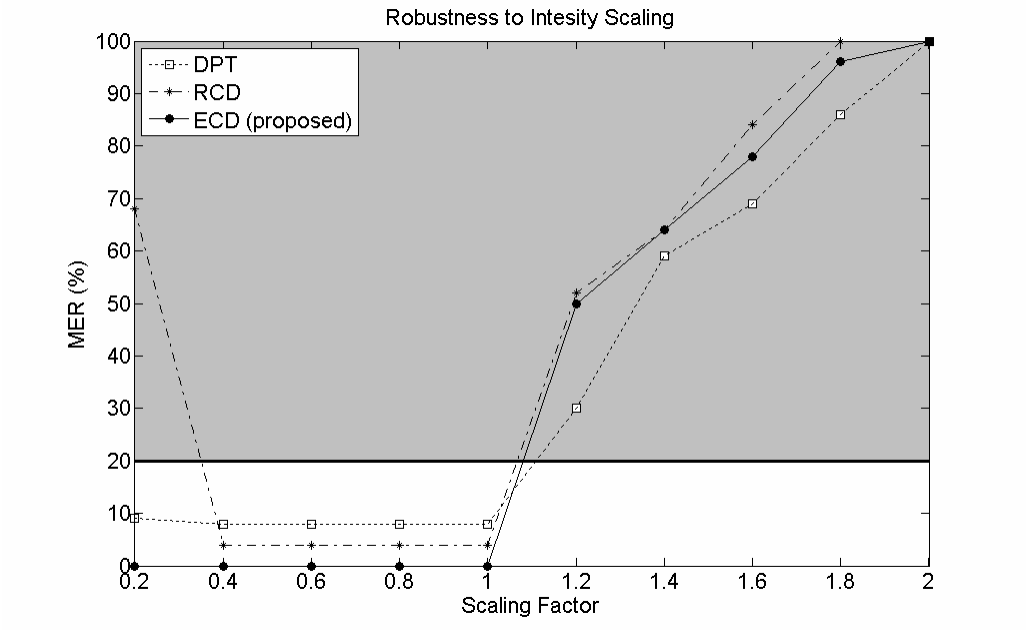


(b)

Figure 2.9 Robustness to low-pass filtering: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired.



(a)



(b)

Figure 2.10 Robustness to intensity scaling: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired.

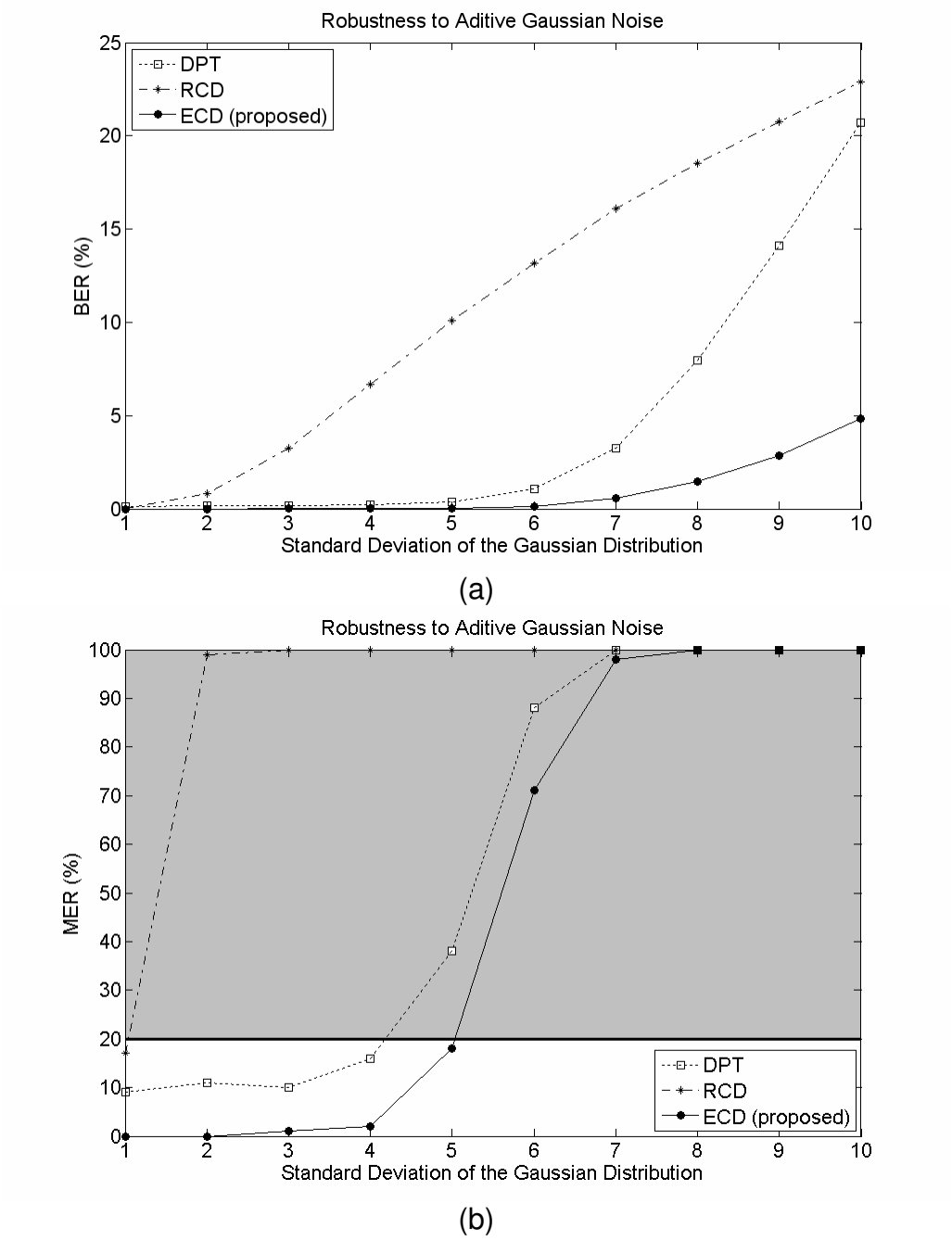
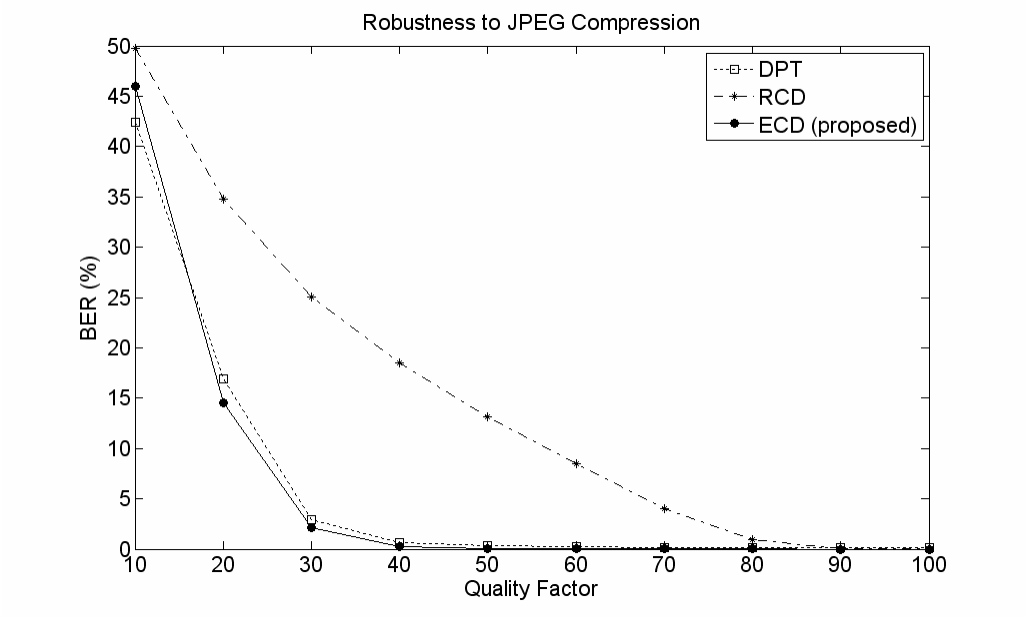
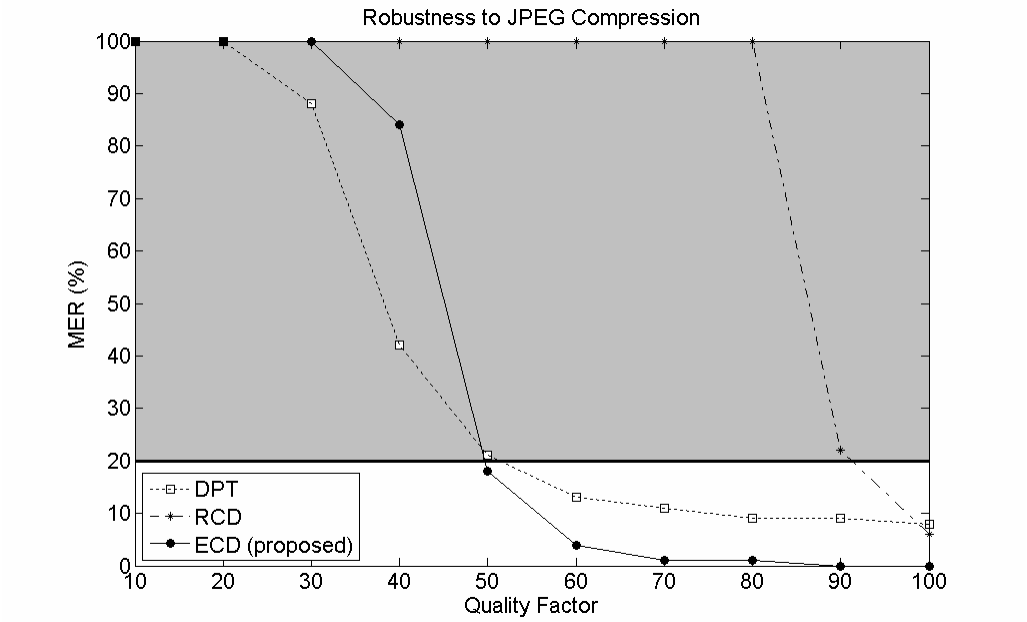


Figure 2.11 Robustness to Gaussian noise: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired.



(a)



(b)

Figure 2.12 Robustness to JPEG compression: (a) BER; (b) MER. Results are shown for Dirty Paper Trellis (DPT), Random Codeword Distribution (RCD) and our method, Even Codeword Distribution (ECD). MER > 20% is not desired.

2.6 References

[1] P. Wayner, *Dissappearing Cryptography*, Second Edition, Ed. New York: Morgan Kaufmann Publishers, 2002, pp. 413.

[2] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *Information Theory, IEEE Transactions on,* vol. 47, pp. 1423-1443, 2001.

[3] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Robust image-adaptive data hiding using erasure and error correction," *Image Processing, IEEE Transactions on,* vol. 13, pp. 1627-1639, 2004.

[4] F. Ourique, V. Licks, R. Jordan and F. Perez-Gonzalez, "Angle QIM: A novel watermark embedding scheme robust against amplitude scaling distortions," in *IEEE International Conference on Acoustics, Speech, and Signal Processing, 2005. Proceedings. (ICASSP '05)* 2005, pp. ii/797-ii/800 Vol. 2.

[5] A. Abrardo and M. Barni, "Informed watermarking by means of orthogonal and quasi-orthogonal dirty paper coding," *Signal Processing, IEEE Transactions on [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on],* vol. 53, pp. 824-833, 2005.

[6] M. L. Miller, G. J. Doerr and I. J. Cox, "Dirty-paper trellis codes for watermarking," in 2002, pp. II-129; II-132 vol.2.

[7] M. L. Miller, G. J. Doerr and I. J. Cox, "Applying informed coding and embedding to design a robust high-capacity watermark," *Image Processing, IEEE Transactions on,* vol. 13, pp. 792-807, 2004.

[8] M. Costa, "Writing on dirty paper (Corresp.)," *Information Theory, IEEE Transactions on,* vol. 29, pp. 439-441, 1983.

[9] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking.* San Francisco, Calif.: Morgan Kaufmann, 2002.

[10] R. Lancini, F. Mapelli and S. Tubaro, "A robust video watermarking technique in the spatial domain," in 2002, pp. 251-256.

[11] J. Lubin, J. A. Bloom and H. Cheng, "Robust, content-dependent, high-fidelity watermark for tracking in digital cinema," in *Security and Watermarking of Multimedia Contents V, Proceedings of SPIE,* 2003.

[12] G. C. Langelaar, I. Setyawan and R. L. Lagendijk, "Watermarking digital image and video data. A state-of-the-art overview," *Signal Processing Magazine, IEEE,* vol. 17, pp. 20-46, 2000.

[13] Hongmei Liu, Nuo Chen, Jiwu Huang, Xialing Huang and Y. Q. Shi, "A robust DWT-based video watermarking algorithm," in 2002, pp. 631-634.

