Digital watermarking for advanced image and video coding standards

Qiu, Gang

2005

Qiu, G. (2005). Digital watermarking for advanced image and video coding standards. Master's thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/3156>

<https://doi.org/10.32657/10356/3156>

Nanyang Technological University

Downloaded on 26 Apr 2023 18:46:23 SGT

Digital Watermarking For Advanced Image and Video Coding Standards

Qiu Gang

School of Electrical & Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfillment of the requirement for the degree of
Master of Engineering

2005

QA
76.9
.A25
Q1
2005

Statement of Originality

I hereby certify that the work embodied in this thesis is the result of original research done by me and has not been submitted for a higher degree to any other University or Institute.

.....

Date

.....

Qiu Gang

*To my wife and parents,
who always give me love and support.*

Acknowledgements

The utmost appreciation is forwarded to my supervisor, Prof. Pina Marziliano. Without her unfailingly patient and constant encouragement, I would have given up my graduate studies long time ago, to say nothing about the progress. Her wise guidance, assiduous instruction, as well as her remarkable sense of responsibility to her students deserve my lifelong respect. It is also my great pleasure to thank Prof. Ho Tung Shuen, and other professors of the Watermarking Group, for organizing the wonderful weekly brainstorm which aroused my interest in watermarking and gave me an insight on this topic.

The same appreciation and reverence are extended to Dr. Sun Qibin, Dr. Lin Xiao, and Dr. Susanto Rahardja. They give me tremendous help in my professional growth. I am truly blessed to be part of the Media Processing Department of the Institute for Infocomm Research (I²R). I am especially grateful to Dr. Sun for his inspiring guidance on my work and invaluable advice on this thesis.

I would also like to thank my fellow graduate students in NTU: Zhang Jin, Lu Rui, and Hao Yanyan and my colleagues in I²R: He Dajun and Zhang Zhishou. I have greatly benefited from the collaboration and discussion with them.

Abstract

Digital watermarking is a promising technology for *intellectual property rights* (IPR) protection of various digital media, including e-books, audio, games, photographs, videos, and others. It complements traditional encryption-based *digital rights management* (DRM) systems by hiding imperceptible but persistent identification information, known as *watermarks*, into the media to be protected. This makes it very attractive for copyright control, file tracing, broadcast monitoring, authentication, etc. Most of the time, media contents are sold and transmitted in compliant with international compression/coding standards. Researchers are therefore motivated to propose watermarking techniques aligned with popular coding standards. In this thesis, we will focus on two of the most common media types: *image* and *video*. New techniques for digital watermarking have been investigated, developed and implemented for the *JPEG-2000 image compression standard* and the *H.264/AVC video coding standard*.

JPEG-2000 adopts the *discrete wavelet transform* (DWT) for efficient compression instead of the *discrete cosine transform* (DCT) featured in the conventional JPEG standard. Moreover, new features like seamless scalability, progressiveness up to lossless compression, and region-of-interest coding are supported by the *Embedded Block Coding with Optimized Truncation* (EBCOT) algorithm. Therefore, we are motivated to take into account the DWT coefficients and EBCOT for watermarking applications and consequently propose

a novel blind image watermarking method. A new parameter *Watermarking embedding bit-rate* (WEBR) is introduced to quantitatively control the watermarking strength, and the resulting distortion can be estimated even before watermarking. The experimental results demonstrate that the proposed technique achieves high watermarking capacity while strikes a good balance between robustness and invisibility.

Whereafter, a set of watermarking techniques are proposed for the newest video coding standard H.264/AVC, which represents the state-of-the-art video coding methodology achieving superior coding efficiency to other existing standards. While the new features such as the 4 x 4 integer DCT and multiple block-size modes greatly improve the coding efficiency, they also impose great challenges for watermarking. Until now, to the best of our knowledge, few watermarking techniques have been proposed for this new standard. In this thesis, a *robust* watermarking technique and a *fragile* watermarking technique are proposed, which exploit respectively two of the most important H.264/AVC syntax elements — *DCT coefficients* and *motion vector differences* (MVD). Moreover, combining the advantages of the two techniques, a *semi-fragile* watermarking scheme is proposed for video authorization. All these techniques are well aligned with the H.264/AVC coding standard, and some features like Lagrangian optimization are supported in our design. The experimental results show that the proposed watermarking approaches are of low computational complexity and are capable of embedding and extracting a watermark in real-time during the H.264/AVC coding process. The experiments also demonstrate that they achieve desirable robustness, visual quality, and flexibility in dealing with low bit-rate videos (≤ 1 Mbit/s).

Contents

Acknowledgements	1
Abstract	iii
List of Figures	viii
List of Tables	xiii
List of Acronyms	xv
1 Introduction	1
1.1 The Need for Watermarking	1
1.2 Fundamentals of Digital Watermarking	4
1.2.1 Origin of Watermarking	4
1.2.2 Generic Digital Watermarking Scheme	6
1.2.3 Features of Digital Watermarking	9
1.3 Motivation and Objective	11
1.4 Major Contributions of the Thesis	13
1.5 Organization of the Thesis	14
2 Review of Digital Watermarking Techniques	17
2.1 Overview	17
2.2 Image Watermarking Techniques	18
2.2.1 Techniques in Uncompressed-domain	18

2.2.2	Techniques in Compressed-domain	20
2.3	Video Watermarking Techniques	24
2.3.1	Techniques in Uncompressed-domain	25
2.3.2	Techniques in Compressed-domain	26
2.4	The Future of Watermarking	31
3	Watermarking JPEG-2000 Images	33
3.1	Introduction	33
3.2	Review of JPEG-2000	35
3.2.1	JPEG-2000 Codec	37
3.2.2	The EBCOT Algorithm	40
3.3	The Proposed Method	42
3.3.1	System Overview	42
3.3.2	Watermarking by BPCS method	45
3.3.3	Embedding Strength Control	48
3.3.4	Embeddable Place Selection	51
3.3.5	Watermark Retrieval	53
3.4	Experiments and Results	54
3.5	Conclusions	66
	Appendix 3A Test Images	67
4	Watermarking H.264/AVC Videos	69
4.1	Introduction	69
4.2	Review of H.264/AVC	71
4.3	Watermarking DCT Coefficients	78
4.3.1	Watermark Formation	78
4.3.2	Perceptual Embedding	79
4.3.3	Retrieval by Frame Accumulation	81
4.3.4	Experiments and Results	83
4.4	Watermarking Motion Vectors	93

4.4.1	Data Embedding in Motion Vector Differences	94
4.4.2	Threshold Decision and Analysis	96
4.4.3	Experiments and Results	98
4.5	A Hybrid Scheme for Semi-fragile Watermarking	102
4.6	Conclusions	104
	Appendix 4A Test Video Sequences	106
5	Conclusions and Recommendations	109
5.1	Major Contributions and Conclusions	109
5.2	Recommendations for Future Work	112
	Bibliography	114
	Publications	129

List of Figures

1.1	The digital rights management (DRM) pillar model [2].	3
1.2	The world first paper money <i>jiaozi</i> (a) and its modern counterpart Chinese <i>yuan</i> (b). While ancient <i>jiaozi</i> bearing visible seals and secret marks, modern Chinese <i>yuan</i> uses paper watermark to prevent forgery.	5
1.3	Generic digital watermarking scheme	6
1.4	Generic watermark embedder, in which the perceptual analysis is optional, denoted by gray dashed line.	7
1.5	Generic watermark extractor, in which the original image/video and perceptual analysis, is optional, denoted by gray dashed line.	8
1.6	Progression of image and video coding standardization	12
2.1	An improved Koch's watermarking algorithm [56] embeds a bit by adapting relationship between three coefficients. The 'X' for watermark bit w denotes a invalid pattern.	22
2.2	A typical hybrid video coding scheme	26
2.3	Basic steps of the compressed-domain watermarking method . .	27
2.4	Typical process of motion prediction. The arrowheaded line is the motion vector. And the gray area is the search window. . . .	29
3.1	The block diagram of JPEG-2000 encoding and decoding processes, in which the ROI scaling and de-scaling are optional. . .	37

3.2	The dyadic decomposition of three levels and the corresponding label for every subband. For example, kHL indicates that during the k -th level of decomposition, the high-pass (H) filter is applied horizontally first, followed by a vertical low-pass (L) filter. . . .	39
3.3	The steps of a three-level wavelet decomposition for the image “Mandrill” using lossy 9/7 filter-bank. Recursively in each step, the kLL is decomposed to yield $(k+1)LL$, $(k+1)HL$, $(k+1)LH$, and $(k+1)HH$	39
3.4	Illustration of the coding process of the EBCOT Tier-1 encoder.	40
3.5	EBCOT quality layer formation. Each column represents the bitstreams generated for all the passes of a codeblock.	41
3.6	Watermark embedding procedure	43
3.7	Watermark retrieval procedure	43
3.8	Two binary patches of 8×8 samples with different complexities, where white represents 0 and black represents 1. (a) $\psi(P_a) = 72$, (b) $\psi(P_b) = 26$. $\psi(\cdot)$ is to compute the complexity.	46
3.9	An example of conjugation: $\mathcal{P}^* = \mathcal{P} \oplus \mathcal{W}_c$, where $\mathcal{P}^*$ and $\mathcal{P}$ represent the conjugated and the original patch respectively, and “ $\oplus$ ” denotes the conjugate operation. It easy to compute that $\psi(\mathcal{P}) = 26$ while $\psi(\mathcal{P}^*) = 86$. Note that $\psi(\mathcal{P}) = \psi_{max} - \psi(\mathcal{P}^*)$ hold true here. $\psi(\cdot)$ is to compute the complexity.	47
3.10	The sign-magnitude representation of the quantized DWT coefficients.	48
3.11	Mapping the embedding patches from the wavelet domain to the spatial domain.	52
3.12	Experiment results for “Cafe”. All of the above images are compressed at 2.5bpp. Figure (a) does not contain any watermark and Figure (b), (c), and (d) are embedded with 5120 bits at WEBR equal to 0.25, 0.5, and 1.5 bpp, respectively.	59

3.13	Experiment results for “Mandrill”. All of the above images are compressed at 5 bpp. Figure (a) is not watermarked and Figure (b), (c), and (d) are embedded with 4096 bits at WEBR equal to 0.5, 1.5, and 3 bpp, respectively.	61
3.14	The robustness against JPEG-2000 compression. Measured using normalized correlation between recovered and embedded message.	64
3.15	The robustness against JPEG compression. Measured using average normalized correlation for the watermark bits of a patch between recovered and embedded message.	65
3.16	The test images.. . . .	67
4.1	Block diagram of the H.264 <i>video coding layer</i> (VCL).	72
4.2	First <i>six</i> of the nine INTRA-4 x 4 luma prediction modes, where the samples {a, b, . . . , p} within the block are predicted by the neighboring samples {A, B, . . . , M}.	74
4.3	Block modes available for motion compensation in H.264/AVC. For each block, a motion vector is encoded. A 16x 16 macroblock can be divided down to 8 x 8 blocks, each of which can be further divided down to 4 x 4 blocks.	75
4.4	Watermark embedding scheme.	80
4.5	Comparison of the rate-distortion performance for H.264 encoding without watermarking (o) and with DCT domain watermarking (x). All the sequences are encoded with fixed QPs.	85
4.6	The 30th frame of the <i>Foreman</i> sequence (512 kbit/s, CIF-size) without watermarking and with DCT domain watermarking.	87
4.7	The 30th frame of <i>Stefan</i> (512 kbit/s, CIF-size) without watermarking (a) and with DCT domain watermarking (b).	87
4.8	The 30th frame of <i>Silent</i> (512 kbit/s, CIF-size) without watermarking (a) and with DCT domain watermarking (b).	88