[14] J. Hamkins and K. Zeger, "Asymptotically dense spherical codes. I. Wrapped spherical codes," *Information Theory, IEEE Transactions on,* vol. 43, pp. 1774-1785, 1997.

[15] L. Coria, P. Nasiopoulos and R. Ward, "A robust content-dependent algorithm for video watermarking," in *Proceedings of the ACM Workshop on Digital Rights Management,* 2006, pp. 97-101.

[16] P. Bourke. (1996, Distributing points on a sphere). *2007.* Available: http://local.wasp.uwa.edu.au/~pbourke/geometry/spherepoints/source1.c

[17] E. L. Altschuler, T. J. Williams, E. R. Ratner, R. Tipton, R. Stong, F. Dowla and F. Wooten, "Possible Global Minimum Lattice Configurations for Thomson's Problem of Charges on a Sphere," *Phys. Rev. Lett.,* vol. 78, pp. 2681-2685, Apr. 1997.

[18] J. R. Morris, D. M. Deaven and K. M. Ho, "Genetic-algorithm energy minimization for point charges on a sphere," *Phys. Rev. B,* vol. 53, pp. R1740-R1743, Jan. 1996.

[19] E. L. Altschuler and A. Perez-Garrido, "Global minimum for Thomson's problem of charges on a sphere," *Physical Review E (Statistical, Nonlinear, and Soft Matter Physics),* vol. 71, pp. 047703, 2005.

[20] Chin Kiong Wang, G. Doerr and I. J. Cox, "Toward a better understanding of dirty paper trellis codes," in *2006 IEEE International Conference on Acoustics, Speech and Signal Processing, 2006. ICASSP 2006 Proceedings.,* 2006, pp. II-233–II-236.

[21] K. Sayood, *Introduction to Data Compression*, Second Edition, United States of America: Morgan Kaufmann Publishers, 2000, pp. 636.

[22] G. J. Doërr, "Dirty Paper Trellis Watermarking Software," 28/11/2005. 2005. Available: http://www.adastral.ucl.ac.uk/~gwendoer/dptWatermarking/

CHAPTER 3: A COMPLEX-WAVELET BASED VIDEO WATERMARKING SCHEME FOR PLAYBACK CONTROL[2]

3.1 Introduction

Digital video technology offers important advantages and opportunities for videomakers. Unfortunately, the ease by which anyone can make identical replicas of digital content and distribute it facilitates piracy, which results in significant losses for the movie industry. Digital Rights Management (DRM) systems are being designed in order to protect and enforce the rights associated with the use of digital content [1]. Since the protection offered by encryption is not enough [2], DRM systems also rely on watermarking in a variety of ways. One example is *playback control* [3]. In this case, the watermark embedded in the video sequence provides information on whether video players are authorized to display the content or not. Compliant devices detect the watermark and obey the encoded usage restrictions.

This application solves the problem that content owners constantly face when handheld video cameras are introduced into poorly supervised movie theatres to record feature films that are being projected. The illegally recorded video sequences are compressed and the resulting files are stored, copied and sold as pirated DVDs. These DVDs do not provide any revenues to the creators of the content. Furthermore, potential audiences might be lost since people might choose to watch movies at home instead of going to the theatres where these movies are being currently released. Over 90% of initial

[2] A version of this chapter has been submitted for publication. Authors: L. E. Coria, M. Pickering, P. Nasiopoulos, and R. K. Ward. The authors would like to thank Dr. Nick Kingsbury for providing the software to perform the DT CWT transform operations.

releases that are pirated are a result of camcordering in movie theatres [4]. Therefore, it is desirable that DVD players can detect a watermark that was inserted in a copy of a movie that was only intended for release in cinemas and choose not to play the illegal content.

Watermarks should be hidden in regions of the video where it is less likely to be noticed by a viewer, such as high textured areas. Although robustness to collusion attacks is required for various video watermarking processes [5], it is not a main concern for this particular type of application since illegal copies are made with different cameras and in different theatres and, therefore, averaging the frames from different video sources will not produce an adequate representation of the movie. For this application, however, watermarks must be robust to a combination of geometric transformations (rotation, scaling), cropping and lossy compression. In addition, watermark detection must be blind as DVD players will not have a copy of the original movie to help retrieve the hidden message.

There are a number of video watermarking algorithms that have been designed to resist geometric attacks. In [6], for example, a 1-bit watermark is employed for geometric reference while a second watermark is employed for data payload. The reference watermark is embedded in the spatial domain, which results in low robustness. A content-based image watermarking approach is presented in [7], where robustness to geometric attacks is achieved by using feature points from the image. Although the scheme was shown to be successful to certain attacks, the techniques involved are computationally intensive and offer low robustness at high compression ratios. In [8] the watermark is embedded in each video frame by replacing certain DWT coefficients with the maximum or minimum value of its neighbouring coefficients. This scheme was proven to be robust

to geometric distortion and compression but provided no mechanism for controlling the amount of distortion introduced into the frame.

Robustness to geometric attacks can also be attained by using *complex wavelets*. It has been shown that the Dual-Tree Complex Wavelet Transform (DT CWT) can provide approximate shift invariance and good directional selectivity, with only a modest increase in signal redundancy and computational load [9]. Although the use of the DT CWT for watermarking has been explored in [10], [11] and [12], there are no methods yet available that can offer a watermarking scheme that is both blind and robust.

In this chapter we introduce a new highly robust watermarking scheme designed for playback control applications. This method is based on the Dual-Tree Complex Wavelet Transform and relies on the orientation of edges rather than pixel positions to embed the watermark which is inherently robust to geometric attacks. The rest of the chapter is structured as follows. Section 3.2 includes a brief description of DT CWT and describes our method. Performance evaluations are discussed in Section 3.3. Finally, Section 3.4 presents the conclusions.

3.2 Proposed Method

3.2.1 Brief introduction to the Dual-Tree Complex Wavelet Transform

The Dual-Tree Complex Wavelet Transform (DT CWT) was introduced in [13]. This transform has the desirable properties of both the discrete wavelet transform and the complex wavelet transform: perfect reconstruction, approximate shift invariance, good directional selectivity, limited redundancy and efficient order-*N* computation [9]. This transform is a variation of the original DWT with the main difference being that it uses

two filter trees instead of one, as shown in Fig. 3.1. For a one dimensional signal, the use of the two filter trees results in twice the number of wavelet coefficients as the original DWT. The coefficients produced by these two trees form two sets that can be combined to form one set of complex coefficients of the form

$$y_a + jy_b$$

or in polar form as

$$me^{j\theta}$$

where

$$m = \sqrt{y_a^2 + y_b^2} \text{ and } \theta = \tan^{-1}(y_b / y_a).$$

The dual-tree approach provides wavelet coefficients that are approximately *shift invariant*, i.e., small shifts in the input signal will not cause major variations in the distribution of energy of DT CWT coefficients at different scales. An insight into the shift-invariant nature of DT CWT can be gained by observing the typical impulse responses of the high-pass decimation filters for each tree. Fig. 3.2 shows these two impulse responses. In Fig. 3.1, the filters used in tree B are designed to produce outputs at sample locations that are discarded in tree A.

Approximate shift invariance is a particularly useful property of DT CWT that can be exploited when designing a video watermark that is robust to geometric distortions. If a frame is re-sampled after scaling or rotation, DT CWT should produce approximately the same set of coefficients as the original frame. This property does not hold for other transforms such as the DCT, DFT or DWT.

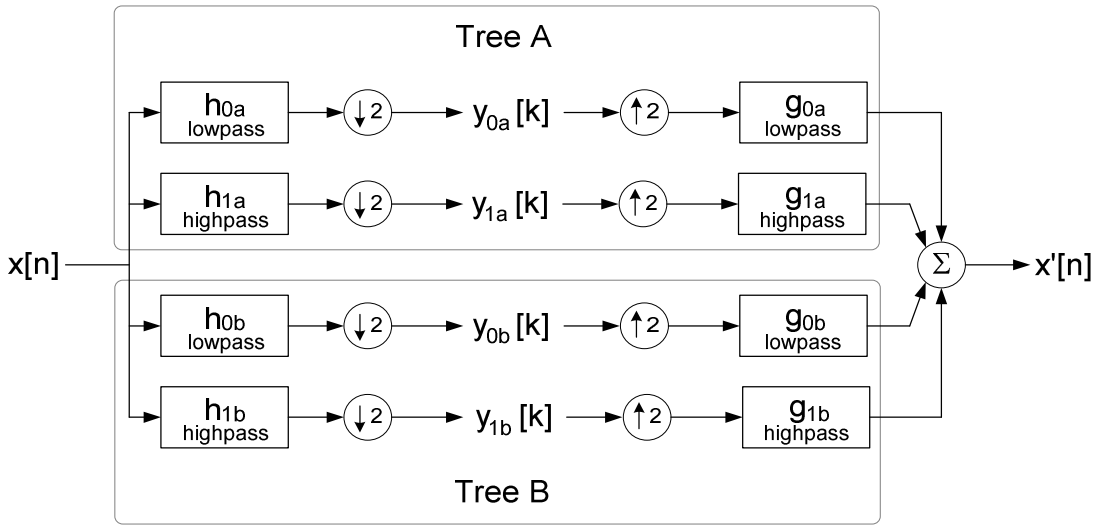


Figure 3.1 Basic configuration of the dual-tree filtering approach used to obtain the DT CWT coefficients (for a real one-dimensional signal $x[n]$).

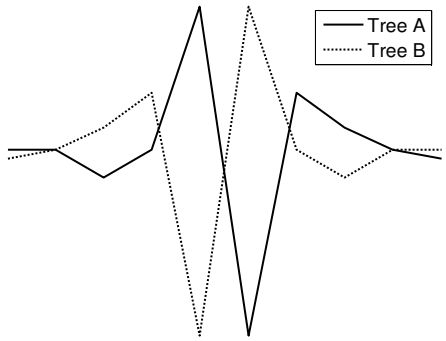


Figure 3.2 Typical impulse responses of the high pass decimation filters for each of the filter trees.

For two dimensional signals, DT CWT requires a 4:1 increase in the number of coefficients and provides approximate shift invariance in both the horizontal and vertical directions. While 2D DWT produces three subbands at each level, corresponding to LH,

HH and HL filtering (0^o, 45^o, and 90^o, respectively), 2D DT CWT produces six subbands that correspond to the outputs of six directional filters oriented at angles of $\pm 15^o$, $\pm 45^o$, and $\pm 75^o$. Fig. 3.3(a) shows the two dimensional impulse responses of the reconstruction filters in 2D DT CWT. If the level (or *scale*) of decomposition is denoted by s and the *direction* of the filter is denoted by d then the set of band-pass complex wavelet coefficients at level s can be written as

$$y_{s,d}(u_s, v_s) = m_{s,d}(u_s, v_s)\, e^{j\theta_{s,d}(u_s, v_s)} \tag{3.1}$$

$$\text{for} \quad d = 1, \ldots, 6$$
$$u_s = 0, \ldots, \frac{N}{2^s} - 1$$
$$v_s = 0, \ldots, \frac{M}{2^s} - 1$$

where N and M are the dimensions of the video frame in pixels. The variables u_s and v_s specify the location of the complex coefficients in each subband. Fig. 3.3(b) shows a wavelet-type output structure for the six directional subbands at each level of DT CWT.

Since the watermarking application considered is *playback control*, the decoder must decide whether or not the watermark is present. This means that a watermark does not need to carry any other information: its presence or absence is all that is required. For example, if a watermark is detected by a compliant DVD player, the copy will be considered of illegal origin and the DVD player will not play the movie.

3.2.2 Creating the watermark

We first create a vector of length six whose elements are 1's and -1's. This vector will be the original watermark $\mathbf{w}_o$. An example of such a watermark is $\mathbf{w}_o$ = [1, -1, 1, -1, 1, -1]. To make the watermark difficult to detect by an attacker, we pseudo-randomly

reorder the elements of the watermark $\mathbf{w}_o$ before inserting them into a frame. This pseudo-random reordering process is achieved by utilizing a key K that is only known to the encoder and decoder. We call this reordered vector $\mathbf{w}$, and this is the watermark that will be embedded in the video frames.