4.9	The 30th frame of <i>Container Ship</i> (512 kbit/s, CIF-size) without watermarking (a) and with DCT domain watermarking (b). . .	88
4.10	Normalized correlation results of the recovered watermark from P-frames for the sequences before (left) and after transcoding attacks (right). The sequences were firstly encoded at 512 kbit/s and then were re-compressed to 256 kbit/s. The sequences were encoded in a IPPP structure. The 1st and the 50th frames are I-frames and the others are P-frames.	92
4.11	Fractional sample search positions.	93
4.12	The number of bits consumed to encode d_x , the x-component of the motion vector difference. The y-component of the MVD has the same value.	96
4.13	The number of bits that can be embedded in motion vectors when $\tau \in \{0, 0.5, 1, 2\}$ (in pixels). Both sequences are encoded at 512 kbit/s.	98
4.14	Rate-distortion (R-D) curves for the H.264/AVC coded streams without watermark and with watermark at different threshold values for the sequences <i>Foreman</i> and <i>Silent</i> . The sequences are compressed with constant QPs.	100
4.15	The 60th frame of <i>Foreman</i> (512 kbit/s, CIF-size) without watermarking (a) and with watermarking (b). (a) PSNR=39.79 dB; (b) PSNR=39.76 dB.	101
4.16	The 60th frame of <i>Silent</i> (512 kbit/s, CIF-size) without watermarking (a) and with watermarking (b). (a) PSNR=42.72 dB; (b)PSNR=42.72 dB.	101
4.17	Watermark generation and insertion into both the DCT coefficients and the motion vectors. W_R is the robust watermark and W_F is the fragile watermark.	103
4.18	Video sequences used in the experiments.	107

List of Tables

3.1	A summary of the functionality evaluation of JPEG-2000, JPEG-LS, and JPEG. The number of the star symbols indicates the comparative strength by which each functionality is supported. This table is adapted from [84].	36
3.2	The test images and the number of embedded watermark bits.	55
3.3	Picture quality comparison (PSNR in dB). In case (a), the test images are encoded with different rates; in (b), the images are watermarked at WEBR and truncated at the same rates.	56
3.4	Experiment results for “Café”	57
3.5	Experiment results for “Mandrill”	57
3.6	Experiment results for some types of <i>StirMark</i> attacks	66
4.1	The PSNR and bit-rate for the unmarked and marked video sequences.	84
4.2	The robustness against transcoding attacks. Measured using normalized correlation between the recovered and embedded message.	89
4.3	The robustness against common signal processing. Measured using normalized correlation between the recovered and embedded message under different sliding-window lengths N . All the three sequences are compressed at 512 kbit/s.	90
4.4	The MV classification in Figure 4.11 used for data embedding.	95

4.5	PSNR of the unmarked and marked test sequences. For marked sequences, 800 bits are embedded with different thresholds τ . The target and achieved bit-rates are given in kbit/s, and the PSNR value are given in dB.	99
-----	---	----

List of Acronyms

AVC	Advance Video Coding
BCH	Bose-Chaudhuri-Hocquenghem coding
BMA	Block-Matching Algorithm
BMP	Windows Bitmap
BPCS	Bit-Plane Complexity Segmentation
CABAC	Context-based Adaptive Binary Arithmetic Coding
CIF	Common Intermediate Format
CPTWG	Copy Protection Technical Working Group
DCT	Discrete Cosine Transform
DRM	Digital Rights Management
DSP	Digital Signal Processing/Processor
DVD	Digital Versatile/Video Disc
DWT	Discrete Wavelet Transform
EBCOT	Embedded Block Coding with Optimized Truncation
ECC	Error Correcting/Correction Code
FFT	Fast Fourier Transform
GOP	Group Of Picture

HDTV	High-Definition Televisions
HVS	Human Visual System
IPMP	Intellectual Property Management and Protection
IPR	Intellectual Property Right
J2K	JPEG 2000 file format
JFCD	Joint Final Committee Draft
JM	Joint Model
JPEG	Join Photographic Experts Group
JPEG-LS	Lossless JPEG standard
JVT	Joint Video Team
LABR	Lowest Authenticable Bit-Rate
LSB	Least Significant Bit
MB	Macroblock
MC	Motion Composition
ME	Motion Estimation
MMS	Multimedia Messaging Services
MPEG	Moving Picture Experts Group
MSB	Most Significant Bit
MSE	Mean-Square Error
MV	Motion Vector
MVD	Motion Vector Difference
NAL	Network Adaptation Layer
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QIM	Quantization Index Modulation
QoS	Quality of Services

QP	Quantization Parameter
ROI	Region-Of-Interest
SAD	Sum of Absolute Difference
SSD	Sum of Squared Difference
UVLC	Universal Variable-Length Coding
VCEG	Video Coding Expert Group
VCL	Video Coding Layer
VLC	Variable-Length Coding
WEBR	Watermark Embedding Bit-Rate

Chapter 1

Introduction

1.1 The Need for Watermarking

Digital multimedia has taken homes by storm in the past few years. Thanks to the proliferation of *personal computers* (PCs) with Internet connection, cheap digital rendering/recoding devices, and broadband connectivity to homes, the use, reproduction and distribution of media works are becoming much easier and faster. In addition, digital media, including photography and video, has penetrated into wireless world and has become increasingly important in terms of revenue for mobile service providers. Digital camera has become so fashionable that cellular phone manufacturers are integrating it into their new models in order to attract young people. By *multimedia messaging services* (MMS), the photos, as well as short video clips, can be distributed to anywhere with just a few clicks. With the advent of 3G mobile networks, mobile users will soon be able to enjoy television or movies wherever they are.

Although digital media offers many distinct advantages including noise-free distribution and reproduction, efficient storage, and software-based manipulation, these merits also make illegal copying much more easier and thus give

rise to concerns about protection and enforcement of *intellectual property rights* (IPR) for the digital media. For example, *digital versatile disc* (DVD) burners, which can facilitate illegal copying but are increasingly common in new PCs, featured heavily at the recent Consumer Electronics Show in Las Vegas. No surprise it was dubbed “piracy facilitation show” [1]. Given the ease with which media files can be copied, a pirated copy of DVD movie is sold below \$1 price in China and many other countries compared with \$10 or more for the genuine article. The rampant piracy implies serious revenue loss for the music, photography, and movie industries.

Digital rights management (DRM) systems have been recently introduced in attempt to address these growing concerns. They typically incorporate encryption, key management, access control, copy control, and media identification and tracing mechanisms. Figure 1.1 shows the major components. Conventional DRM systems are heavily based on encryption of the content or parts in order to prevent uncontrolled access [2]. For example, cable TV signals are commonly scrambled to prevent unauthorized viewing. But encryption alone is not sufficient. It protects contents only during the transmission from the sender to receiver. Once received and subsequently decrypted, the media data is no longer protected. But current technologies already enable consumers to capture perfect copies of a media file as it plays on a computer or a TV set [3].

Digital watermarking techniques complement encryption by hiding a secret signal, a watermark, into the original data in an imperceptible way. Like a tattoo, the watermark can not be removed without damaging original data. Therefore, the watermark persists within the work after decryption and maybe even survives some malicious attacks. Thus watermarking has been adopted in modern DRM systems and copyright protection in general for many applications:

Figure 1.1: The digital rights management (DRM) pillar model [2].

Signature The watermark identifies the creator of the media content. It can help to settle the ownership dispute.

Fingerprinting Watermarks, such as user IDs, can identify individual media content buyers. It can be used to track the source of pirated copies and enable legal action to the user who breaks the licence agreement.

Broadcast Monitoring By detecting the watermark embedded in the television and radio programs being broadcasted, an automatic monitor can verify when and where a specific program is transmitted and viewed.

Copy control The watermark bears the usage and copying rules such as “copy once” or “copy never”, which are enforced by the content owner. So the media players and recoding devices can test for the watermark and abide by the rules. It has been proposed for DVD video copy protection [63].

Authentication Watermarks can be used to check whether the data or the meaning of the content have been altered and even localize where the content was tampered.

But watermarking techniques has nowadays been used for applications beyond the limits of copyright protection proposes. For example, it can be used for **secret communication** by hiding secret data in a media work and even used as a means to **improve video quality** [35], [95]. But they are generally considered falling into a larger category of data embedding [12].

1.2 Fundamentals of Digital Watermarking

1.2.1 Origin of Watermarking

Although the digital watermarking comes into being only in the last 10-15 years, the concept of watermarking has its roots deep in the ancient craft of steganography, or “covert writing” [11]. It is the art and science of concealing secret data into a cover signal (e.g. an image) as adding noise so that third-party interceptors are even not aware of the existence of the secret message. In fact watermarking is a special case of data-hiding or steganography. One of the earliest allusion to covert writing appears in Herodotus’ *Histories*. Histiaeus, tyrant of the ancient Greek city of Miletus, shaved and tattooed secret commands on a slave’s head. After the slave’s hair grew back, no one would be aware of the message covered by the hair. He could be safely sent out as a messenger [4] [11]. More interesting historical stories on steganography can be found in [4]. Even God might take advantage of steganography. Some theologians claim that God hides His prophecies in the first five books of the *Old Testament* in the manner of a skip code — that is, every fifth letter in a sentence forms a word [5].

However, the use of the term *watermark* in context of digital data, coined by Tirkel *et al.* in 1993 [45], is in fact inspired by the watermarks in the money bills [10]. It obviously comes from the analogy that money bills use watermarks

(a) *Jiaozi* (b) Chinese *yuan*

Figure 1.2: The world first paper money *jiaozi* (a) and its modern counterpart Chinese *yuan* (b). While ancient *jiaozi* bearing visible seals and secret marks, modern Chinese *yuan* uses paper watermark to prevent forgery.

to fight against counterfeit money. Similar to our search for methods to protect valuable media data, our ancestors were faced similar problems to protect the paper money. When the world first official bill *jiaozi* debuted in China in about 1023, it bore imperial treasury's seals [6] (see Figure 1.2(a)). Notwithstanding many painful efforts had been taken like hiding secret signs in the graphs in order to identify the genuine bills, since both the seals and the marks were visible in nature and easy to forge, rampant false money compel our ancestors to abandon *jiaozi* in the end. This problem was partly solved by the paper watermark technique. It was invented by chance in Fabiano's paper factory in Italy in about 1292 [10]. They came up with the idea to lightly thin the paper at some locations. Holding the treated paper up against a strong light source, one could then see an image or logo produced by the thinner parts of

Figure 1.3: Generic digital watermarking scheme

the article. Figure 1.2(b) shows a modern example of watermark within a 100 yuan banknote. The inserted image was called *watermark* because it resembled the watery area on the paper [7].