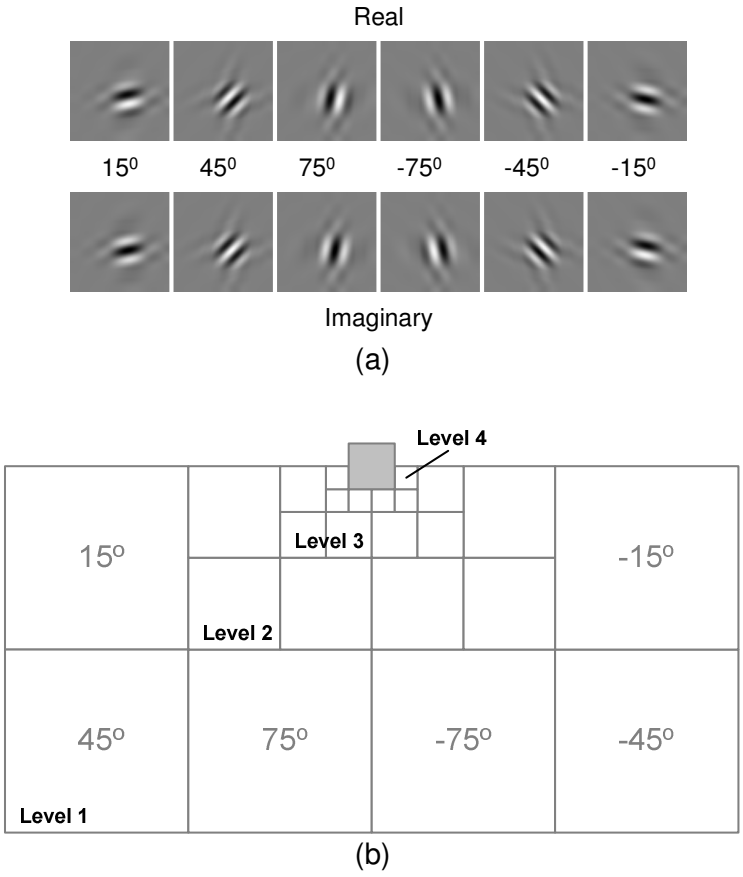


Figure 3.3 (a) Two dimensional impulse responses of the reconstruction filters in a DT CWT; (b) Structure of the DT CWT coefficients for a four-level decomposition.

The watermark detector must maintain synchronization with the reordering process in order to extract the watermark. Maintaining synchronization becomes difficult if frames are dropped. To overcome this problem we use the same reordering key for β

consecutive frames, keeping in mind that β should be small enough so that an attacker cannot detect and remove the watermark by averaging frames and long enough so that if some frames are dropped the watermark can still be detected. The DT CWT is computed on every frame and the watermark is embedded in some carefully chosen coefficients.

3.2.3 Embedding the watermark

For each frame, a four-level DT CWT is applied in order to find the coefficients. We then find the minimum of the magnitude of the level-3 coefficients across all six directions and construct the 2D array $z(u_3, v_3)$ given by

$$z(u_3, v_3) = \min_d \left[m_{3,d}(u_3, v_3) \right] \tag{3.2}$$

for $d = 1, ..., 6$.

We apply a low-pass filter to this array and sub-sample by a factor of 2 in each direction to create the array $p(u_4, v_4)$ with the same dimensions as the level-4 sub-bands. We then find the values in $p(u_4, v_4)$ that exceed an embedding threshold $\tau > 0$. The watermark **w**, which has been reordered using the key K, will be added to the magnitudes of the level-4 coefficients at these locations. As the watermark is added, the magnitude of the DT CWT coefficients in each direction is modified with one of the six elements of **w** in the following manner:

$$m_{4,d}(u_4, v_4) \leftarrow \begin{cases} m_{4,d}(u_4, v_4) + \alpha \, \mathrm{w}_d & \text{if } p(u_4, v_4) > \tau \\ m_{4,d}(u_4, v_4) & \text{otherwise} \end{cases} \tag{3.3}$$

for $d = 1, ..., 6$.

The scaling factor α is a positive number that is used to control the strength of the embedded watermark. It is obtained in the following way:

$$\alpha = \frac{\left(R_{fix} - R_0\right)}{6}. \tag{3.4}$$

R_{fix} is the fixed robustness strength that we wish to apply to every frame. R_0 is the original strength of the frame, which is computed before embedding the watermark:

$$R_0 = \sum \left\langle m_4(u_4, v_4), \mathbf{w} \right\rangle \quad \text{if } p(u_4, v_4) > \tau \tag{3.5}$$

Finally, the video frame is reconstructed by performing the inverse DT CWT using the watermarked coefficients. The watermark embedding process is illustrated in Fig. 3.4.

It should be noted that this approach is content-dependent since knowledge of the video frames is used in order to embed the watermark. The procedure for effectively finding the coefficients to be watermarked selects the regions of the video frame which have some energy in all directions. Thus this method relies on the position of the edges contained in the frame, rather than on pixel positions, to embed (and locate) the watermark. This makes the scheme inherently robust to geometric attacks. By avoiding the embedding of the watermark in areas with low texture, the scheme also provides some level of masking by content.

3.2.4 Detecting the watermark

The strength of the watermark in a video sequence is found by taking the inner product of the magnitude of selected level-4 coefficients and the original watermark. The

strength of the detected watermark R is recalculated every time $p'(u_4, v_4)$ has a value greater than τ:

$$R \leftarrow \begin{cases} R + \langle m'_4(u_4, v_4), \mathbf{w} \rangle & \text{if } p'(u_4, v_4) > \tau \\ R & \text{otherwise} \end{cases} \tag{3.6}$$

where $p'(u_4, v_4)$ and $m'_4(u_4, v_4)$ are calculated as in the encoder but using the coefficients from the suspected illegal copy of the video. Since the key K is known by the decoder, the reordering process that was applied during encoding can be utilized again to obtain the watermark vector $\mathbf{w}$. The watermark is detected using λ sets of β frames each so that the strength of the watermark in any one frame is very small but increases gradually over a large number of frames.

Hence, if the decoder is applied to a watermarked sequence (and the right key is provided), the sum of the inner products of the original watermark and the selected magnitudes should result in some large positive value. This will be an indication of the presence of a watermark. Alternatively, if the decoder is applied to an un-watermarked video sequence it will compute the inner product of the original watermark and a vector with random values, since no watermark has been embedded. Therefore, after summing over λ sets of β frames, a strength value close to zero is expected. The watermark detection process is illustrated in Fig. 3.5.

3.3 Experimental Results

To test the proposed watermarking method we employed five standard QCIF (176 × 144) video sequences (*Container*, *Hall Monitor*, *Mother and Daughter, News*, and *Suzie*). The watermarks were embedded in the luminance components of these sequences.

An example of a watermarked video frame can be seen in Fig. 3.6(a). 10 tests were performed for each sequence, using a different key K each time. Performance evaluations showed that the best results were obtained by setting β to 10 and λ to 50. The threshold τ was set to 8 and R_{fix} was set to different values each time in order to keep the average picture fidelity of the watermarked frames constant (at an average PSNR value of 42.5 dB). This parameter can be modified by choosing different values for τ and R_{fix}. In order to have a better estimate of the performance of our method, we compared it with the scheme presented in [8]. This method applies the DWT to the video frames and the watermark is embedded by reordering some of the coefficients. Since each method measures the strength of the decoded watermark differently, all results were normalized by assigning the value of 100 to the highest decoded watermark on each scheme.

Having the right key for each tested video sequence, we applied the watermark decoder to both watermarked (W) and non-watermarked (NW) video sequences in order to measure the discrepancy of the obtained watermark strength values. When the obtained strength value is high, the decoder decides that a watermark is present. On the other hand, if the strength value is small, the decoder assumes that no watermark was embedded. Table 3.1 shows the results for our DT CWT method. In order to fit the results in one table, the 10 watermark strength values obtained from a single video sequence (using different keys) were averaged. As expected, when a watermark is not present (NW) the detected strength is small (an average value of 5), since there is no hidden information in the video. However, we observe that for the sequences that were watermarked (W), the average detected strength is 82.

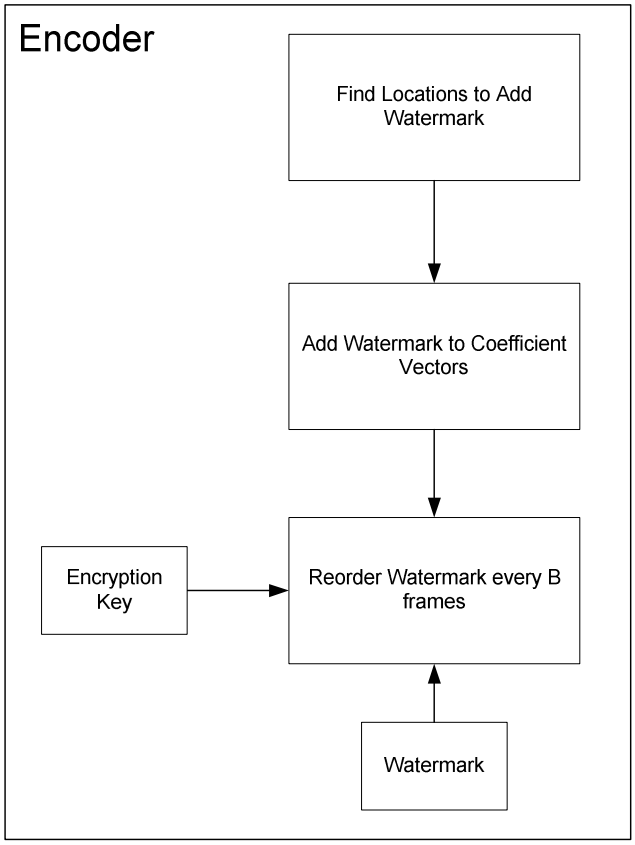


Figure 3.4 The watermark embedding process.

The results obtained with the DWT scheme are similar to those of our method when it comes to detecting non-watermarked and watermarked sequences, as can be seen in Table 3.2. We observe that the average detected strength for non-watermarked sequences has a mean of -2, whereas the mean strength for watermarked sequences is 100. Notice that, since the DWT scheme is not content-dependent, the watermark is embedded with exactly the same strength in every frame of every sequence. Although the resulting strength is always 100, the fidelity of the watermarked frame cannot be regulated. Next, we examined the effect that some attacks had on the watermarked video files.

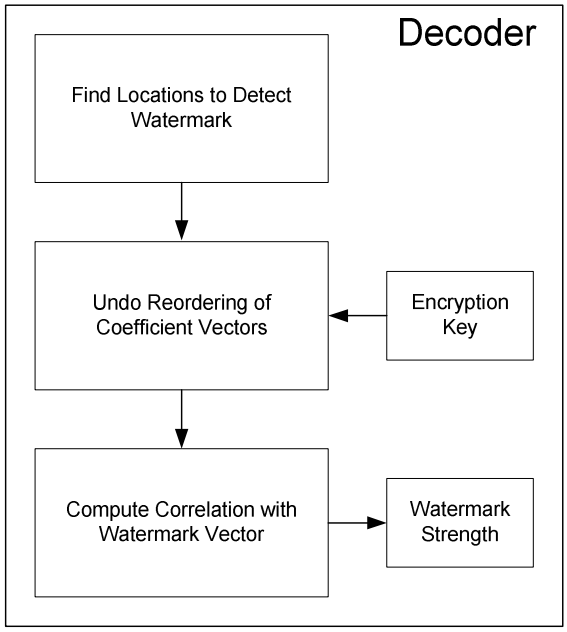


Figure 3.5 The watermark detection process.

3.3.1 Frame scaling and cropping

Each frame was scaled by 10% and then cropped back to the original frame size (176 × 144). This is illustrated in Fig. 3.6(b). The results for our method can be seen in Table 3.1. We observe that the average watermark strength value detected in the scaled and cropped (S+C) sequences is 62, almost a 25% decrease from non-distorted watermarked streams. Assuming that the decision threshold is set to a value of 30, we can still accurately differentiate non-watermarked from watermarked video sequences, even though the content has been scaled and cropped. Results from using the DWT scheme are illustrated in Table 3.2. We observe that this method is not robust to scaling and cropping, since the detected strength from S+C sequences completely overlaps with the detected strength of non-watermarked sequences NW. In this case, the presence of the watermark would never be detected.

3.3.2 Frame rotation

For this test, each frame was rotated to the left by 3^{o}, as can be seen in Fig. 3.6(c). Our method proves to be very robust to rotation as can be seen in Table 3.1. The recovered strength values from the rotated sequences R have decreased by almost 20% (average strength value is now 67). If the decision threshold is maintained at 30 for this approach, both false positive and false negative probabilities will be 0. Although there is no overlap between non-watermarked sequences NW and watermarked sequences that have gone through rotation R, the DWT scheme is not as robust to this attack. Almost 70% of the message's strength has been lost (see Table 3.2).

3.3.3 Compression

The video sequences were encoded using H.264/AVC. Every 15^{th} frame was set to be an I-frame and the rest were chosen to be P-frames. The quantization parameter QP for both I and P frames was set to 25 (this parameter can be any integer value between 0 and 51), which results in a compression ratio of around 60:1. Tables 3.1 and 3.2 show that both methods were able to successfully withstand lossy compression. The detected watermark strength was similar in both cases: 68 in the case of DT CWT and 74 for the DWT scheme.

3.3.4 Joint attack

The final experiment involved all the previous attacks together, as seen in Fig. 3.6(d). We first scaled the frames by 10% and subsequently cropped them back to their original frame size (176 × 144). We then rotated each frame by 3^{o} and finally, we used H.264/AVC to compress the video sequences, using the same compression ratio as

before. Again, if the same threshold of 30 is used for our method, it is possible to separate non-watermarked images from those that have been watermarked and gone through several distortions. This can be seen in Table 3.1. Results from the DWT scheme are presented in Table 3.2. It can be observed that this method is not robust at all to a joint attack, since the detected strength from degraded watermarked video sequences completely overlaps with the detected strength of non-watermarked sequences. In this case, the presence of the watermark would never be detected.