While hailing our ancestors' wisdom, we are now facing the digitalization of our world. With the advent of digital multimedia data and transmission, including still-image, video, and audio among others, the historical concept of steganography and watermark has been expanded to the digital world to hide digital data into a media work in order to identify the rightful owner, authorized distributors and intended recipients as well as to authenticate contents by using modern *digital signal processing* (DSP) technology. Since the idea of digital watermarking arose around 1990 [43] and 1993 [45], it has gained a lot of attention. Numerous papers and practical systems have been developed. In the following, we will introduce the fundamental ideas of digital watermarking.

1.2.2 Generic Digital Watermarking Scheme

Figure 1.3 shows how a digital watermarking system functions. In general, any watermarking scheme consists of two parts: *the watermark embedder* (see Figure 1.4) and *the watermark detector/extractor* (see Figure 1.5). The watermark

Figure 1.4: Generic watermark embedder, in which the perceptual analysis is optional, denoted by gray dashed line.

embedder imperceptibly alters a media work to embed a message. This message is called *watermark*. It is generally a metadata, such as a company logo, user-ID, and content feature to mention a few, even a short video clip [9]. The multimedia work in which the watermark hidden is often referred to as *host signal* or *cover signal*. At the receiver side, we can detect the presence of the watermark only or, sometimes, extract the exact form of the watermark. In addition, during the transmission, the watermarked signal, as well as the watermark itself, might be distorted by noise or malicious attacks.

Embedding Process

The scheme in Figure 1.4 shows the basic building blocks of the watermark embedding process for image and video data. The watermarking can be performed either in the spatial domain or in the transform domain. In transform domain watermarking methods, the host image/video is first transformed to a domain that facilitates data embedding. Next, the watermark is generated from the message to be hidden. The message may need to be encrypted by the secret

Figure 1.5: Generic watermark extractor, in which the original image/video and perceptual analysis, is optional, denoted by gray dashed line.

key to increase the security or to be encoded by *error correction codes* (ECC) to increase the robustness. In addition, in order to minimize noticeable distortions to the cover signal, the embedding strength is often adapted according to some perceptual models. Therefore, optionally, the original media or its transformed counterpart might go through the perceptual analysis block. The result is often called the *perceptual-mask*. Many applications only adopt implicit perceptual models since explicit perceptual analysis is often prohibitively complex. Finally, according to the transmission/storage requirements, the watermarked image/video may need to be transformed back to the spatial domain.

Extraction Process

The scheme in Figure 1.5 illustrates the watermark retrieval process. The input is the watermarked image or video sequence. In *watermark detection*, we only

detect whether a given watermarking signal is present in an image or video. In this case, the output is some kind of confidence measure, which indicates the probability for a watermark existing in the received image. If a watermark is extracted in its exact form, we call the process *watermark extraction*.

Note that in some applications, the original image/video file, watermark and the secret key are also available, which greatly simplifies the detection/extraction process. Such a detection system is called a *non-blind* watermarking system. But in most applications, neither the original data nor the key is available. Under this scenario, the watermarking system is called *blind* watermarking system. More details on the definitions can be found in [11].

1.2.3 Features of Digital Watermarking

There are many important features for a watermarking system. But their relative importance depends on the applications. Different applications have their own specific requirements and thus their own features. In spite of this, we present some general features described in [10] [12] [14]:

Invisibility While there do exist visible digital *watermarking*, most of the literature has focused on *invisible digital watermarking*, which have wider applications [11] The embedded watermark must be perceptually transparent and must not introduce any noticeable artifacts.

Security Unauthorized access to the watermark must be impossible even if the watermarking algorithm is known, as long as the hacker does not obtain the right key. Usually, this is achieved by using the cryptographic keys.

Robustness The robustness refers to how reliably the embedded information can be detected or extracted after the watermarked work undergoes vari-

ous manipulations or tampering. According to this requirement, the watermarks can be divided into:

- *Fragile watermark*: it is sensitive to any manipulation, which is quite useful for authentication.
- *Semi-fragile watermark*: some applications require the watermark to be insensitive to some common signal processings, such as filtering and compression, but capable to detect malicious attacks. This type of watermark is called *semi-fragile watermark*.
- *Robust watermark*: it is expected to survive any attacks.

Capacity It refers to the number of bits that can be embedded into the cover signals. Generally, the payload should be as high as possible.

Complexity Generally speaking, the algorithm complexity is not a concern for imagery applications. But it is particularly true for *real-time* video watermarking applications. In these cases, the complexity of the watermarking algorithm is expected to be as low as possible [15]. In addition, there generally exists an asymmetry between the requirement for speed of insertion and that of detection. For example, in case of DVD copy control, while we can afford complex insertion methods by high-cost professional equipments during DVD manufacture, the detection process must be simple enough to be able to conduct in real-time on low-cost customer players.

Standard compatibility Most media contents are stored and transmitted in some compression standards in order to facilitate exchange information between different entities. As a result, the watermarking method is often required to align with media compression standards.

We should note that all of the above features are related to each other. For example, a watermark can increase its robustness by either increasing the energy of the watermark signal or by making a lot of changes in different places. However, both the methods may degrade the quality of the cover signal and make the watermark noticeable. In addition, the appropriate embedding places in the cover signal are generally very limited. Large amount of modification means decreasing the capacity. Therefore, a practical watermarking system must strike a delicate balance between these related requirements and features.

1.3 Motivation and Objective

The proliferation of digital media products and services owe much to the advance of image/video coding techniques and the emergence of several successful international standards, which not only offer efficient compression methods but also offer the essential inter-operability between devices and systems designed by different manufacturers around the world, hence facilitating the growth of the photography and video entertainment markets.

Currently, two of the most preeminent standardization bodies are ISO/IEC¹ and ITU-T². For still-images, they have been working together and established the *Joint Picture Expert Group* (JPEG). For videos, ITU-T has Video Coding Expert Group (VCEG) and ISO/IEC has the *Moving Picture Experts Group* (MPEG). Since the mid 1980s, they have developed a series of image and video standards addressing a wide range of applications with different requirements (see Figure 1.6). For example, current main-stream digital cameras use

¹The International Standardization Organization and the International Electrotechnical Commission.

²The International Telecommunications Union, Telecommunication Standardization Sector.

Figure 1.6: Progression of image and video coding standardization

JPEG still-image coding standard to save storage space. MPEG-2 is the coding standard for Digital TV and DVD-video. New models of Pocket PCs and cameraphones³ take advantage of MPEG-4 to provide mobile video services. Given such popularity, the interest in developing watermarking techniques aligned with these standards is high. For example, the DVD standard will contain a copy protection system employing watermarking.

In December 2002, JPEG released its second-generation still-image compression standard — JPEG-2000. In contrast to the conventional JPEG standard, which is based on *discrete cosine transform* (DCT), JPEG-2000 uses *discrete wavelet transform* (DWT) for efficient compression. Thus, it offers several new features such as good quality at high compression ratios, progressiveness up to lossless coding, and region of interest coding. It is highly promising that JPEG-2000 will become a popular image coding standard in the coming future. Naturally, researchers have been motivated to develop DRM systems to securely exchange JPEG-2000 images over the networks. A new part of the

³mobile phones with a built-in digital camera

standard, Secure JPEG-2000 – Part 8 (JPSEC), has been standardized recently [70]. Therefore, our research endeavor is first focusing on the development of novel digital watermarking method aligned with JPEG-2000 standard.

On the other hand, H.264/AVC⁴ is the newest standard and the state-of-the-art coding methodology for digital video. It has been shown that some core components of H.264/AVC are quite different from its predecessors, such as tree-structured motion segmentation from 16 x 16 down to 4 x 4 block size rather than exploiting a fixed 16 x 16 block size used in previous standards, and rate-constrained motion estimation and mode decision. While these new features enable H.264/AVC to achieve superior performance in terms of coding efficiency, they impose great challenges for traditional watermarking methods. It is worthwhile to investigate new approaches for H.264/AVC watermarking. Our goal is to improve the visual quality, flexibility and robustness of the watermarking system in dealing with the low bit-rate real-time videos.

1.4 Major Contributions of the Thesis

The thesis has two major contributions as described in the following:

o A novel proposal for JPEG-2000 watermarking

We present a novel blind watermarking method for JPEG-2000 image compression standard. Based on our observation of the EBCOT algorithm, which is the foundation of JPEG-2000 encoder, as well as the wavelet coefficient representation, we introduce a new parameter WEBR (watermarking embedding bit-rate) to quantitatively control the watermarking strength so that the watermarking is distortion bounded and the result-

⁴ITU-T Recommendation H.264 and also known as ISO/IEC International Standard MPEG-4 part 10 Advanced Video Coding (AVC)

ing distortion can be estimated even before watermarking. We insert the watermark bits by taking advantage of BPCS (Bit-Plane Complexity Segmentation) steganography method in order to strike a good balance between robustness and invisibility. The combination of WEBR and BPCS method not only provides robust approach for JPEG-2000 watermarking but also offers the flexibility to embed only a code-block or a subband.

- **A set of watermarking methods for H.264/AVC standard**

Two syntax elements of H.264/AVC — *DCT coefficients* and *motion vector differences* (MVD) — are exploited respectively for *robust* watermarking and *fragile* watermarking. Combining the advantages of the two techniques, a *semi-fragile* watermarking scheme is also proposed for video authorization. All these techniques are aligned with the H.264/AVC coding standard, some features of which like Lagrangian optimization are incorporated in our fundamental design. The experimental results show that the proposed watermarking approaches are of low computational complexity and capable of embedding and extracting a watermark on-the-fly in the H.264/AVC coding process. The experiments also demonstrate that the methods achieve desirable robustness, visual quality, and flexibility when dealing with low bit-rate videos (≤ 1 Mbit/s).

1.5 Organization of the Thesis

We outline the organization of the thesis as follows:

In this chapter, a brief introduction to the thesis has been provided including the background of our research work, the motivation, objectives, and the main contributions.

Chapter 2 gives a review of various watermarking technologies for images and videos, respectively. Video watermarking fundamentals have also been introduced in this chapter.

Chapter 3 describes our new watermarking method for JPEG-2000 compressed images.

Chapter 4 is devoted to the watermarking of H.264/AVC compressed videos. Three watermarking approaches are described and experiments are conducted to demonstrate their performances.

Finally, Chapter 5 summarizes our contributions and recommendations for future research endeavors.

Chapter 2

Review of Digital Watermarking Techniques

2.1 Overview

Since the publication of the Tanaka's, and others, seminal work [43] in 1990, numerous digital watermarking schemes have been developed for various media types, for example, documents, softwares, e-books, images, videos, audio and games. In this thesis, we will focus on watermarking *images* and *videos*. Digital watermarking techniques take advantage of limitation of the *human visual system* (HVS). They hide a watermark in cover signals in such a way that it can not be noticed by human eyes and can only be detected and extracted by DSP techniques. While digital watermarking for images have been extensively studied since the infancy of the watermarking, watermarking video is a fairly new area of research [15]. Two watermarking approaches are commonly used. One is to hide the information on host data without taking into account compressions, that is, a watermark is embedded by directly modifying the spatial-domain pixel values. The other is to explicitly exploit a specific image or video compression

standard. Indeed, most of the time, images or videos are stored in a compressed format, like JPEG, H.26x, MPEG-4 etc., in order to save storage space and transmission bandwidth.

In this chapter, we will begin with watermarking techniques for still images. Then, the specialities of watermarking for video contents will be discussed in the later part of this chapter. For image and video, we further classify the watermarking methods in the context of un-compression and compression.

2.2 Image Watermarking Techniques

2.2.1 Techniques in Uncompressed-domain

It is straightforward that the *least significant bits* (LSB) of an image does not contain visually significant information. Thus, the LSBs can be modified to conceal a large number of watermark bits. This methodology was adopted by Tirkel *et al.* in [45], in which the watermark, in form of an *m*-sequence derived *pseudo-noise* (PN) code, either is added to or replaces the LSBs. The latter uses autocorrelation for watermark detection. Then, an improved version was published in 1994, titled *A Digital Watermark* [46], from which the term “digital watermarking” came about. A more recent and sophisticated example of making use of LSB modification can be found in [47]. It embeds an image to itself as a means for anti-tampering. Generally, this type of technique is fragile to any manipulation.

Tanaka *et al.* [43] proposed a method to embed a signal in an image presented by dithering (for monotone printing). And later, Matsui *et al.* [49] proposed three methods based on different coding schemes: predictive coding scheme, dithering, and run-length codes (for fax).

Bender *et al.* [48] proposed two approaches, called *Patchwork* and *Texture Block Coding* for low bit-rate data hiding. In particular, “Patchwork” method chooses pairs of pixels (a_i, b_i) pseudo-randomly by using a key. The ‘1’ is embedded by raising a_i ’s by δ and decreasing b_i ’s by the same amount of δ at the same time. Assuming the expected value of the difference between the non-watermarked pairs is zero, the watermarking process shifts that of watermarked pairs by 26. This method is quite robust to tampering but the watermark capacity is extremely low.

Kutter *et al.* [50] presented a spread-spectrum watermarking method based on amplitude modulation. A single bit is multiply embedded by modifying pixel values in the blue component of an image in RGB format. According to the watermark bit w , the blue value either adds or subtracts a value which is proportional to the luminance:

$$B_{m,n} \leftarrow B_{m,n} + \alpha(-1)^w L_{m,n}, \quad (2.1)$$

where $B_{m,n}$ denotes the blue value of a pixel located at (m,n) , which is pseudo-randomly selected, $L_{m,n}$ denotes the luminance value at the same location, and α denotes the embedding strength. Moreover, the original (un-watermarked) image is not necessary to extract the watermark, which implies it is a *blind* watermarking method. The original value can be estimated by linear combination of neighboring pixels. This method has shown to be resistant to some attacks, such as filtering, and geometrical attacks.

The characters of the *human visual system* (HVS) are also exploited in designing the watermarking schemes. Macq *et al.* [51] introduced a watermarking method taking advantage of the *masking* effect. The watermark is firstly low-pass filtered and modulated. The embedding location and the frequencies of

modulation is determined by a secret key. Then, the watermark is masked and added to the host image.

Moreover, *reversible* (or *lossless*) watermarking techniques have been proposed recently [52][53][8]. It is desirable for some applications such as medical diagnosis and law enforcement, which require the marked image be able to be reversed to the original cover image after watermark retrieval.

2.2.2 Techniques in Compressed-domain

As mentioned before, most images are now circulated in compressed formats. Nowadays the most popular compression format is JPEG, which is widely used in Internet, mobile phones, and digital cameras. Therefore, it is no surprise JPEG-based watermark methods are numerous. On the other hand, with the standardization of JPEG-2000, more and more JPEG-2000-based compressed methods have been proposed. The most significant benefit of watermarking the compressed images directly is that of compatibility. When a watermarking method takes advantage of the DCT/DWT transform and the coding structure of a specific compression format, the robustness and transparency generally can be more easily achieved. That is because watermarking DCT or DWT domain spreads the energy of embedded signal all over the spatial domain and furthermore the features of HVS can be incorporated into the transformed coefficients more effectively [59].

One of the prominent jobs was done by Cox *et al.* in [54], which incorporates the idea of *spread spectrum communications* [60] into watermarking. It is the basis for many current works [58][59]. In spread spectrum communications, a narrow-band signal (the watermark messages in our case) is modulated by a wide-band signal, such as Gaussian noise, in order to spread the narrow-band

signal over a wide band frequencies [60]. In [54], the watermark is constructed as a set of independent and identical distributed (i.i.d.) values w_i with Gaussian distribution. Then, a full frame DCT is applied to the whole image and the perceptually most significant AC components of the host image are modified according to the watermark. The embedding formula can be one of the following:

$$v'_i = v_i(1 + \alpha w_i), \quad (2.2)$$

$$v'_i = v_i + \alpha w_i, \quad (2.3)$$

$$v'_i = v_i e^{\alpha w_i}, \quad (2.4)$$

where v_i and v'_i are the DCT coefficients before and after modification, respectively, and α determines the watermarking strength, which can be adaptive to different spectral component.

In watermark detection process, the embedded watermark can be easily extracted from the received, possibly distorted, image by referring the original image. Then, the extracted watermark w^* is compared with the original embedded watermark w . The normalized correlation is used to measure the similarity

$$\text{sim}(w, w^*) = \frac{w \cdot w^*}{\sqrt{w^* \cdot w^*}}. \quad (2.5)$$

Finally, the presence of the watermark w can be determined if $\text{sim}(w, w^*) \geq T$, where T is a threshold. Experiments show that this non-blind watermarking scheme can resist JPEG compression at a 5% quality factor and several other kinds of attacks. While this work is done in the DCT domain, it has been extended to other domains such as DWT and FFT.

In order to improve Cox's method, many algorithms have been proposed. Lu *et al.* [58] uses *cocktail* watermark to improve the robustness and used HVS

Figure 2.1: An improved Koch's watermarking algorithm [56] embeds a bit by adapting relationship between three coefficients. The 'X' for watermark bit w denotes a invalid pattern.

to maintain high fidelity of the watermarked image. Huang *et al.* [59] embed a watermark by modifying the DC components.

In the following, we focus on standard-compliant watermarking methods.

JPEG-based Watermarking

Koch and Zhao [55] proposed one of the earliest JPEG-based watermarking techniques. The image is firstly divided into 8×8 blocks and then a sequence of blocks are pseudo-randomly selected. DCT and quantization are applied to the selected blocks. In order to embed a bit w , a pair of quantized coefficients $\{X_q(m_1, n_1), X_q(m_2, n_2)\}$ in the middle frequency ranges of a block are slightly modified so that:

$$\Delta_q = X_q(m_1, n_1) - X_q(m_2, n_2) = \begin{cases} \geq \delta & \text{if } w = 1 \\ < -\delta & \text{if } w = 0 \end{cases} \quad (2.6)$$

where δ controls the embedding strength. The authors chose middle frequency components for watermarking based on a belief that the alteration in low frequencies will be perceptually significant whereas the high frequencies are vulnerable to attacks. This scheme was extended in [56], in which a watermark bit is embedded by enforcing a relationship between three DCT coefficients. Figure 2.1 shows an example. This method shows good robustness to JPEG compression for a quality factor as low as 50%. Hsu and Wu [57] describe a method in which the watermark in form of a binary sequence is inserted into the mid-band frequencies of the 8×8 DCT coefficients.

JPEG-2000-based Watermarking

Some of the early works on wavelet-based watermarking methods include Xia *et al.* [76] and Wang *et al.* [78][77]. With the emergence of the new JPEG-2000 standard, more and more researchers have considered DWT-based watermarking methods in the context of JPEG-2000 compression standard.

Su *et al.* [80] proposed an integrated method for JPEG-2000 images compression and watermarking. Basically, it adopts the same spread-spectrum watermarking method as previous mentioned Cox's method [54]. A pseudo-noise number is generated as a watermark and added onto the significant wavelet coefficients as Equation 2.3. The significant wavelet coefficients are those with a magnitude greater than a threshold. To detect the watermark, Equation 2.5 is used to identify the watermark. Clearly, the original watermark is required in the watermark detection process. Grosbois and Ebrahimi [81] improved Su's work by adjusting the embedding strength α according to some HVS properties:

$$\alpha = M_{dyn} * \rho_{act} * \rho_{mask} \quad (2.7)$$

where M_{dyn} is the maximum magnitude of the embedded noise; ρ_{act} represents the activity of a block of wavelet coefficients; ρ_{mask} is a visual intra-band masking strength factor.

Noda *et al.* [88] applied the *bit-plane' complexity segmentation* (BPCS) steganography [87] to JPEG-2000 encoded images. This work formed the basis for our work and will be detailed in Chapter 3.

Chen *et al.* [82] proposed another watermarking method based on bit-plane modification, in which the watermark bits are firstly scattered by *Toms Automorphisms*, and then bit-planes of wavelet coefficients are modified according to a distortion reduction operation.

Su and Kuo [79] analyzed the JPEG-2000 codec and proposed a high capacity data hiding method. A special entropy coding mode — *lazy mode* — are employed to inert the watermark during entropy coding of JPEG-2000. This method, like other steganography methods, is not robust against distortions.

2.3 Video Watermarking Techniques

A raw (uncompressed) video sequence is a set of successive frames², where each frame is a 2-D image. Therefore, many still image watermarking techniques can be extended to video data. But the availability of the third dimension — temporal dimension — of video sequences provides much larger space for watermarking, which not only provide new opportunities but also impose new challenges for watermarking [15]. For example, more attacks are valid for video, such as frame dropping, averaging, and jittering [15]. Furthermore, low complexity is also an important constraint for video watermarking algorithm design.

¹Bit-plane refers to a 2-D array of bits of the same magnitude in all coefficients or samples.

²In the context of digital video, we interchangeably use the terms of image, frame and picture.

Watermark embedding and retrieval must be performed in *real time* to avoid slowing down video viewing, recording, archiving, and so on. Thus, some researchers believe the explicit models of HVS are too complex to be used in video watermarking [10]. In addition, most video products like DVD are published in a compressed version. Distributors are likely to recompress their video with a different compression ratio. And customers may like to convert their video from one format to another. These processes are in fact deemed as a kind of attack.

Similar with the image watermarking techniques, video watermarking methods can also be classified into the uncompressed and compressed domain and will be discussed respectively in the following sections. We focus on compressed video watermarking because it differs much more from the image-like methods. Furthermore, our new video watermarking method proposed in Chapter 4 is also for compressed video stream.

2.3.1 Techniques in Uncompressed-domain

JAWS (*JustAnother Watermarking Method*), proposed by Kalker *et al.* [62] [63], is one of the leading real-time watermarking methods for DVD video copy protection. The watermark consists of a repeated $M \times M$ pattern generated by a secret key. It is embedded before compression with a scaling factor based on the local activity of a frame. Detection is performed after decompression by a correlation detector.