3.4 Conclusion

A watermarking scheme for playback control was presented. The embedded watermark is imperceptible (with an average frame PSNR of 42.5 dB) so as not to degrade the quality of the movie shown in cinemas. The watermark was proven to be robust to scaling, cropping, rotation, compression and a combination of all of these attacks. This is a severe test for a watermarking scheme and some schemes, such as the one presented in [8], are not robust to a joint attack as this one. Our method, however, showed to be highly robust against it resulting in no false detections when checking the presence of the watermark in a degraded (attacked) sequence.

The watermark detection process is blind since DVD players will not have a copy of the original movie. Furthermore, the watermark is secure since attackers cannot remove the watermark without knowing a secret key even if they have knowledge of the algorithm. Finally, since the strength of the watermark is very low in any of the sets of β frames, an attacker cannot detect and remove the watermark by averaging over multiple frames.

Our proposed method is simple to implement so as not to add to the cost and complexity of DVD players. All these characteristics make it suitable for playback control of digital video.

Table 3.1 Strength of the decoded watermark values obtained when using the DT CWT method. The tests involve non-watermarked (NW) videos, as well as watermarked (W) sequences that go through several attacks such as scaling and cropping (S+C), rotation (R), lossy H.264 compression (H264) and, finally, a joint attack (Joint).

VIDEO	WATERMARK STRENGTH					
	NW	W	S+C	R	H264	Joint
Container	1	86	46	66	78	42
Hall	7	79	66	69	55	49
Mother	5	77	58	62	65	47
News	6	73	60	63	53	47
Suzie	4	95	78	74	89	78
Average	**5**	**82**	**62**	**67**	**68**	**53**

Table 3.2 Strength of the decoded watermark values obtained when using the DWT method. The tests involve non-watermarked (NW) videos, as well as watermarked (W) sequences that go through several attacks such as scaling and cropping (S+C), rotation (R), lossy H.264 compression (H264) and, finally, a joint attack (Joint).

VIDEO	WATERMARK STRENGTH					
	NW	W	S+C	R	H264	Joint
Container	-2	99	1	20	82	-12
Hall	-3	100	-13	22	64	-19
Mother	-1	100	-10	22	75	-4
News	-3	100	2	47	78	-28
Suzie	-1	100	-6	45	71	-15
Average	**-2**	**100**	**-5**	**31**	**74**	**-16**

Figure 3.6 A QCIF video frame of the sequence *News*: (a) watermarked with the DT CWT method, (b) after scaling and cropping, (c) after rotation and (d) after a joint attack.

3.5 References

[1] E. I. Lin, A. M. Eskicioglu, R. L. Lagendijk and E. J. Delp, "Advances in digital video content protection," Proc. IEEE, vol. 93, pp. 171-183, Jan 2005.

[2] G. Doerr and J. Dugelay, "A guide tour of video watermarking," Signal Processing: Image Communication, vol. 18, pp. 263-282, 2003/4.

[3] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking.* San Francisco, Calif.: Morgan Kaufmann, 2002.

[4] K. Bernards, E. Kaltman, J. Feehery and G. Orsterberg. (2006, Movie pirates thwarted in attempt to camcord Mission Impossible: III in Los Angeles, Evansville, Taipei theaters. 2006(05/08), pp. 2. Available: http://www.mpaa.org/PressReleases.asp

[5] M. Wu, W. Trappe, Z. J. Wang and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," *Signal Processing Magazine, IEEE,* vol. 21, pp. 15-27, 2004.

[6] C. V. Serdean, M. A. Ambroze, M. Tomlinson and J. G. Wade, "DWT-based high-capacity blind video watermarking, invariant to geometrical attacks," IEE Proceedings -- Vision, Image & Signal Processing, vol. 150, pp. 51-58, Feb 2003.

[7] P. Bas, J. M. Chassery and B. Macq, "Geometrically invariant watermarking using feature points," IEEE Transactions on Image Processing, vol. 11, pp. 1014, 2002.

[8] P. W. Chan, M. R. Lyu and R. T. Chin, "A novel scheme for hybrid digital video watermarking: approach, evaluation and experimentation," Circuits and Systems for Video Technology, IEEE Transactions on, vol. 15, pp. 1638-1649, Dec. 2005.

[9] N. Kingsbury, "Image processing with complex wavelets," Philosophical Transactions. Mathematical, Physical, and Engineering Sciences, vol. 357, pp. 2543, 1999.

[10] P. Loo and N. Kingsburry, "Digital watermarking using complex wavelets," in International Conference on Image Processing, ICIP, 2000, pp. 29-32.

[11] P. Loo and N. Kingsburry, "Watermarking using complex wavelets with resistance to geometric distortion," in X European Signal Processing Conference (Eusipco), 2000.

[12] N. Terzija and W. Geisselhardt, "Digital image watermarking using complex wavelet transform," in *MM&Sec '04: Proceedings of the 2004 Workshop on Multimedia and Security,* 2004, pp. 193-198.

[13] N. Kingsbury, "The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters," *Proc. 8th IEEE DSP Workshop,* pp. Paper no. 86, 1998.

CHAPTER 4: A VIDEO WATERMARKING SCHEME BASED ON THE DUAL-TREE COMPLEX WAVELET TRANSFORM[3]

4.1 Introduction

Piracy, the practice of selling, acquiring, copying or distributing copyrighted materials without permission is a great concern to Hollywood studios and independent filmmakers. Although digital technology has brought many benefits to both the content creators and the public, it has also increased the ease by which movies can be pirated. This chapter addresses *Theatrical camcorder piracy* which is one of the most common ways of illegally copying a movie [1]. This method consists of someone taking a camcorder into a poorly supervised theatre and creating a copy of the movie that is being shown. These recordings are illegally duplicated, packaged and distributed all over the world. Consequently, the film appears in street markets just days after the theatrical release. This translates in a significant loss of revenue for the film producers.

Currently, some technical measures that prevent this practice are being developed. Camcorder jamming technologies, for instance, have been proposed to disable camcorders inside movie theatres [1]. For a little more than a decade, watermarking techniques have been designed to, among other purposes, control access to digital content [2]. In the case of a *playback control* application, the watermark embedded in the video sequence is designed to provide information on whether video players are authorized to

[3] A version of this chapter has been submitted for publication. Authors: L. E. Coria, M. Pickering, P. Nasiopoulos, and R. K. Ward. The authors would like to thank Dr. Nick Kingsbury for providing the software to perform the DT CWT transform operations.

display the content or not [3]. Compliant devices detect the watermark and obey the encoded usage restrictions.

A common problem that video watermarking algorithms have to endure is collusion attacks. For instance, if several copies of a video sequence are made and each one gets a different watermark for forensic purposes, attackers can join forces and create a new version of the video, for example, by averaging the frames of all the copies and, therefore, destroying the watermarks. Fortunately, for the type of application we are dealing with, this attack does not represent a concern since it is unlikely to happen. We are assuming that copies are being made using camcorders inside movie theatres. Theatres have different sizes and shapes and the cameras will be positioned in different places. Thus, every copy of a movie will be significantly different from any other existing copy and, therefore, averaging the frames of several video sequences that come from different sources will not produce a correct representation of the movie that has been pirated. Nevertheless, controlling access to media content that was re-recorded with a camera inside a movie theatre is a challenging problem. To begin with, the recorded video might be a slightly resized, rotated, cropped and noisy version of the original content. Furthermore, these copies are also subjected to video compression. Since the original content is not available during the decoding process (i.e., it is a blind procedure), extracting the watermark is not a straightforward task. The decoding process must, to a certain extent, be robust to some geometric distortions (rotation and scaling), as well as cropping, additive noise and lossy compression.

Several watermarking methods that are robust to common geometric distortions have been presented. For example, in [4], two watermarks are employed. The first one is

used to embed the message while the second one, a 1-bit watermark, is employed as a geometric reference. This reference watermark is embedded in the spatial domain which results in low robustness. Information hidden in the space domain can be easily lost to quantization, which makes the watermarking scheme vulnerable to lossy compression and other attacks. Once the reference watermark has been changed, the decoder assumes that there is no watermark embedded in the content and, therefore, does not search for the hidden message.

A content-based image watermarking method is offered in [5], where robustness to geometric attacks is achieved using feature points from the image. This scheme is shown to be successful to certain attacks, but it is computationally intensive and, therefore, may not be practical for real-time video applications.

Multiresolution analysis has been considered as an important tool for designing watermarks that can withstand geometric distortions. A method for image watermarking in the wavelet domain is presented in [6]. The watermark is applied to the discrete wavelet transform (DWT) coefficients of a sub-image. This sub-image is constructed from the original content using small blocks that are chosen via a chaotic map. Although the scheme is extremely robust to cropping, it does not provide an adequate solution for a rotation attack. A video watermarking method that also relies on wavelets is presented in [7]. In this case, the watermark is embedded in every video frame by applying DWT to the frames and replacing certain coefficients with the maximum or minimum value of their neighbouring coefficients. This scheme was proven to be robust to mild geometric attacks and high compression. However, since the embedding process is blind, the amount of distortion introduced in the frames cannot be controlled. In general, the

watermarks in the above algorithms have all been designed to convey information at a much higher rate than is necessary for playback control. Consequently, the amount of distortion introduced by these algorithms would almost certainly be unacceptable to the motion picture content providers.

Complex wavelets have also been employed to create watermarks that are robust to geometric distortions. The complex wavelet transform is an overcomplete transform and therefore creates redundant coefficients but it also offers some advantages over the regular wavelet transform. Two of the main features of complex wavelets are approximate shift invariance and directional selectivity [8]. These properties can be employed to produce a watermark that can be decoded even after the original content has undergone extensive geometric distortions. When dealing with signals that have more than one dimension, the Dual-Tree Complex Wavelet Transform (DT CWT) [8] is a particularly valuable solution since it adds perfect reconstruction to the list of desirable properties that regular complex wavelets have.

Most watermarking methods rely on embedding a pseudorandom pattern in the transform coefficients of the host image or frame. The same, however, cannot be achieved with the Dual-Tree Complex Wavelet Transform. DT CWT is a redundant transformation and, therefore, some components of the watermark might be lost during the inverse transform process [9]. In order to reduce this problem, a watermark that consists of a pseudorandom sequence constructed with valid CWT transform coefficients is proposed in [9] and [10]. A four-level DT CWT is applied to the original content and the watermark is added to the coefficients from levels 2 and 3. Although the ideas portrayed in these efforts show some potential, the robustness of the schemes is never

tested. Another watermarking method that uses DT CWT is presented in [11]. In this method, the content is also subjected to a four-level DT CWT decomposition and the watermark is added to the two highest levels using the spread spectrum technique. However, the decoding process is not blind and, therefore, the applications for this scheme are very limited. More recently, a blind decoding watermarking scheme that uses DT CWT to overcome geometric distortions was proposed by the authors in [12]. This method, designed specifically for playback control of digital content, is robust to rotation, cropping and H.264 compression. However, since the redundant nature of the transform is not taken into account, the watermark needs to be extracted from a considerable number of frames in order to reach the correct decision (*play* or *do not play* the content).

This chapter introduces a new watermarking method that is robust to lossy compression, additive noise, and some common geometric attacks such as rotation, scaling and cropping. In our method, the watermark is a random set of 1's and -1's. A one-level DT CWT is applied to this watermark and the coefficients of this transformation become the data that are embedded into the video sequence. Every frame of the original video sequence is transformed with a four-level DT CWT. The content is examined to determine how strongly the watermark embedding should be. Thus, the watermark coefficients are properly weighted and added to the coefficients of levels 3 and 4.

The remainder of the chapter is structured as follows. Section 4.2 depicts our method. Performance evaluations are presented in Section 4.3. Finally, Section 4.4 offers the conclusions.

4.2 Proposed Method

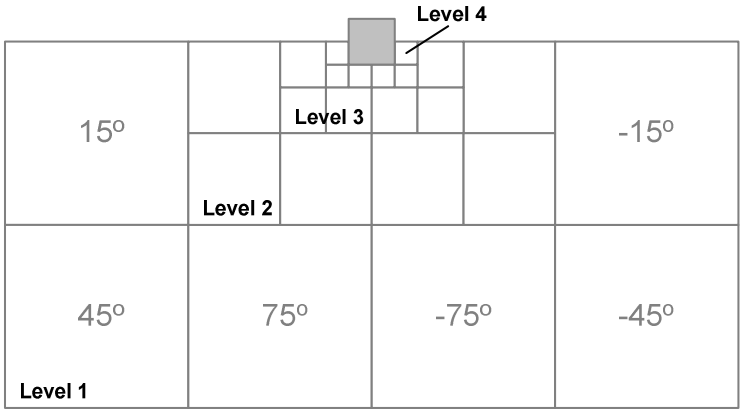
4.2.1 Creating the watermark

The proposed method uses the Dual-Tree Complex Wavelet Transform (DT CWT). For a brief introduction to DT CWT, please refer to subsection 3.2.1 of this thesis. A more detailed introduction can be found in [13]. Fig. 4.1(a) shows a wavelet-type output structure for the six directional subbands at each level of DT CWT and Fig. 4.1(b) describes the notation that will be used throughout this chapter.

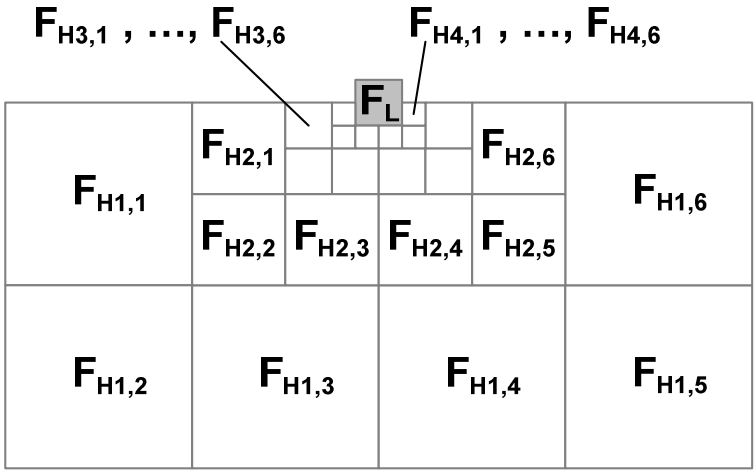
In our method, the watermark is inserted in every frame of the video sequence. To ensure robustness the watermark will be embedded in the coefficients of the higher decomposition levels. In our implementation, the watermark is embedded in levels 3 and 4 of a 4-level DT CWT decomposition. The watermark is a 2D array that is sixty-four times smaller than the video frame where it will be embedded (i.e. its height and width are one eighth of the frame's height and width, respectively). The watermark **w** is a pseudorandom sequence of 1's and -1's. It is created using a key

$$K = K_o + K_F,$$

where K_o is a constant (positive integer) provided by the user. K_F is a positive integer number that changes every β frames according to some formula. The use of the same K for β consecutive frames offers some robustness to temporal synchronization attacks. This is as long as β is small enough (so that an attacker cannot detect and remove the watermark by frame averaging) but long enough (so that if some frames are dropped the watermark can still be detected). Although this process provides limited robustness to temporal synchronization, more elaborate temporal synchronization strategies such as the ones presented in [14] could be later incorporated into our method.



(a)



(b)

Figure 4.1 Structure of the DT CWT coefficients for a four level decomposition: (a) For each level there are six subbands that correspond to the output of six directional filters oriented at angles of ±15°, ±45°, and ±75°; (b) The notation employed for the proposed method.

Usually, watermarking algorithms rely on the addition of a pseudorandom sequence (such as **w**) to the host content coefficients in some frequency domain. This approach, however, cannot be used in this exact fashion when working in the DT CWT domain. The reason is that DT CWT is a redundant transformation. Thus, some

components of the arbitrary pseudorandom sequence in the DT CWT domain may be lost during the DT CWT inverse transformation process. The lost information corresponds to the part of the pseudorandom sequence that lies in the null space of the inverse DT CWT [10]. Thus, to reduce this information loss, we embed the DT CWT coefficients of the watermark in the host content, *instead of* embedding the actual watermark. Thus, the one-level DT CWT transform is applied to the watermark **w** (as in Fig. 4.2). This results in a low-pass component, $\mathbf{W}_L$, and six subbands that contain the details, $\mathbf{W}_{H1}$, ... $\mathbf{W}_{H6}$. The coefficients of the 6 subbands form the data to be embedded in the host video frame.

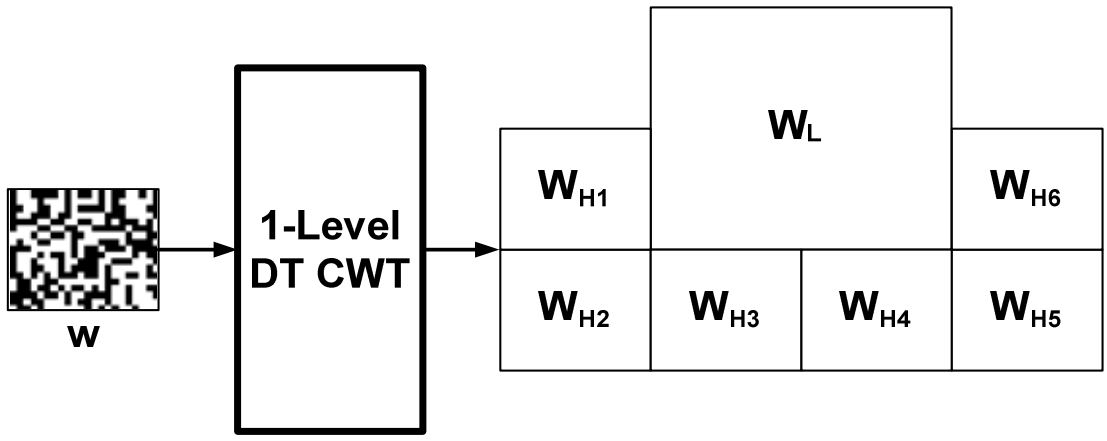


Figure 4.2 The level-1 DT CWT transform is applied to the watermark w. The coefficients of $\mathbf{W}_{H1}$, ... $\mathbf{W}_{H6}$ are the data to be embedded in the video frames.

4.2.2 Embedding the watermark

A four-level DT CWT is applied to every video frame. For each frame, the watermark is embedded in the coefficients of level 3 and level 4 since coefficients at finer levels are not robust to compression. Please note that the number of coefficients in level

one of the watermark transform is equal to the number of coefficients in level 4 of the frame's transform. This means that the data will be embedded several times and in different places. The embedding strength is decided based on information from the frame's coefficients of level 2. The embedding algorithm is now described in detail.

4.2.2.1 Perceptual masks

Robustness to compression is increased when the watermark is embedded in the frame's coefficients that are located in the highest levels of the DT CWT transform. This process however might significantly decrease the content's fidelity since the human visual system is very susceptible to changes in the low frequencies. Better results can be achieved by the prior examination of the content, i.e. *before* making any decisions on how strongly the watermark embedding should be. This can be done by using perceptual masks. These masks provide information on how much the magnitude of the host coefficients can be altered without the watermark becoming visible. In our approach, information from level 2 of the DT CWT decomposition is used to create a rough representation of the magnitudes of the higher-level-coefficients (levels 3 and 4). This coarse information will be used to create the perceptual masks. A large coefficient can endure a more significant change than a small one. Thus, these perceptual masks provide an estimate of the strength that can be used to embed the watermark in every coefficient of levels 3 and 4. The elements in these masks will be used as weights during the embedding process. Since the watermark is not embedded in the coefficients of level 2, the masks can be retrieved at the decoder without losing any information (provided the video frames have not been distorted in anyway).

Considering one frame at a time, a perceptual mask is created for each of the six subbands of level 3. In order to obtain these masks we apply a low-pass filter to every level 2 subband $\mathbf{F}_{H2,1}$, ..., $\mathbf{F}_{H2,6}$ and then down-sample the resulting arrays by a factor of 2. The elements of these arrays are divided by a step value Δ and then rounded to the next higher integer value (this operation is represented by the symbol $\lceil x \rceil$).The resulting arrays $\mathbf{mask}_{3,1}$, ..., $\mathbf{mask}_{3,6}$ have the same dimensions as the level 3 subbands. This process is described as:

$$\mathbf{mask}_{3,d} = \left\lceil \frac{\left[(\downarrow 2)(\mathbf{F}_{H2,d} * \mathbf{h}_{LP})\right]}{\Delta} \right\rceil \qquad \text{for } d = 1, 2, ..., 6. \tag{4.1}$$

where $\mathbf{h}_{LP} = \begin{bmatrix} \frac{1}{4} & \frac{1}{4} \\ \frac{1}{4} & \frac{1}{4} \end{bmatrix}$.

The masks for the level 4 subbands are created as follows. The low-pass filter $\mathbf{h}_{LP}$ is applied to every level 2 subband $\mathbf{F}_{H2,1}$, ..., $\mathbf{F}_{H2,6}$ and then the resulting arrays are down-sampled by a factor of 2. The same process of low-pass filtering and down-sampling is applied again to these arrays. The elements of the resulting arrays are divided by the step value Δ and then rounded to the next higher integer value. These new arrays become the masks $\mathbf{mask}_{4,1}$, ..., $\mathbf{mask}_{4,6}$ for the level 4 subbands and have the same dimensions as these subbands. The process is described in (4.2) and illustrated in Fig. 4.3.

$$\mathbf{mask}_{4,d} = \left\lceil \frac{\left[(\downarrow 2)\left[(\downarrow 2)(\mathbf{F}_{H2,d} * \mathbf{h}_{LP})\right] * \mathbf{h}_{LP}\right]}{\Delta} \right\rceil \quad \text{for } d = 1, 2, ..., 6. \tag{4.2}$$

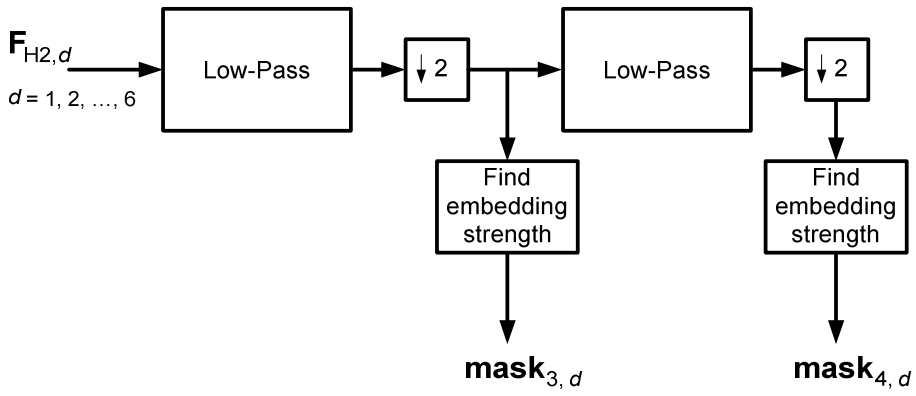


Figure 4.3 Construction of the masks for coefficients from levels 3 and 4 are obtained from level 2 subbands.

4.2.2.2 Adding the watermark

For each frame, the watermark's high frequency coefficients $\mathbf{W}_{H1}$, ..., $\mathbf{W}_{H6}$ are added to the magnitudes of the coefficients of level 3 and level 4 ($\mathbf{F}_{H3,1}$, ..., $\mathbf{F}_{H3,6}$ and $\mathbf{F}_{H4,1}$, ..., $\mathbf{F}_{H4,6}$, respectively). Since the number of watermark coefficients is the same as the number of coefficients of level 4, the data will be embedded in this level by weighting the watermark coefficients with the corresponding mask and multiplying by a scalar factor α. Then, the resulting array is added to the magnitudes of the frame's level 4 coefficients. In the case of level 3, the watermark coefficients in every subband are four times the frame's coefficients in that subband. The watermark coefficients are embedded by adding their magnitudes to those of each corresponding mask that is weighted and multiplied by the scalar α. The resulting array is added to the magnitudes of the level 3 coefficients. These are described next:

$$\mathbf{F}_{W3,d} = \left[\left| \mathbf{F}_{H3,d} \right| + \alpha \cdot \left(\mathbf{mask}_{3,d} \bullet \begin{bmatrix} \mathbf{W}_{Hd} & \mathbf{W}_{Hd} \\ \mathbf{W}_{Hd} & \mathbf{W}_{Hd} \end{bmatrix} \right) \right] \bullet \angle \mathbf{F}_{H3,d} \tag{4.3}$$

for d = 1, 2, ..., 6.

$$\mathbf{F}_{\mathrm{W4},d} = \left[\left|\mathbf{F}_{\mathrm{H4},d}\right| + \alpha \cdot \left(\mathbf{mask}_{4,d} \bullet \mathbf{W}_{\mathrm{H}d}\right)\right] \bullet \angle\mathbf{F}_{\mathrm{H4},d} \tag{4.4}$$

for d = 1, 2, ..., 6.

where $\left|\mathbf{F}_{\mathrm{H3},d}\right|$ and $\left|\mathbf{F}_{\mathrm{H4},d}\right|$ are the 2D arrays formed of the magnitudes of the complex elements of $\mathbf{F}_{\mathrm{H3},d}$ and $\mathbf{F}_{\mathrm{H4},d}$ as follows:

$$\left|\mathbf{F}_{\mathrm{H3},d}\right| = \begin{bmatrix} \left|F_{\mathrm{H3},d}(0,0)\right| & \cdots & \left|F_{\mathrm{H3},d}\left(0,\frac{M}{8}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F_{\mathrm{H3},d}\left(\frac{N}{8}-1,0\right)\right| & \cdots & \left|F_{\mathrm{H3},d}\left(\frac{N}{8}-1,\frac{M}{8}-1\right)\right| \end{bmatrix} \tag{4.5}$$

$$\left|\mathbf{F}_{\mathrm{H4},d}\right| = \begin{bmatrix} \left|F_{\mathrm{H4},d}(0,0)\right| & \cdots & \left|F_{\mathrm{H4},d}\left(0,\frac{M}{16}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F_{\mathrm{H4},d}\left(\frac{N}{16}-1,0\right)\right| & \cdots & \left|F_{\mathrm{H4},d}\left(\frac{N}{16}-1,\frac{M}{16}-1\right)\right| \end{bmatrix} \tag{4.6}$$

for d = 1, 2, ..., 6. M and N are the dimensions of the video frame in pixels.