The DCT domain still-image watermarking method has been adopted for video data. For example, Busch *et al.* [61] adopted Koch and Zhao's algorithm [55] to mark the video frames by altering the DCT coefficients. The video is marked directly after recording and before entering the TV production, and finally broadcasted as an MPEG-2 bitstream.

Cox *et al.* [54] applies the block-based spread spectrum watermarking method to video data, as well as still images. Hartung *et al.* [14] proposed an additive spread-spectrum watermarking scheme, in which the video is treated as a one-dimensional signal acquired by line-scanning.

2.3.2 Techniques in Compressed-domain

Fundamentals of Video Coding and Watermarking

Figure 2.2: A typical hybrid video coding scheme

Currently, most contemporary video compression standards employ DCT and *block-based hybrid coding* methodology [23], which compress video frames in either *intra-coding* or *inter-coding* in order to exploit spatial and temporal redundancy respectively. The *intra-coded frame* (or *I-frame*) is encoded by block-based DCT without referring to other pictures, and with basically the

Figure 2.3: Basic steps of the compressed-domain watermarking method

same technique of JPEG standard; on the other hand, *P-frame* and *B-frame*, collectively known as *Inter-coded frame*, are predicted from other frames by motion estimation and compensation process, and thus only the prediction errors (i.e. the changes from current picture to its reference frames) and motion vectors need to be encoded. Figure 2.2 shows the block diagram of a typical block-based hybrid video encoder.

Most of the time, digital video products, such as DVD and video-on-demand, will already be in a compressed format, when the watermark needs to be inserted. It will be much more convenient and desirable to watermark compressed video stream directly, without going through a full decoding, watermarking, and re-encoding process. Therefore, a compressed-domain watermarking method often exploits one of the previous mentioned syntax elements in code-stream, such as *DCT coefficients* and *motion vectors*, or their *run-length codes*, in order to hide the information [15]. Figure 2.3 depicts the basic steps of the compressed-domain watermarking. The original compressed video bitstream partially decoded to expose the syntax elements of the compressed video for watermarking.

Next, these elements are modified to insert a watermark. Finally, the modified video data are reassembled to produce a syntactically valid bitstream, which can be decoded by a standard decoder. In the following sections, we will introduce the watermarking techniques according to the to-be-modified syntax elements.

Watermarking DCT Coefficients

Hartung and Girod [14] extended their uncompressed-domain spread-spectrum watermarking method to compressed-domain. An MPEG-2 video is first partially decoded to obtain the DCT coefficients. These coefficients are modified by adding the watermark which is also DCT transformed. The watermark detection is performed by correlation after the encoded video is fully decoded. Alattar *et al.* [41][64] further extended this method for MPEG-4 video watermarking.

Langelaar and Lagendijk [65] [66] proposed the *differential energy watermark* (DEW) algorithm for real-time watermarking JPEG compressed images or I-frames of an MPEG compressed video. One bit is embedded by selectively discarding some high frequency DCT coefficients, and thus introducing an energy difference between two regions in a group of 8×8 blocks.

Watermarking Motion Vectors

The *Motion vector* (MV) is one of the most important syntax element of MPEG compressed video and is found in motion estimation process (see Figure 2.2). Figure 2.4 illustrates the basic idea of the most popular motion estimation method: the *block-matching motion estimation* (BMME), which computes MVs on a block-by-block basis. The to-be-estimated block in the current frame is compared with those candidate blocks in the reference frame within a searching range, or *searching window*, in order to find the best matching block for a given matching criterion, most commonly minimum *sum of absolute difference* (SAD).

Figure 2.4: Typical process of motion prediction. The arrowheaded line is the motion vector. And the gray area is the search window.

As a result, the MV is the relative displacement from the current block to the best matching block.

Exploiting MVs to carry the watermark is firstly proposed by Jordan *et al.* [32] for MPEG-4 video blind watermarking. One watermark bit $w \in \{0,1\}$ is inserted into a MV component v according to the following rule:

$$\begin{aligned}
 &\text{if } (v \cdot q + T) \bmod 2 \neq w \\
 &\quad v' = v + \delta \\
 &\text{else} \\
 &\quad v' = v
 \end{aligned}
 \tag{2.8}$$

where $T = 2x$ (x is the search window used for motion estimation), v' is the modified MV component, and $\delta = (2n + 1)/q$, n is integer, and q is used to specify the amplitude of MV modification (for example, $q = 2$ for half-pel signing). The embedded bit w can be extracted from the MPEG-4 bit-stream during decoding

when the modified MV component v' is reconstructed:

$$w = (v' \cdot q + T) \bmod 2. \quad (2.9)$$

The complexity of this method is neglectful. But authors didn't show the experiment results for quality of the watermarked frames.

Zhang *et al.* [34] improve the above insertion rule by introducing two principles: the watermark bit should be embedded into the MVs with large magnitude $|\vec{v}| \geq E$ (E is a threshold), and into the MV component with greater magnitude. The experiments show that the watermarking degrades the quality of the video by very little.

Song and Liu [33][35] propose to embed watermark bits by modifying the half-pixel motion estimation process. According to the watermark bit, the MV component is matched to either the integer-pixel position or half-pixel position.

However, above methods suffer from serious drawbacks. A simple filter can destroy the parity of the MV components. Therefore, they basically are fragile against any manipulation.

Bodo *et al.* [36] propose a robust MV watermarking method based on a hierarchical motion analysis.

Watermarking Other Syntactic Elements

Langelaar *et al.* [65] add a label directly into the MPEG-2 bitstream by changing *variable length codes*(VLCs).

Linnartz *et al.* [67] propose to modify the MPEG encoding procedure to choose *the picture type* of video-frames (I-, P-, and B-frames) according to a message one would like to embed.

2.4 The Future of Watermarking

Many previous mentioned watermarking methods treat the cover signal as noise and do not exploit knowledge of the cover signal to reduce the interference between the watermark and the cover signal. These are known as *blind* watermarking methods. The most prominent representative is the *Spread Spectrum* watermarking method. However, in recent years, more and more researchers agree that the *Informed Embedding Watermarking* outperforms the blind watermarking method [109] in many ways, especially in terms of capacity. One of the representatives is the *Quantization Index Modulation* technique, proposed by Chen and Wornell in [108]. This class of techniques embeds the watermark in a cover signal through quantization; a different quantization vector is used to embed a different watermark value. Please refer to [107], [110], and [111] for more details. However, we have not seen any application of QIM on standard-compliant watermarking problems.

Chapter 3

Watermarking JPEG-2000

Images¹

3.1 Introduction

JPEG-2000 is the latest image compression standard from the *Joint Photographic Experts Group* (JPEG). Based on the *discrete wavelet transform* (DWT), it creates a unified platform, which provides not only higher compression efficiency but also a pile of new features essential for the Internet era [84], such as seamless scalability, progressiveness up to lossless compression, and region-of-interest coding. As JPEG-2000 standard is beginning to gain widespread favor in imagery applications, industries and researchers have been motivated to find ways to allow applications to securely generate, consume, and exchange JPEG-2000 bitstreams [70]. Several attempts to introduce image steganography and watermarking techniques into JPEG-2000 system have been reported recently in [80][88][79]. Readers are referred to the previous chapter for more reviews. Especially, the *blind* watermarking methods catches researchers' attention because

¹This work has been completed in the Institute for Infocomm Research (I²R), Singapore.

they do not require the original image during watermark detection and recovery. It is worthwhile to investigate efficient blind watermarking methods which are fully aligned with this new standard. Therefore, this chapter is dedicated to the watermarking of JPEG-2000 images.

As we mentioned before, the art of the design of a watermarking scheme is to strike a delicate balance among security, visual quality, flexibility, etc. We construct the novel blind watermarking scheme based on the *bit-plane complexity segmentation* (BPCS) steganography, which has proven to achieve larger capacity than other traditional steganography methods without introducing more distortion to different types of images, including gray-level images [87], *embedded zero-tree wavelet* (EZW) encoded images [90], and JPEG-2000 encoded images [88][89]. However, the BPCS method is fragile in nature. It thus cannot survive even the most common manipulations such as multiple cycles of recompression and format conversion from one to another among a variety of popular image formats, like JPEG-2000, JPEG, and BMP². In order to improve its robustness, we introduce a new parameter called *watermark embedding bit-rate* (WEBR)³ to quantitatively control the watermarking strength. It originates from the *lowest authenticable bit-rate* (LABR) concept first proposed by Sun *et al.* in [86] for JPEG-2000 image authentication. Similar with [86], our scheme uses a user-defined rate, i.e. WEBR in our case, to determine watermarking bit-planes of the wavelet coefficients. But we improve in only embedding the bit-planes below the marginal bit-planes found by WEBR, which implies the distortion resulted from watermarking is bounded. Since this process can directly take advantage of the rate control mechanism of JPEG-2000, our method is well aligned with the scalability feature of the JPEG-2000. After embedding, the watermarked

²The file format for windows bitmap.

³In this chapter, the bit-rate of an image is measured as *bit per pixel* (bpp).

JPEG-2000 code-stream can be truncated according to some progressive orders without tampering the embedded watermarks as long as the compression rate is larger than the WEBR to some extent. In addition, the proposed watermarking method features a simple but efficient mechanism to select places appropriate for embedding. In this mechanism, we jointly consider the visual factors of the spatial domain and the transformed domain in terms of effects of watermarking to perceptual qualities.

This chapter is organized as follows. In Section 3.2, we provide a brief review of the core elements of JPEG-2000. The proposed watermarking method is detailed in Section 3.3. Experimental results will be shown in Section 3.4 to demonstrate the feasibility of the proposed method. Finally, Section 3.5 will give conclusions.

3.2 Review of JPEG-2000

JPEG-2000 (ISO/IEC 15444 | ITU-T Rec. T.800) [69], as noted previously, is the newest image compression standard designed by the *Joint Photographic Experts Group* (JPEG), which is officially chartered as ISO/IEC JTC1/SC29/WG1⁴ and also joints the efforts from ITU-T.

The first generation standard of this committee — JPEG standard [68] — has enjoyed phenomenal success since its debut in 1993, which is partially attributed to its simple but efficient coding system [75]. In the lossy compression mode, the image is divided into blocks of 8×8 pixels, and each of them undergoes the *discrete cosine transform* (DCT), followed by quantization and Huffman coding. In the lossless compression mode, known as JPEG-LS (lossless

⁴International Standardization Organization and the International Electrotechnical Commission, Joint Technical Committee number 1, Study Committee 29, Working Group 1.

Functionalities	JPEG-2000	JPEG-LS	JPEG
Lossless compression performance	***	****	*
Lossy compression performance	*****	*	***
Progressive bitstreams	****		*
Region of Interest (ROI) coding	***		
Random access	**		
Low complexity	**	*****	*****
Error resilience	***	*	*
Non-iterative rate control	***		

Table 3.1: A summary of the functionality evaluation of JPEG-2000, JPEG-LS, and JPEG. The number of the star symbols indicates the comparative strength by which each functionality is supported. This table is adapted from [84].