$\angle\mathbf{F}_{\mathrm{H3},d}$ and $\angle\mathbf{F}_{\mathrm{H4},d}$ are 2D arrays formed with the phase of the complex elements of $\mathbf{F}_{\mathrm{H3},d}$ and $\mathbf{F}_{\mathrm{H4},d}$ as follows:

$$\angle\mathbf{F}_{\mathrm{H3},d} = \begin{bmatrix} e^{j\angle F_{\mathrm{H3},d}(0,0)} & \cdots & e^{j\angle F_{\mathrm{H3},d}\left(0,\frac{M}{8}-1\right)} \\ \vdots & \ddots & \vdots \\ e^{j\angle F_{\mathrm{H3},d}\left(\frac{N}{8}-1,0\right)} & \cdots & e^{j\angle F_{\mathrm{H3},d}\left(\frac{N}{8}-1,\frac{M}{8}-1\right)} \end{bmatrix} \tag{4.7}$$

$$\angle\mathbf{F}_{\mathrm{H4},d} = \begin{bmatrix} e^{j\angle F_{\mathrm{H4},d}(0,0)} & \cdots & e^{j\angle F_{\mathrm{H4},d}\left(0,\frac{M}{16}-1\right)} \\ \vdots & \ddots & \vdots \\ e^{j\angle F_{\mathrm{H4},d}\left(\frac{N}{16}-1,0\right)} & \cdots & e^{j\angle F_{\mathrm{H4},d}\left(\frac{N}{16}-1,\frac{M}{16}-1\right)} \end{bmatrix} \tag{4.8}$$

for $d = 1, 2, ..., 6$.

The symbol $\bullet$ denotes the element-wise matrix product and the value α is a strength parameter that is greater than zero and is used to control the fidelity impact of the watermark.

Once $\mathbf{F}_{W3,d}$ and $\mathbf{F}_{W4,d}$ are obtained, they replace $\mathbf{F}_{H3,d}$ and $\mathbf{F}_{H4,d}$ when computing the inverse DT CWT that provides the watermarked frame.

4.2.3 Decoding the watermark

The decoding process is blind, that is, the watermark is decoded without relying on any information from the original video file. Essentially, the decoder performs the inverse operations of the encoder. For every frame of the watermarked video sequence, the 4-level DT CWT is applied. The masks for levels 3 and 4 are obtained via (4.1) and (4.2), respectively. The arrays $\mathbf{imask}_{3,1}$, ..., $\mathbf{imask}_{3,6}$ and $\mathbf{imask}_{4,1}$, ..., $\mathbf{imask}_{4,6}$ are then obtained in the following way:

$$\mathbf{imask}_{s,d} = \begin{bmatrix} \frac{1}{mask_{s,d}(0,0)} & \cdots & \frac{1}{mask_{s,d}\left(0, \frac{M}{2^s} - 1\right)} \\ \vdots & \ddots & \vdots \\ \frac{1}{mask_{s,d}\left(\frac{N}{2^s} - 1, 0\right)} & \cdots & \frac{1}{mask_{s,d}\left(\frac{N}{2^s} - 1, \frac{M}{2^s} - 1\right)} \end{bmatrix} \tag{4.9}$$

for $s = 3, 4$ and $d = 1, 2, ..., 6$. M and N are the dimensions of the video frame in pixels.

The watermarked level 3 and level 4 coefficients $\mathbf{F}_{W3,d}$ and $\mathbf{F}_{W4,d}$ are multiplied by the **imask** arrays in order to compensate for the different weights associated with every coefficient during the watermark embedding process.

$$\mathbf{F}'_{Ws,d} = \mathbf{F}_{Ws,d} \bullet \mathbf{imask}_{s,d} \qquad \text{for } s = 3, 4 \text{ and } d = 1, 2, ..., 6. \qquad (4.10)$$

Next, **W'**, the level-1 DT CWT representation of the decoded watermark **w'** is obtained. Since the low-pass component $\mathbf{W}_L$ was not encoded in the watermarked video sequence, $\mathbf{W'}_L$ is considered to be an array of zeros. However, the six subbands with the details, $\mathbf{W'}_{H1}$, ... $\mathbf{W'}_{H6}$ can be estimated as follows.

$$W'_{Hd} = \begin{bmatrix} |F'_{W3,d}(0,0)| & \cdots & \left|F'_{W3,d}\left(0,\frac{M}{16}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F'_{W3,d}\left(\frac{N}{16}-1,0\right)\right| & \cdots & \left|F'_{W3,d}\left(\frac{N}{16}-1,\frac{M}{16}-1\right)\right| \end{bmatrix} + \begin{bmatrix} \left|F'_{W3,d}\left(0,\frac{M}{16}\right)\right| & \cdots & \left|F'_{W3,d}\left(0,\frac{M}{8}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F'_{W3,d}\left(\frac{N}{16}-1,\frac{M}{16}\right)\right| & \cdots & \left|F'_{W3,d}\left(\frac{N}{16}-1,\frac{M}{8}-1\right)\right| \end{bmatrix} +$$

$$\begin{bmatrix} \left|F'_{W3,d}\left(\frac{N}{16},0\right)\right| & \cdots & \left|F'_{W3,d}\left(\frac{N}{16},\frac{M}{16}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F'_{W3,d}\left(\frac{N}{8}-1,0\right)\right| & \cdots & \left|F'_{W3,d}\left(\frac{N}{8}-1,\frac{M}{16}-1\right)\right| \end{bmatrix} + \begin{bmatrix} \left|F'_{W3,d}\left(\frac{N}{16},\frac{M}{16}\right)\right| & \cdots & \left|F'_{W3,d}\left(\frac{N}{16},\frac{M}{8}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F'_{W3,d}\left(\frac{N}{8}-1,\frac{M}{16}\right)\right| & \cdots & \left|F'_{W3,d}\left(\frac{N}{8}-1,\frac{M}{8}-1\right)\right| \end{bmatrix} +$$

$$\begin{bmatrix} |F'_{W4,d}(0,0)| & \cdots & \left|F'_{W4,d}\left(0,\frac{M}{16}-1\right)\right| \\ \vdots & \ddots & \vdots \\ \left|F'_{W4,d}\left(\frac{N}{16}-1,0\right)\right| & \cdots & \left|F'_{W4,d}\left(\frac{N}{16}-1,\frac{M}{16}-1\right)\right| \end{bmatrix} \qquad (4.11)$$

for $d = 1, 2, ..., 6$. M and N are the dimensions of the video frame in pixels.

The inverse DT CWT is applied and the resulting 2D array **w'** is correlated with the original watermark **w**, which can be obtained via the same process that was used at the encoder. The correlation between **w'** and **w** is computed for every frame and the

resulting values are added until a certain amount of frames is reached (a hundred or more is recommended). When a watermark is decoded from every frame, the added correlation will be a relatively high positive number when compared to the correlation value obtained after an un-watermarked video sequence has gone through the decoder. By looking at the resulting strength value, a decision can be made at the decoder as to whether the video sequence that has gone through the decoding process has a watermark embedded.

4.3 Experimental Results

To test the proposed watermarking method we employed ten QCIF (176 × 144) video sequences. Each consists of 300 frames. Five of them were the standard video files *Container*, *Hall Monitor*, *Mother and Daughter, News*, and *Suzie*. The other five were formed using different short frame sequences from these standard video files. The formed sequences had a change of scene every 30 frames. We refer to the formed sequences as *Mix 1*, *Mix 2*, ..., *Mix 5*. Five different watermarked video sequences were created for each of the 10 sequences, by using a different key K_o each time. For every 15 frames different K_F's were used. The watermarks were embedded in the luminance components. Tests were performed on each of the 10 × 5 = 50 sequences. For each sequence, 300 frames were used to determine the strength of the watermark. For our tests, α was set to 15 so that the average PSNR of the watermarked frames was 41 dB. Computer experiments showed that setting Δ to 25 yielded the best results.

In order to study the performance of our method we compared our results against two algorithms that employ the regular discrete wavelet transform (DWT). These two methods also have the advantages over others in that they are blind and computationally

less demanding. Thus the original content is not needed to retrieve the watermark and the frames do not need to be geometrically restored before detecting the watermark. The first method we use as reference is basically the same algorithm as proposed in this chapter except DWT replaces DT CWT. We will refer to this method as DWT1. The second method is the one presented in [7], which is also based on DWT. In this method, which we denote as DWT2, video frames are watermarked by replacing the values of some coefficients of the frames by the highest or lowest values of the neighbour coefficients (depending if a 0 or a 1 is being embedded). The average PSNR of the sequences watermarked with these methods was also set to 41 dB.

The watermark decoder measures the correlation between the transform coefficients of every frame and the coefficients of the watermark. The measured correlation for a particular frame will be very small but, over several frames (300 for these tests), the accumulation of the correlations will provide an indication as to whether or not the video sequence is watermarked. When the normalized value of the correlation strength over 300 frames is between ±10, the decoder deduces that no watermark has been embedded. Otherwise the decoder decides that a watermark is present. Fig. 4.4 illustrates the difference between the strength values obtained for watermarked and non-watermarked sequences.

In order to compare the performance of the different methods, the correlation strength values were normalized: for every method, the highest correlation (after 300 frames) was set to 100 and the other values were proportionally modified. For sequences which did not suffer any distortion attack, the results obtained using the three methods are shown in Table 4.1. The normalized correlation strength values are displayed for both

watermarked (W) and non-watermarked (NW) sequences. To fit the information in one table, the 5 tests performed for every video sequence were averaged.

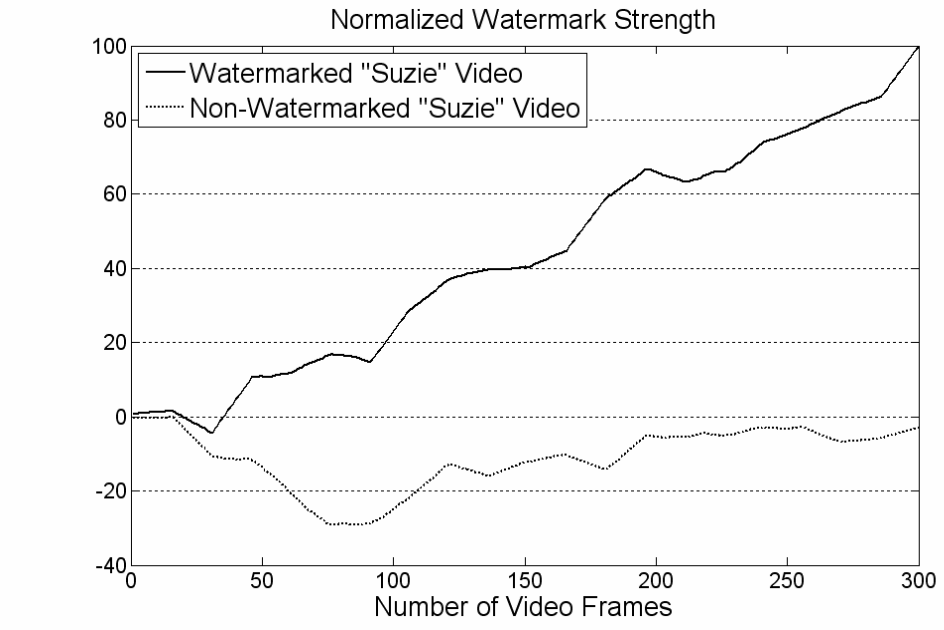


Figure 4.4 After 300 frames, the decoder detects a high watermark strength value for the watermarked Suzie sequence while the strength decoded for the non-watermarked Suzie video is close to zero.

The three watermarking schemes show similar results in sequences which did not undergo any distortion attack. There is a significant difference between the normalized correlation values obtained from watermarked sequences and the values obtained from non-watermarked ones. For each method, when the decoder attempts to extract a watermark from a non-watermarked video sequence the resulting correlation values are very low (an average of 1, -1 and 0 for DT CWT, DWT1 and DWT2, respectively) since the information in the video frames is not correlated with the actual watermark. In

contrast, correlation values obtained from watermarked sequences are constantly high (average values of 94, 87 and 100 for DT CWT, DWT1 and DWT2, respectively).

Table 4.1 Comparison of normalized correlation values obtained by the three watermarking methods: DT CWT, DWT1 and DWT2.

VIDEO	DT CWT		DWT1		DWT2	
	NW	W	NW	W	NW	W
Container	7	98	-5	100	1	99
Hall	-1	91	4	79	2	99
Mother	0	93	-3	84	-5	99
News	0	100	-1	74	0	100
Suzie	0	87	1	100	-1	100
Mix 1	3	95	1	91	-2	100
Mix 2	2	95	-3	80	1	100
Mix 3	0	93	-6	88	-2	99
Mix 4	0	94	3	92	-1	100
Mix 5	0	92	1	84	2	100
Mean	**1**	**94**	**-1**	**87**	**0**	**100**

We then tested the robustness of our method to common distortions. In one experiment, watermarks were decoded after the video sequences had gone through some scaling and cropping distortions. For the second test, the video sequences were rotated by a few degrees and the watermark was later decoded. We also tested the effects of additive Gaussian noise and lossy compression. Finally, all these distortions: scaling, rotation, cropping, additive noise and lossy compression were put together as a joint attack.

4.3.1 Frame scaling and cropping

We examined the robustness of all three methods when the watermarked sequences were subjected to scaling and cropping. Every video sequence was scaled up

by 5%, 10% and 15% using bicubic interpolation. The frames were later cropped to fit their original size (176 × 144). A visual example of this process can be seen in Fig. 4.5. Results are summarized in Table 4.2.

Table 4.2 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to scaling (by 5%, 10% and 15%) and cropping.

VIDEO	DT CWT						DWT1						DWT2					
	5%		10%		15%		5%		10%		15%		5%		10%		15%	
	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W
Container	9	69	5	41	3	26	0	41	0	9	0	-2	18	54	12	22	-14	-6
Hall	-4	60	-4	39	-7	22	1	34	1	7	0	0	19	36	18	18	-1	4
Mother	0	67	-1	42	-2	26	-1	25	0	5	0	1	8	50	1	0	-7	-12
News	-1	72	0	46	0	30	0	23	-1	6	0	1	-4	54	3	1	-4	-13
Suzie	1	66	2	43	1	28	0	31	0	1	1	-1	6	48	-19	5	-11	-5
Mix 1	2	67	0	42	-3	25	-1	31	-1	4	-1	-2	-1	49	-7	6	-15	-9
Mix 2	2	67	1	43	-2	25	0	27	0	5	1	0	-8	53	0	8	-7	-2
Mix 3	1	67	1	42	0	26	0	33	0	5	1	-1	12	41	1	8	-5	-6
Mix 4	-1	65	-2	40	-3	25	0	34	1	10	0	1	27	44	25	13	-1	-8
Mix 5	2	67	2	44	2	30	0	31	1	6	1	0	16	55	-4	11	-9	-6
Mean	**1**	**67**	**0**	**42**	**-1**	**26**	**0**	**31**	**0**	**6**	**0**	**0**	**9**	**48**	**3**	**9**	**-7**	**-6**

From these results we notice that only DT CWT is able to withstand a scaling and cropping attack, particularly for scales higher than 5%. For DT CWT, the higher the scaling of the video sequences, the lower the strength of the detected watermark. However, even after the frames have been scaled up by 15% and, more importantly, cropped to fit the QCIF format, the presence of the watermark is evident and in no risk of going undetected. Although DWT1 and DWT2 tolerate scaling by 5%, both methods lose track of watermarks when the sequences are scaled by 10% or more.

(a)

(b)

(c)

Figure 4.5 A watermarked frame of the sequence *Suzie* is scaled and then cropped: (a) 5%, (b) 10% and (c) 15% scaling.

4.3.2 Frame rotation

Robustness to frame rotation was then tested. Each frame was rotated counterclockwise by 3^{o}, 6^{o} and 9^{o}. Bilinear interpolation was employed and the resulting images were cropped to fit the QCIF format. An example of this attack can be seen in

Fig. 4.6. Table 4.3 shows comparisons of the performance by the three watermarking methods to this particular type of distortion.

Table 4.3 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to rotation (by 3^o, 6^o and 9^o) and cropping.

VIDEO	DT CWT						DWT1						DWT2					
	3^o		6^o		9^o		3^o		6^o		9^o		3^o		6^o		9^o	
	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W
Container	9	71	3	35	2	19	0	20	0	2	0	0	23	53	10	-10	4	-2
Hall	-2	61	-2	33	-4	15	0	18	0	1	0	-2	8	17	4	1	-4	14
Mother	0	62	-2	33	-2	17	-1	19	-1	-1	-1	-2	16	60	-13	1	-9	-2
News	-1	68	1	39	2	21	0	14	0	-5	0	-2	3	49	-20	2	-8	-5
Suzie	-1	56	-1	30	-2	16	0	16	0	0	0	2	-4	48	-5	0	8	-6
Mix 1	3	65	3	37	4	22	0	21	-1	-1	-1	2	23	43	1	1	11	-3
Mix 2	2	65	1	35	-1	18	0	14	0	-4	0	-5	-3	46	-24	-4	-18	4
Mix 3	0	63	2	35	3	21	0	21	0	-1	0	-2	23	38	-7	-7	-2	-5
Mix 4	-1	63	-6	28	-5	13	0	16	0	-1	0	-2	3	39	-2	1	-14	4
Mix 5	1	63	1	34	-5	13	1	16	0	2	0	3	1	61	7	3	14	-1
Mean	**1**	**64**	**0**	**34**	**-1**	**18**	**0**	**17**	**0**	**-1**	**0**	**-1**	**9**	**45**	**-5**	**-1**	**-2**	**0**

Once again, it can be observed that, for the case of DT CWT, the higher the degree of rotation, the lower the strength of the watermark correlation. Even though the watermark correlation strength is low when frames are rotated by 9^o, this value (18 on average) is high enough to be disassociated with a non-watermarked sequence. Regarding the other two methods, DWT1 and DWT2, it is evident that they are extremely fragile to rotation. None of these schemes was able to withstand a rotation attack higher than 3^o. The watermarks embedded with these schemes went completely undetected.

Figure 4.6 A watermarked frame of the sequence *Suzie* is rotated and then cropped: (a) 3º, (b) 6º and (c) 9º rotation.

4.3.3 Additive noise

We tested the robustness of the schemes to additive noise. Gaussian noise with a standard deviation of 5 was added to the frames. All three watermarking methods proved to be robust to additive noise. The average detected watermark strength values for DT

CWT, DWT1 and DWT2 is 90, 87, and 96, respectively. The results are shown in Table 4.4.

4.3.4 Compression

In order to test the robustness of the proposed scheme to compression we encoded the video sequences using H.264/AVC. Every 15^{th} frame was set to be an I-frame and the rest were chosen to be P-frames. The quantization parameter QP for both I and P frames was set to 15, which results in a compression ratio of around 40:1. In this instance, the three watermarking methods demonstrate to be robust to compression. The average detected watermark strength is 85, 77 and 93 for DT CWT, DWT1 and DWT2, respectively. Results are summarized in Table 4.4.

4.3.5 Joint attack

The final experiment involved all the previous attacks together. For this joint attack, we scaled the video frames by 5% and rotated them by 5^{o}. The frames were later cropped to fit their original size (176 × 144). Gaussian noise with a standard deviation of 5 was added and H.264/AVC was used to compress the video sequences (same compression ratio as before). An example of a video frame that has gone through this joint attack can be seen in Fig. 4.7. The performance of the three methods is shown in Table 4.5.

Results for DT CWT indicate that the method can successfully survive a joint attack. After 300 frames, the watermark correlation strength value averages 37. Considering the poor performance of DWT1 and DWT2 to the previous distortions, it is not surprising that these methods are not able to withstand a joint attack. Although these

approaches are very robust to noise and lossy compression, they can only endure mild geometric distortions.

(a)

(b)

Figure 4.7 (a) A watermarked frame of the sequence *Suzie*. (b) The same frame is subjected to a joint attack (rotating by 5°, scaling up by 5%, cropping, adding noise and using H.264 compression).

4.4 Conclusion

A new video watermarking algorithm for playback control that takes advantage of the properties of the dual-tree complex wavelet transform (DT CWT) is introduced. This transform maintains the advantages but avoids the shortcomings of regular wavelets. DT CWT provides important features such as perfect reconstruction, approximate shift

invariance and good directional selectivity. Our method relies on these characteristics to create a watermark that is robust to geometric distortions.

Table 4.4 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to lossy compression (H.264 with a QF of 15) and also to a joint attack (H.264 compression with a QF of 15, scaling up by 5%, rotating by 5º, and cropping back to QCIF).

VIDEO	DT CWT				DWT1				DWT2			
	Noise		H.264		Noise		H.264		Noise		H.264	
	NW	W	NW	W	NW	W	NW	W	NW	W	NW	W
Container	7	93	7	89	-5	99	0	88	1	97	-20	90
Hall	-1	87	-1	83	4	79	0	70	2	96	15	93
Mother	0	90	0	84	-3	84	-1	73	-5	97	-10	96
News	0	97	0	88	-1	74	0	64	0	96	-9	90
Suzie	0	82	0	79	1	100	0	88	-1	97	5	96
Mix 1	3	91	3	86	1	90	0	80	-2	97	3	93
Mix 2	2	91	2	86	-3	81	0	70	1	97	-13	95
Mix 3	-1	89	0	84	-6	88	0	77	-2	94	-26	91
Mix 4	0	89	0	84	3	91	0	81	-1	96	13	91
Mix 5	0	88	0	83	1	83	0	75	2	97	6	94
Mean	**1**	**90**	**1**	**85**	**-1**	**87**	**0**	**77**	**0**	**96**	**-4**	**93**

The watermark was embedded using information from the source content in order to keep distortion to a minimum (41 dB). The robustness of our method was tested against several attacks, which included lossy compression, additive noise, rotation, scaling, cropping and a joint attack, which involved a combination of all the previous distortions. Our method successfully detected the presence of the watermarks in all the corrupted video sequences. The joint attack was employed to simulate a video sequence that has been recorded from a movie screen with a handheld camcorder and then stored in a digital form.

Table 4.5 Comparison of normalized correlation values for three watermarking methods: DT CWT, DWT1 and DWT2. Watermarked and Non-Watermarked sequences are subjected to a joint attack (additive noise with a standard deviation of 5, rotating by 5º, scaling up by 5%, cropping back to QCIF, and H.264 compression).

VIDEO	JOINT ATTACK					
	DT CWT		DWT1		DWT2	
	NW	W	NW	W	NW	W
Container	7	39	0	3	16	-1
Hall	-3	36	1	1	9	-16
Mother	-3	34	0	0	-3	-14
News	1	42	1	0	-3	-2
Suzie	-1	32	0	-1	-7	-10
Mix 1	-1	35	-1	3	15	-10
Mix 2	0	37	0	-3	-11	-8
Mix 3	1	37	0	1	-3	-9
Mix 4	-2	35	1	2	8	-8
Mix 5	2	39	1	0	3	-8
Mean	**0**	**37**	**0**	**0**	**3**	**-9**

In order to compare the performance of our scheme and evaluate the advantages of using DT CWT as a watermarking tool, we subjected two DWT-based watermarking algorithms to the same fidelity standards and the same attacks. Although these methods were robust to compression and noise, they did not survive any significant geometric distortions.

Our proposed method is simple to implement; this is important when considering the added cost and complexity to DVD players. Furthermore, it is robust to geometric distortions and to lossy compression. All these characteristics make our algorithm suitable for the playback control of digital video.

4.5 References

[1] Motion Picture Association of America, *2007.* Available: http://www.mpaa.org/piracy.asp

[2] P. B. Schneck, "Persistent Access Control to Prevent Piracy of Digital Information," *Proceedings of the IEEE,* vol. 87, pp. 1239-1249, July, 1999.

[3] I. J. Cox, M. L. Miller and J. A. Bloom, Digital Watermarking. San Francisco, Calif.: Morgan Kaufmann, 2002.

[4] C. V. Serdean, M. A. Ambroze, M. Tomlinson and J. G. Wade, "DWT-based high-capacity blind video watermarking, invariant to geometrical attacks," *IEE Proceedings -- Vision, Image & Signal Processing,* vol. 150, pp. 51-58, Feb 2003.

[5] P. Bas, J. M. Chassery and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Transactions on Image Processing,* vol. 11, pp. 1014, 2002.

[6] Z. Dawei, C. Guanrong and L. Wenbo, "A chaos-based robust wavelet-domain watermarking algorithm," Chaos, Solitons & Fractals, vol. 22, pp. 47-54, 2004/10.

[7] P. W. Chan, M. R. Lyu and R. T. Chin, "A novel scheme for hybrid digital video watermarking: approach, evaluation and experimentation," *Circuits and Systems for Video Technology, IEEE Transactions on,* vol. 15, pp. 1638-1649, Dec. 2005.