JPEG), the DCT is no longer necessary. The compression is wholly based on a predictive coding approach followed by Huffman coding or arithmetic coding.

However, with the emergence of new imagery applications such as digital libraries, medical imagery, and interactive imaging over Internet and mobile, it is becoming increasingly apparent that the traditional JPEG standard is limited in fulfilling the requirements of these emerging applications. Instead of the DCT, the new JPEG-2000 standard takes advantage of the *discrete wavelet transform* (DWT) and the *embedded block coding with optimized truncation* (EBCOT) to achieve not only higher compression efficiency but also rich scalability, which are essential for the Internet and mobile environments [84].

Table 3.1 lists some of the major merits and features of JPEG-2000. Since some of these features are not really new, we also give a comparative evaluation for JPEG-2000 and its precedent JPEG-LS (lossless JPGE) and JPEG (lossy JPEG). Obviously, JPEG-2000 is superior in many ways to JPEG-LS and JPEG at the expense of the computational complexity. Readers are referred to [72], [83], and [84] for more reviews.

3.2.1 JPEG-2000 Codec

Figure 3.1: The block diagram of JPEG-2000 encoding and decoding processes, in which the ROI scaling and de-scaling are optional.

The block diagram of the JPEG-2000 codec (COder/DECoder pair) is illustrated in Figure 3.1. A typical JPEG-2000 encoder comprises preprocessing, DWT, scalar quantization, and EBCOT. EBCOT is of a two-tiered structure, where tier-1 is a context-based arithmetic coder and tier-2 is for post-compression rate allocation. The decoder is basically the reverse procedure of the encoder. We briefly introduce each step of the encoding process as follows:

A. Preprocessing: The input image is partitioned into components and non-overlapping rectangular area, called *tiles*. In order to save memory during processing, tile-component is the basic unit for the whole following encoding and decoding processes. But it is common to include the whole image in a tile. The components may go through a component transform to improve compression.

B. Discrete wavelet transform: The DWT can be performed with either the reversible integer Le Gall 5/3 filter, which provides for lossless coding, or

with the Daubechies 9/7 filter, which provides for higher compression but does it in a lossy way. The 2-D DWT filter-bank is obtained by applying the 1-D DWT filter-bank horizontally and then vertically in sequence, which yields four subbands: LL, HL, LH, and HH ("H" denotes high-pass filtering; "L" denotes low-pass filtering). Thus, the tile-component data are decomposed into different decomposition levels and wavelet subbands in a *dyadic* fashion as shown in Figure 3.2, where the number indicates the decomposition level. Figure 3.3 shows the steps of three-level wavelet decomposition for the image "Mandrill".

C. Quantization: Each subband is further divided into relative smaller blocks, known as *codeblocks*, typically with a size of 64 x 64 samples except at image boundaries. They are quantized by an embedded dead-zone scalar approach and normalized for each subband. The results are represented in sign-magnitude: the sign is located in the *most significant bit* (MSB) and directly followed by the absolute value of the coefficient's magnitude.

D. Region of interest (optional): Some applications desire certain region of the image, called *region of interest* (ROI), to be of higher quality relative to the other part of the image after encoding. The ROI functionalities can be realized by the *Maxshift* [69][83] method, which scales up the wavelet coefficients in the relevant regions.

E. Embedded block coding with optimized truncation (EBCOT): Just as the DWT is fundamental to resolution scalability, EBCOT, proposed by Taubman [71], is the key to distortion scalability. EBCOT generates a "finely embedded bit-stream" [71], so that each of those subsets can efficiently represent the image at a reduced rate (bits per pixel) or increased distortion. In other words, publishers just need to *compress the image only once*, but consumers can *decompress the image in many ways* [72]. We will therefore take a closer look at EBCOT in the following section.

3LL	3HL	2HL	1HL
3LH	3HH		
2LH		2HH	
1LH			1HH

Figure 3.2: The dyadic decomposition of three levels and the corresponding label for every subband. For example, kHL indicates that during the k-th level of decomposition, the high-pass (H) filter is applied horizontally first, followed by a vertical low-pass (L) filter.

Figure 3.3: The steps of a three-level wavelet decomposition for the image “Mandrill” using lossy 9/7 filter-bank. Recursively in each step, the kLL is decomposed to yield $(k + 1)$ LL, $(k + 1)$ HL, $(k + 1)$ LH, and $(k + 1)$ HH.

3.2.2 The EBCOT Algorithm

Figure 3.4: Illustration of the coding process of the EBCOT Tier-I encoder.

Quantized wavelet coefficients, grouped as codeblocks $\{C_i\}$, where i is the serial number of a codeblock, are encoded by the two-tiered EBCOT algorithm. Figure 3.4 illustrates the coding procedure of the Tier-1 coding, in which, the bit-planes⁵ of the codeblock C_i are encoded one-by-one. The bit-plane with high magnitude are sent first, followed by the next and so on until all bit-planes in the integer portion are coded. Each bit-plane is further fractionalized and processed through three *passes*: significance pass, magnitude refinement pass, and cleanup pass. Eventually, the arithmetic coder generates a finely embedded bitstream [71]. By *finely embedded*, we mean the bitstream can be truncated at a number of points and each of the resulting subsets still offers rate-distortion (R-D) optimized representation of the image [72]. Thus, the output of the Tier-1 coder for the codeblock C_i is a collection of compressed bitstreams and their

⁵A bit-plane is a 2-D array of bits of the same magnitude in all coefficients or samples.

Figure 3.5: EBCOT quality layer formation. Each column represents the bitstreams generated for all the passes of a codeblock.

corresponding $(R_i^{n_i}, D_i^{n_i})$ information at truncation points, where $R_i^{n_i}$ denotes the number of bits of the truncated bitstream between the truncation point n_i and $(n-1)_i$, and $D_i^{n_i}$ denotes the resulted distortion. And $D_i^{0_i}$ denotes the distortion when the whole codeblock is discarded. Note that the truncation points $\{n_i\}$ are located at the boundaries of the passes.

The resulting bitstreams of all codeblocks are then collected and organized in the Tier-2 coder to form the so called *quality layers* as illustrated in Figure 3.5, where "layer 1" offers the basic representation and each of the following layers (layer 2, 3, and 4) represents an increment in image quality [72]. The quality layer may contain some (possible empty) incremental contributions from each of the codeblocks. The output is organized as *packets*. The final bitstream is formed according to a pre-specified compression bit-rate and progression order.

3.3 The Proposed Method

3.3.1 System Overview

In the proposed scheme, the watermark is embedded into the wavelet coefficients during JPEG-2000 compression process while the watermark extraction is performed during the decompression process. This simplifies the implementation of the proposed method. As shown in Figure 3.6, we simply add the embedding module between the normal quantization and ROI scaling module. Thus, we embed the watermark right after the wavelet coefficient quantization but before the ROI scaling and the watermark retrieval is done before the dequantization as shown in Figure 3.7.

As shown in Figure 3.6, there are three inputs for the proposed embedding procedure: the cover image, the watermark message and the *Watermark embedding bit rate* (WEBR). The WEBR is designed to quantitatively control the embedding strength. It means that we can use the WEBR to find the highest bit-planes for watermarking while keeping the resulted distortion bounded. In order to achieve better perceptual quality and large watermarking capacity, we adopt the *bit-plane complexity segmentation* (BPCS) method [88] to inset the watermark. In addition, we select the candidate embedding place based on the information collected from the original image and the wavelet coefficients.

In order to let readers have an overall appreciation, we outline the embedding steps as follows:

- (1) The cover image goes through the normal JPEG-2000 encoding process but this process is halted right after the quantization. At this point, the wavelet subbands have been divided into codeblocks $\{\mathcal{C}_i\}$, where i is the serial number of a codeblock. We store all quantized codeblocks into memory.

Figure 3.6: Watermark embedding procedure

Figure 3.7: Watermark retrieval procedure

- (2) The stored codeblocks are then encoded by the EBOCT algorithm with a user-defined WEBR, which acts as the target compression rate. Thus, EBCOT finds for each codeblock $\mathcal{C}_i$ the truncation point n_i , from which we deduce the bit-plane p_i for watermarking.
- (3) On the other hand, the stored codeblocks are further divided into small *patches* of $N \times N$ samples. A *visual factor* $F_{m,i}$ is computed for each patch. The *embeddable patches* (i.e. the patches appropriate for embedding) are selected according to this factor. A decision is made on the decomposition level for watermarking.
- (4) The watermark is generated by segmenting the secret message into a number of 8×8 binary blocks, called *secret patches*. *Error correction codes* (ECC) may be applied in order to increase robustness so that the embedded information can be correctly extracted even under incidental distortions. Our method encodes the secret message by BCH and then formulates block-based watermarks.
- (5) The secret patches are embedded with the BPCS method, which replaces the bit-plane at p_i of the embeddable patch with the secret patch.
- (6) Once all the watermarks have been embedded, we resume the normal JPEG-2000 encoding process to generate the compressed bitstream.

Extraction of the secret data is the reverse of the embedding procedure.

Hereunder, we describe above watermarking steps in detail. Because Step (5) is the key step of the proposed method, we will firstly describe the whole watermark embedding procedure of the BPCS method in Section 3.3.2. Section 3.3.3 addresses the embedding strength control by using WEBR. The embed-

dable place selection is detailed in Section 3.3.4. Finally, Section 3.3.5 outlines the watermark retrieval procedure.

3.3.2 Watermarking by BPCS method

In the proposed method, we adopted the JPEG-2000-BPCS steganography [88][89] for watermark embedding. JPEG-2000-BPCS steganography is basically an extension of the BPCS method proposed by Kawaguchi *et al.* in [87], which is based on the observation that the human visual system (HVS) is generally good at spotting anomalies in areas of homogenous texture, but less sensitive to in recognizing them in complex regions. Therefore, Kawaguchi *et al.* proposed to decompose the image into bit-planes and replace the noise-like regions of the *least significant bit* (LSB) bit-plane with the to-be-hidden data, which ideally also appears as noise. Due to the difficulty for the HVS to distinguish the differences between two noise-like areas, the BPCS method can achieve larger capacity than other traditional data-hidden methods without introducing more distortion.

In the BPCS steganography, the complexity of a patch of bit-plane is defined as the number of non-border transitions from 1 to 0 and from 0 to 1 both horizontally and vertically [88]. Figure 3.8 shows two patches with different complexities, where white represents 0 and black represents 1. Figure 3.8(a) contains so many changes that it looks like noise. So its complexity is smaller than that in Figure 3.8(b), which contains large uniform black or white areas. Therefore, the patches similar to Figure 3.8(a) can be replaced with other noise-like patches with little effect on the visual quality. In contrast, if the Figure 3.8(b) is used instead, it will cause visible distortion because of its regular simple patterns. It is easy to prove, for any $N \times N$ patches, the maximum com-

Figure 3.8: Two binary patches of 8×8 samples with different complexities, where white represents 0 and black represents 1. (a) $\psi(\mathcal{P}_a) = 72$, (b) $\psi(\mathcal{P}_b) = 26$. $\psi(\cdot)$ is to compute the complexity.

plexity ψ_{max} is $2N(N - 1)$ (as in the case with a chessboard pattern illustrated in Figure 3.9(b) is the case). The minimum complexity is 0 of course.

Similar with JPEG-2000-BPCS steganography, we treat the bit-planes of the quantized wavelet coefficients as binary images and used to embed secret data with conventional BPCS steganography. Hereunder, we will describe the procedure used in our proposed method. The watermark to-be-embedded is firstly organized into $N \times N$ binary patches, known as *secret patches*. If the complexity of the secret patch is less than a threshold ψ_T (in our case, $\psi_T = 0.3\psi_{max}$), we XOR (exclusive OR) each bit within the patch with the corresponding bit within a chessboard pattern (see Figure 3.9(b)) to increase its complexity. This operation is called *conjugation* and illustrated in Figure 3.9. Obviously, the less complex patch $\mathcal{P}$ becomes much more noise-like after the conjugation (see $\mathcal{P}^*$). Note that the conjugated patch can be recovered by applying conjugation twice: $\mathcal{P} = (\mathcal{P}^*)^*$. Then, we choose a bit-plane of the codeblock for watermarking, say the p_i -th bit-plane. Thus, the noise-like secret patch replaces a block of the bit-plane, which is of p_i magnitude and within a selected region of

Figure 3.9: An example of conjugation: $\mathcal{P}^* = \mathcal{P} \oplus \mathcal{W}_c$, where $\mathcal{P}^*$ and $\mathcal{P}$ represent the conjugated and the original patch respectively, and “ $\oplus$ ” denotes the conjugate operation. It easy to compute that $\psi(\mathcal{P}) = 26$ while $\psi(\mathcal{P}^*) = 86$. Note that $\psi(\mathcal{P}) = \psi_{max} - \psi(\mathcal{P}^*)$ hold true here. $\psi(\cdot)$ is to compute the complexity.

the codeblock. If a secret block is conjugated, this fact should be recoded in a conjugation map, which is required during watermark retrieval process. Finally, the watermarked wavelet coefficients are subjected to the remaining JPEG-2000 compression process.

The proposed embedding procedure differs from JPEG-2000-BPCS method in that our method inserts the watermark in to the bit-plane with an adjustable magnitude p_i , while JPEG-2000-BPCS always use the lowest bit-planes of wavelet coefficients for watermark hiding [88]. Therefore JPEG-2000-BPCS method comes with the drawback of fragility with respect to lossy compression. It means that if the watermarked image is re-compressed again, the hidden secret data will be destroyed [89]. Furthermore, the embedded JPEG-2000 bitstream cannot be decompressed at other bit-rates without destroying the hidden information. Thus it is not aligned with the fundamental idea behind the JPEG-2000 design — “compress once, decompress many ways.”

In order to make the embedded watermark robust enough, the most straightforward approach is to increase the watermark signal energy. But the higher

the watermark energy is, the more significant quality distortion would be. So choosing an appropriate magnitude p_i for the watermark is important to achieve a good balance between robustness and fidelity of the watermarked image. We will investigate embedding strength control in the following section.

3.3.3 Embedding Strength Control

Figure 3.10: The sign-magnitude representation of the quantized DWT coefficients.

The key for embedding strength control is to find an appropriate magnitude for the watermarks. Therefore, we examine the JPEG-2000 wavelet coefficients representation first. As sketched in Figure 3.10, the quantized wavelet coefficients are represented in sign-magnitude. The total number of bits P depends on the operating system of computers. The MSB indicates the sign (e.g. 0 if positive and 1 if negative), followed by the absolute magnitude, in which M_i bits contain an integer portion of the coefficient and the rest N_i bits contain the decimal portion. These coefficients, grouped as codeblocks $\{\mathcal{C}_i\}$, are encoded by the EBCOT algorithm. It scans the bit-planes within the integer portion and the decimal portion are simply discarded as insignificant. Therefore, we embed the watermark by replacing a bit-plane within the integer portion of the coefficients but avoid changing the sign bits (i.e. the MSB) and the decimal portion. Assuming a bit $w_b \in \{0,1\}$ is inserted in the p -th bit of the coefficient

c (the LSB is of index 0), the energy of the watermark signal W_c will be

$$\|W_c\| = \|w_b \times 2^p\| \leq 2^p, \quad (3.1)$$

where 2^p is in fact a scaling factor. Thus we may increase the watermark signal energy by shifting the watermark a few bits to the left (i.e. increasing p). This strategy matches the EBCOT bit-plane coding well since the bit-planes with higher magnitude will be coded earlier. But in order to keep image quality degradation in check, the value of p is usually adjusted according to the different subband characteristics.

In traditional JPEG-2000 watermarking methods like [82], the common practice is to assign to different subbands different scaling factors, which are often found by heuristic ways and result in unpredictable image degradation. Moreover, with multiple quality layers being employed in EBCOT, we know the incremental contributions from each codeblock can vary significantly (see Figure 3.5). Thus finding an appropriate scaling factor for each codeblock would be a formidable task in traditional ways.

In our method, we proposed a novel concept *watermark embedding bit-rate* (WEBR), which is able to adaptively adjust the bit-plane index p where the watermark will be inserted for each codeblock by taking advantage of the EBCOT mechanism. As mentioned in Section 3.2.2, given a target bit-rate (it is the WEBR in our case), the EBCOT rate-control mechanism can form the quality layers and locate the optimal truncation point for each codeblock in terms of R-D efficiency for the given rate. Therefore, in our proposed method, we use EBCOT to tentatively form the quality layers with a WEBR as the target bit-rate. For a codeblock $\mathcal{C}_i$, EBCOT will find out the truncation point n_i , where i indicates the serial number of the codeblock. Since the truncation points are

always aligned with the boundary of the passes of the bit-planes [71], we can easily deduce the corresponding cutting bit-planes where the truncation point n_i is located. Thus we can locate the bit-plane p_i that is just *below* the cutting bit-plane for watermarking.

The proposed method offers several advantages. Most importantly, the data embedding is distortion-bounded. By *distortion-bounded*, we mean if the image is watermarked with $R_{web} = B$ bpp, the PSNR of the resulting image will not be less than the PSNR of the image compressed at the same bit-rate B bpp. This results from the fact that the bit-planes higher than p_i are kept intact during watermarking so that the quality layers corresponding to $R \geq R_{web}$ will not be affected. The EBCOT algorithm has provided a way to estimate the additional distortion in wavelet domain along with the generation of bitstream [69]. Assume that for a given truncation point n_i in the codeblock $\mathcal{C}_i$, the associated distortion is $D_i^{n_i}$. The overall distortion D is the sum of the distortions of all codeblocks, i.e.,

$$D = \sum_i D_i^{n_i} = \sum_i \left\{ \omega_{b_i}^2 \sum_{\mathbf{k} \in \mathcal{C}_i} (s_i^{n_i}[\mathbf{k}] - s_i[\mathbf{k}])^2 \right\}, \quad (3.2)$$

where i denotes the serial number of a codeblock, $s_i[\mathbf{k}]$ denotes the subband samples within $\mathcal{C}_i$, $s_i^{n_i}[\mathbf{k}]$ denotes the quantized representation samples associated with n_i , and ω_{b_i} denotes the L2-norm of the wavelet basis function for the subband b_i to which the codeblock $\mathcal{C}_i$ belongs. While it seems complex at first glance, this computation can be simplified with the aid of two lookup tables. For more information, please refer to [69]. We omit them here for simplicity. Therefore, using WEBR allows us to predict the distortion of the host image even before the watermark is embedded.

Besides, since only bit-planes at p_i and below are affected by watermark em-

bedding, the proposed watermarking scheme does not degrade the JPEG-2000 coding efficiency greatly. In addition, the implementation is simple. We just reuse the EBCOT algorithm for the embedding strength control and no further programming is required. Experiments show that the setting of p_i achieves pretty good balance between image quality and robustness of the watermark.

3.3.4 Embeddable Place Selection

In order to achieve better perceptual quality, the choosing of embeddable spatial regions within subbands is of the same importance. As mentioned in Section 3.3.1, we divide the host codeblocks into patches. For each patch, we assign a *visual factor*, based on which we choose the embeddable places. Assuming the visual factor is $F_{m,i}$ (m and i are the serial number of the current patch and the codeblock, respectively), we compute the value by jointly considering the average pixel value and the complexity of the patch:

$$F_{m,i} = w_{b_i}(S_{m,i} + \mu \cdot G_{m,i}) \quad (3.3)$$

where $S_{m,i}$ indicates the local texture complexity, $G_{m,i}$ denotes the average value of the pixels of the original image corresponding to the patch, and μ is a user-defined weighting factor for the trade-off between $S_{m,i}$ and $G_{m,i}$. In addition, considering the HH, LH, and HL subbands are of different importance to image visual quality, a scaling factor w_b can be assigned for a subband b_i .

In Equation 3.3, the local texture complexity $S_{m,i}$ is used to exploit the fact that the HVS is generally good at spotting anomalies in areas of homogenous texture, but less sensitive to see them in complex areas. In our implementation,

Figure 3.11: Mapping the embedding patches from the wavelet domain to the spatial domain.

we define it as the sum of the complexity of the nonzero bit-planes above p_i , i.e.

$$S_{m,i} = \sum_{l=p_i}^{P-1} \psi_{m,i}^l, \quad (3.4)$$

where $\psi_{m,i}^l$ denotes the complexity of the E -th bit-plane, which is computed in the same way of the BPCS method.

Furthermore, it is noticed that human eyes are less sensitive to the changes in the dark areas. Therefore, in Equation 3.3, $G_{m,i}$ is the luminance of current patch, which in fact is the average pixel value of the region within the host image corresponding to the patch. Obviously, a lower average value indicates a darker area. The correspondence is illustrated in Figure 3.11. The gray blocks in HL, LH and HH subbands correspond to the same spatial area in the original image.

For all of the candidate host patches, the embeddable place are selected according to their $F_{m,i}$ from high to low. Based on the user's specification or

the number of embeddable patches in each decomposition, we can decide the decomposition level L for watermark embedding. Note that we never cast the watermark into blocks of the DC subband since it may lead to serious fidelity degradation in the watermarked image.

3.3.5 Watermark Retrieval

Figure 3.7 illustrates the watermark retrieval procedure when the received image file is in compressed form. In order to extract the watermark, we need the same WEBR that is used in embedding process as well as the received JPEG-2000 image. The received watermarked image can be in either uncompressed form or compressed form. In either case, we encode or decode the watermarked file at WEBR to find out for each code-block the truncation points, based on which we can decide possible watermarked bit-plane p'_i and the decomposition level L' for watermark extraction.