[8] N. Kingsbury, "Image processing with complex wavelets," *Philosophical Transactions. Mathematical, Physical, and Engineering Sciences,* vol. 357, pp. 2543, 1999.

[9] P. Loo and N. Kingsburry, "Digital watermarking using complex wavelets," in *International Conference on Image Processing, ICIP,* 2000, pp. 29-32.

[10] P. Loo and N. Kingsbury, "Digital watermarking with complex wavelets," in IEE Seminar on Secure Images and Image Authentication, 2000, pp. 10/1-10/7.

[11] N. Terzija and W. Geisselhardt, "Digital image watermarking using complex wavelet transform," in *MM\&Sec '04: Proceedings of the 2004 Workshop on Multimedia and Security,* 2004, pp. 193-198.

[12] M. Pickering, L. E. Coria and P. Nasiopoulos, "A novel blind video watermarking scheme for access control using complex wavelets," in *International Conference on Consumer Electronics,* 2007, pp. 1-2.

[13] N. Kingsbury, "The dual-tree complex wavelet transform: a new technique for shift invariance and directional filters," *Proc. 8th IEEE DSP Workshop,* pp. Paper no. 86, 1998.

[14] E. T. Lin and E. J. Delp, "Temporal synchronization in video watermarking," Signal Processing, IEEE Transactions on [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on], vol. 52, pp. 3007-3022, 2004.

CHAPTER 5: CONCLUSIONS

5.1 Overall Significance of the Research

As described in the introductory chapter, piracy is a multifaceted problem that needs to be addressed by many disciplines. There are, for instance, legal, social and technical aspects that need to be examined. The technical aspects of the piracy problem have led to Digital Rights Management (DRM) strategies that deal with networking and content security. Watermarking is only one of the many elements that are necessary to assemble a DRM system capable of controlling access to digital data.

The research work presented in this thesis addresses some applications of the digital watermarking field. Their needs and challenges are explored and a few reliable and creative solutions are provided.

First, we consider those applications that require the embedding of a long message into the content (e.g. proof of ownership). Chapter 2 presents a fast high-capacity watermarking scheme that is robust to common distortions such as additive noise, valumetric scaling, low-pass filtering, and lossy compression.

The second type of application we focus on is video playback control. Many pirate DVDs are created by recording movies from poorly supervised theatres. The resulting video is usually a slightly rotated and cropped version of the original movie. Inserting a watermark that is robust to this type of distortions can prevent illegal DVDs from being displayed. In this kind of application, compliant video players search for the presence of the watermark, which is an indication that the content should not be

displayed. The scheme presented in chapter 3 embeds a watermark that is able to withstand lossy compression as well as scaling, cropping and rotation. Robustness to geometric distortions is attained due to the use of the Dual-Tree Complex Wavelet Transform (DT CWT). Approximate shift invariance and good directional selectivity, the two main features of DT CWT, are employed to construct this watermarking scheme.

DT CWT is however a redundant transformation and, therefore, any components of the watermark that lie in the null space of the inverse DT CWT are lost during the inverse transform process. Because of this, the proposed scheme requires to decode the watermark from many frames of the video sequence in order to reach a decision concerning the display of the content.

We address this problem in chapter 4. To reduce the information loss, we embed the DT CWT coefficients of the watermark in the host content, *instead of* embedding the actual watermark. The new scheme is also robust to lossy compression and geometric distortions. Moreover, the decoder only requires half the number of frames from the previous method to detect the watermark.

5.2 Potential Applications of the Research Findings

As previously mentioned, there are several applications for the proposed schemes. Devices that require embedding a long message (logo, serial number, text, etc.) can take advantage of the Even Codeword Distribution (ECD) method proposed in chapter 2. This algorithm is fast and robust. Also, using informed coding reduces the amount of distortion introduced in the host content.

- ECD can be employed for owner identification purposes. Owner identification refers to the practice of embedding a watermark that provides information about the owner of the content. If someone finds this content and wants to make proper use of it, the content's creator can be tracked via the watermark. An example of this application can be found in Digimarc's free software product MyPictureMarc [1]. This software allows any user to extract watermarks from images. If a subscription fee is paid, users are also allowed to embed watermarks into their images. When MyPictureMarc software recognizes a watermark, it contacts a central database over the Internet, and uses the watermark message as a key to find the contact information of the owner of the image.
- Proof of ownership [2] is another application where the proposed ECD scheme can be useful. In this case, a message is embedded into the content so that it can be used as proof of ownership if there is a dispute over who owns the rights to this content. For this application, the embedder and decoder are restricted to the public to increase the level of security. If there is an argument over the ownership of an image, only the rightful owner is able to extract the watermark by using the decoder. The watermark can be a logo or some text that identifies the owner.
- The proposed ECD scheme can also be employed for digital fingerprinting, which is a technique used to discourage people from illegally redistributing the digital data they have legally obtained. Fingerprinting involves the insertion of unique labels (watermarks) into the different copies of the content prior to its distribution [3]. When many copies of the same content are required, then long distinctive messages are needed.

- The watermarking schemes introduced in chapters 3 and 4 are intended for playback control of video content. In these algorithms, the player searches for a particular watermark hidden in the video. The presence of the watermark is an indication that the content should not be allowed to be displayed. The content can only be played if the decoder does not extract such a watermark. Conversely, if the watermark is found, the content is not displayed. This method is designed as a deterrent to camcorder piracy. This is a very common illegal practice whereby a movie theatre patron would tape a current movie with a camcorder in order to profit from the recorded content.

5.3 Discussion and Conclusions

5.3.1 Summary of contributions

Three watermarking algorithms are proposed in the preceding chapters. The contributions of the first method, introduced in chapter 2, are:

- High capacity. The use of spherical codewords reduces the distortion of the watermarked content and allows the design of high-capacity watermarking schemes [4]. In the proposed scheme, codewords used for watermarking are carefully constructed via an iterative algorithm that distributes the codewords evenly on a sphere. Once the codewords are chosen, the method embeds one message bit in every 8×8 pixel block of the image and uses information from only that particular block to ensure robustness and image fidelity.
- Robustness. This watermarking method is proven to withstand several image distortions such as additive noise, valumetric scaling, low-pass filtering, and lossy

compression. The Bit Error Rate (BER) was used to asses the performance of the proposed method. It was found that this scheme offers a lower BER than two other comparable methods.

- Low computational time. In addition to achieving a better performance, the computational demands of the proposed algorithm are significantly lower than those of similar schemes, making it more suitable for video applications where speed is a main concern. Furthermore, since every message bit is embedded independently from other bits in the complete message string, parallel processing can be employed. This makes the proposed method appropriate for real-time applications.

Chapter 3 presents a video watermarking algorithm designed for playback control, i.e., there is information hidden in the content that dictates compliant players as to whether or not to play the video. The scheme has the following features:

- Robustness. As this type of application arises when video is recorded e.g. by a camcorder, from a theatre or other type of screens, it is essential that the watermark is able to withstand geometric distortions as well as lossy compression. In order to address this issue, the proposed method relies on the orientation of edges rather than pixel positions to embed the watermark. This is achieved by using DT CWT, which provides important features such as perfect reconstruction, approximate shift invariance and good directional selectivity.
- High image fidelity. The information contained in the video frames is used in the embedding process. The watermark is only embedded in textured areas, i.e.,

regions where coefficient values from every direction are above a specified threshold. This helps ensure that embedding the watermark does not cause significant distortion to the host content.

- Easy to implement. The proposed method is simple to implement so as not to add to the cost and complexity of DVD players. Moreover, the decoding process is blind since the original content will not be available at the decoder end.

The scheme proposed in chapter 4 shares the same features with the method from chapter 3. Additionally, there is an advantage:

- Less data needed to detect the watermark. The main drawback of DT CWT is that it is a redundant transform. Therefore, any components of the watermark that lie in the null space of the inverse DT CWT are lost during the inverse transform process. To reduce the information loss, we embed the DT CWT coefficients of the watermark in the host content, *instead of* embedding the actual watermark. Because of this, the decoder only requires half the number of frames from the previous method to detect the watermark.

5.3.2 Three common advantages of the proposed schemes

All watermarking algorithms presented in this thesis share three advantages. First, they are all content-dependent. That is, the information contained in the host image is used in designing the embedded watermark. This strategy, known as informed embedding, helps ensure that embedding the watermark does not cause significant distortion to the host content.

The lack of computational complexity is another common element amongst these three schemes. For instance, in chapter 2 a careful design of the set of codewords employed to represent the watermark message bits allows the proposed scheme to rely on an informed embedding strategy that is much simpler and performs better than other algorithms. The algorithms from chapters 3 and 4 rely on the DT CWT. Although this is a redundant transformation that creates four times more coefficients than a regular wavelet transform, the embedding process is not time consuming. Furthermore, only a few seconds of a video file are needed to retrieve the watermark. Therefore, the decoder does not need to operate on the entire sequence in order to correctly recover the hidden data.

The third common feature is that all the proposed methods utilize blind decoders. This means that the original host content is not known during the decoding process. The only needed information in a blind decoding scheme is the watermark key. The key is a number employed as a seed that generates the pseudorandom pattern or patterns employed as watermarks. The obtained watermarks are then correlated with the watermarked content so as to retrieve the appropriate message.

The proposed schemes however have some differences. The algorithm presented in chapter 2, for instance, is a scheme with high watermark capacity, i.e. it can embed a long message, while the methods introduced in chapters 3 and 4 embed a one-bit watermark. This is because the decoding algorithms of the latter methods are only concerned with detecting the existence or non-existence of a watermark inside the video sequences. These watermarking systems are useful for applications such as playback control where the only objective is to find out if there is a restriction on displaying the video content. In this case, there is no need for a long message since there are only two

outcomes expected from the decoder: 'permit the video to be played' or 'do not permit the video to be played.'

The algorithm introduced in chapter 2 relies on informed coding and, therefore, is able to embed a long message into an image or a video frame (one message bit per every 8×8 pixel block). This method is intended for applications where a long binary message (representing a logo, a serial number or other information) is required. This scheme is not capable of withstanding geometric distortions, but it is robust to other attacks such as compression, additive Gaussian noise and low-pass filtering. This scheme can be useful for a video application that requires protection at the first or second stages of the video distribution process (i.e., when the data are either in the compressed or pixel domain).

5.4 Comments on Future Research

Although the proposed methods provide fine solutions to some existing problems, there are several ideas that can be further explored, in order to improve the present schemes.

5.4.1 Error-correcting codes

For instance, the robustness of the Even Codeword Distribution method of chapter 2 can be enhanced by applying error-correcting codes [5] to the watermark message. Error-correcting codes are typically implemented by increasing the length of the binary sequence used to represent a message. Therefore, the capacity of the watermarking system will decrease if an error-correcting code is implemented. Nevertheless, since the watermark capacity achieved with our scheme is high, the scheme can still accommodate long messages even with the inclusion of error-correcting codes.

5.4.2 Better perceptual models

Regarding DT CWT-based algorithms introduced in chapters 3 and 4, more research can be done in order to improve the perceptual masks employed in these schemes. By providing better perceptual models, the watermark can be embedded with a higher strength, which can result in higher robustness.

5.4.3 Temporal synchronization

As described in chapter 1, temporal synchronization is a challenge that needs to be also addressed. A simple operation such as frame dropping can desynchronize the data if the embedded watermark is not the same for all frames of the video sequence. On the other hand, having the same watermark for every frame makes the watermarking method vulnerable to a collusion attack in which the watermark can be found and then removed by frame averaging. The schemes from chapters 3 and 4 utilize a simple measure against these attacks in which the watermark is modified every certain amount of frames. Results can be improved, however, by combining our algorithm with a method that is specifically designed to resist temporal synchronization attacks, e.g. the one proposed in [6]. Alternatively, new ideas can be explored with respect to the temporal synchronization problem. For instance, some reliable features could be obtained from the source content in order to construct a set of keys. These keys can be employed as the seeds to generate the watermarks (which are pseudorandom arrays).

5.5 References

[1] Anonymous "Digimarc MyPictureMarc," 2007. Available: http://www.digimarc.com/mypicturemarc/

[2] I. J. Cox, M. L. Miller and J. A. Bloom, *Digital Watermarking.* San Francisco, Calif.: Morgan Kaufmann, 2002.

[3] M. Wu, W. Trappe, Z. J. Wang and K. J. R. Liu, "Collusion-resistant fingerprinting for multimedia," *Signal Processing Magazine, IEEE,* vol. 21, pp. 15-27, 2004.

[4] A. Abrardo and M. Barni, "Informed watermarking by means of orthogonal and quasi-orthogonal dirty paper coding," *Signal Processing, IEEE Transactions on [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on],* vol. 53, pp. 824-833, 2005.

[5] D. J. C. MacKay, Information Theory, Inference, and Learning Algorithms. Cambridge University Press, 2003.

[6] E. T. Lin and E. J. Delp, "Temporal synchronization in video watermarking," *Signal Processing, IEEE Transactions on [See also Acoustics, Speech, and Signal Processing, IEEE Transactions on],* vol. 52, pp. 3007-3022, 2004.