Note that if the received watermarked image undergoes any incidental distortions and malicious attacks, it is possible that the p'_i found in the watermark retrieval process is unequal to the p_i found in the embedding process. The changing of wavelet coefficients will make the EBCOT finding wrong truncation points and therefore the watermarking bit-plane p'_i . For the incidental distortions we intend to resist, including multiple cycles of recompression and format conversion from one to another format such as the JPEG and BMP format, we find that p'_i is around $p_i \pm 1$ in the worst cases. It is because the proposed method replaces complex areas within the original image with similar complex watermark blocks. The number of bits after watermarking is kept relatively the same as that as before watermarking. Therefore, in our experiments, the EBCOT can find the watermarking bit-planes directly in most code-blocks. Only in some

code-blocks, we need to double-check neighboring bit-planes. In addition, generally several secret patches have been embedded in one codeblock. If several secret patches have been successfully extracted from the same bit-plane, we can be sure that the bit-plane is the watermarked bit-plane.

For each binary block of $N \times N$ samples extracted from the candidate bit-planes, the following steps are applied: First of all, BCH decoding is applied to identify the true secret patch. Since the watermark in each secret block is BCH encoded before embedding, the true watermarked patch must be decodable. We try to decode both conjugated patch and non-conjugated patch if the conjugated map is not available. However, it is possible that both the conjugated patch and non-conjugated patch are decodable. Even though it is rare to appear, in order to reduce its possibility, the complexity requirement and the embedability criteria, which has been used by the embedder, are employed to further verify the candidate patch. If all of the above processes succeed, a block of watermark message is extracted. In the above process, if the conjugation map is available, the process can be simplified.

3.4 Experiments and Results

In this section, we show some experimental results to demonstrate the robustness and the resultant visual quality of the proposed watermarking method.

Test Conditions

The proposed watermarking method has been integrated into *JJ2000*, a free Java reference software of JPEG-2000 emerging from the JJ2000 group [74]. The modularized architecture of the JJ2000 software eases our implementation. If not noted otherwise, we use the default parameters of JJ2000 for our exper-

Table 3.2: The test images and the number of embedded watermark bits.

Image	Size (width×height)	Watermark bits
Cafe	512 × 640	5120
Woman	512 × 640	5120
Lena	512 × 512	4096
Mandrill	512 × 512	4096

iments. This means that the lossy Daubechies 9/7 filter-bank is employed to decompose the images into a five-level representation and the EBCOT algorithm generates a *layer progressive* code-stream, i.e. in the LRCP progression order⁶, with multiple quality layers.

The watermark to be embedded is actually a pseudo-random binary sequence. The watermark bits are grouped as 8 × 8 secret patches. Then, the devised patches are inserted during JPEG-2000 compression. In order to implement the BPCS method, we set the complexity threshold $\psi_T = 0.3\psi_{max}$. It means that if the complexity of a secret patch is lower than $0.3\psi_{max}$, it should be conjugated into a more noise-like patch. In addition, the most appropriate subband and decomposition level for embedding can be chosen by the EBCOT algorithm automatically according to the user-specified WEBR. In following experiments, the lowest wavelet decomposition level is chosen.

Table 3.2 lists the four images (see Appendix 3A) used in our experiments. “Cafe” and “Woman” are anchor images for JPEG-2000 testing; “Lena” and “Mandrill” are classic test images.

The Quality After Watermarking

First of all, we will demonstrate our claim that the distortion resulted from our watermarking scheme is bounded, i.e. the image watermarked at WEBR R_{webR}

⁶Layer-Resolution-Component-Positionprogression order.

Table 3.3: Picture quality comparison (PSNR in dB). In case (a), the test images are encoded with different rates; in (b), the images are watermarked at WEBR and truncated at the same rates.

Bit-rate(bpp)	0.5	1.0	1.5	2.0	3.0
Cafe (a)	21.927	24.812	26.959	28.819	31.766
Cafe (b)	21.866	24.761	26.935	28.827	31.738
Lena (a)	32.826	34.815	35.916	36.882	38.378
Lena (b)	32.825	34.693	35.715	36.698	38.137
Mandrill (a)	23.007	24.922	26.372	27.518	29.190
Mandrill (b)	22.934	24.898	26.361	27.518	29.183
Woman (a)	32.544	36.235	38.605	40.618	43.252
Woman (b)	32.504	36.175	38.255	40.187	42.575
Avg. $ \Delta $	0.041	0.064	0.147	0.156	0.238

will not give an inferior quality compared with the image compressed at the rate R_{cmpr} , as long as $R_{webr} \leq R_{cmpr}$. The bound is reached when $R_{webr} = R_{cmpr}$. Table 3.3 shows the experimental results. In case (a), we compressed the test images from 0.5 bpp to 3.0 bpp; in case (b), we watermarked the images with a set of WEBRs and then truncated them with the same rates. We compared the PSNR values of the resulted images. Obviously, the difference is very limited. In fact, in case (b), the watermark have been removed by the truncation operation. The PSNR differences mainly come from the lossy compression and quantization errors rather than from watermarking.

While the above table demonstrates the lower bound of the quality of the watermarked images under different WEBRs, in practice, the WEBR should always be less than the compression rate. Thus, we do the following experiments: in case (a), we compress the original image at a bit-rate; Then we conduct watermarking with different WEBRs (b),(c), and (d), and the resulted images are also compressed at the same bit-rate of (a). We measure the resulted file sizes and PSNR values.

Table 3.4 and 3.5 show the experimental results for “Café” and “Mandrill”,

Table 3.4: Experiment results for “Café”

Image	WEBR(bpp)	Rate(bpp)	Secret Bits	File Bytes	PSNR(dB)
(a)	-	3	-	122,830	31.77
(b)	0.5	3	5120	122,981	28.15
(c)	1	3	5120	122,775	29.25
(d)	1.5	3	5120	122,785	29.50

Table 3.5: Experiment results for “Mandrill”

Image	WEBR(bpp)	Rate(bpp)	Secret Bits	File Bytes	PSNR(dB)
(a)	-	5	-	163,815	32.65
(b)	0.5	5	4096	163,681	29.74
(c)	1.5	5	4096	163,671	31.42
(d)	3	5	4096	163,570	31.39

respectively. Figure 3.12 and 3.13 demonstrate the perceptual quality of the watermarked images. From Table 3.4 and 3.5, we see that as the WEBR decreases, which means greater embedding strength, the PSNR value of the resulted images also decreases⁷. Comparing Table 3.3 and Table 3.4 and 3.5, we also notice that the PSNR of the images watermarked at WEBR is always higher than that of the images compressed at the same bit-rate. For example, when “Cafe” is compressed at 0.5 bpp, its PSNR is 21.927 dB (refer to the first row of Table 3.3); when the same image is watermarked at WEBR=0.5 bpp, its PSNR is 28.15 dB, obviously higher than that of its compressed counterpart. According to this rule, we can estimate the PSNR value of a watermarked image even before actual watermark embedding.

⁷Readers may notice that in Table 3.5 as WEBR increases from 1.5 to 3, the PSNR drops a little. It is resulted from file length decrease, which drops from 163,671 bytes to 163,570 bytes. Comparing the cases of WEBR=0.5 and WEBR=1.5, we see even when the file length decreases a little, the PSNR increases with the WEBR increasing. By the way, the file length after compression is automatically decided by the EBCOT in order to achieve the optimal rate-distortion trade-off.

(a)

(b)

(c)

(d)

Figure 3.12: Experiment results for “Cafe”. All of the above images are compressed at 2.5bpp. Figure (a) does not contain any watermark and Figure (b), (c), and (d) are embedded with 5120 bits at WEBR equal to 0.25, 0.5, and 1.5 bpp, respectively.

(a) no watermark

(b) WEBR=0.5 bpp

(c) WEBR=1.5 bpp

(d) WEBR=3 bpp

Figure 3.13: Experiment results for "Mandrill". All of the above images are compressed at 5 bpp. Figure (a) is not watermarked and Figure (b), (c), and (d) are embedded with 4096 bits at WEBR equal to 0.5, 1.5, and 3 bpp, respectively

Experimental Results of Robustness Tests

Next, we will examine the robustness of the proposed watermarking schemes. First, we consider the compression attacks, i.e. to compress the watermarked images with other coding schemes or at lower bit rates. JPEG-2000 itself and DCT-based JPEG codec are the two compression attacks under test. The attacked images are encoded into JPEG-2000 stream for watermark retrieval. Then we compare the similarity between the original, W , and the extracted watermark sequence W^* . In all of the following experiments, we use normalized correlation as the measure

$$\rho = \frac{W \cdot W^*}{\|W\| \cdot \|W^*\|}. \quad (3.5)$$

Figure 3.14 demonstrates the robustness of our method against JPEG-2000 compression. All the test images are watermarked at WEBR equal to 0.6 bpp and then compressed with full rate. Then, the compressed images are truncated at smaller target rates from 0.6 bpp to 2.6 bpp. This process simulates possible JPEG-2000 recompression attacks at different rates. The truncation process may remove some watermark bits. So we perform watermark retrieval on the truncated images and compute correlation. Figure 3.14(a) shows the experimental results of the average value of the normalized correlation of an embedded secret patch. Figure 3.14(b) shows the minimum normalized correlation value of all patches. We can see that when the truncation rate is approaching WEBR, more watermark bits will be lost. It is because the watermark bits are allocated by EBCOT into the layers just a little above WEBR. If the compression rate is higher than 1.2 bpp, almost all bits can be correctly extracted if we adopt error correction codes (ECC) to increase the robustness. Figure 3.14(b) give the worst case of all secret patches. It will give us the clue to choose ECC codes

with sufficient error-correcting capability.

The proposed watermark scheme can also resist some JPEG compression. We embed the images with WEBR equal to 0.5 bpp and 1.0 bpp respectively. Then, we decompress the watermarked JPEG-2000 code-stream and attack the watermarked images by compressing it again with lossy JPEG with varying compression factors. The experimental results are shown in Figure 3.15. In both (a) and (b), the correlation decreases as the JPEG compression factor decreases from 100 to 80. As we know, the proposed method inserts a watermark in form of noise-like patches. It is virtually equal to adding a high-frequency signal into the images. Since the JPEG standard compresses the images by discarding the high-frequency components of the DCT coefficients as redundant and insignificant information, the watermark signal is consequently vulnerable. But comparing Figure 3.15 (a) with (b), the watermark embedded at WEBR = 0.5 bpp is obviously more robust than that embedded at WEBR = 1.0 bpp. Therefore, we can increase the robustness against JPEG compression attack by increasing the WEBR for embedding.

The proposed watermark method is *semi-fragile* because it can not resist some attacks. Table 3.6 shows the experimental results for common types of attacks by using *StirMark*⁸ [91]. All of the images are watermarked at WEBR 0.5 bpp. We find that the smoothing filter like Gaussian filter and Median filter have greater effects to the embedded watermark than the sharpening filter. It still comes from the fact that we hide high-frequency watermark signals into the images. As for the additive noise, the proposed watermark scheme can resist little amount of noise.

⁸A software for simple robustness testing of image watermarking algorithms [97][98].

(a) Average correlation per patch

(b) The minimum correlation of all patches

Figure 3.14: The robustness against JPEG-2000 compression. Measured using normalized correlation between recovered and embedded message.

(a) WEBR=0.5bpp

(b) WEBR=1.0bpp

Figure 3.15: The robustness against JPEG compression. Measured using average normalized correlation for the watermark bits of a patch between recovered and embedded message.

Table 3.6: Experiment results for some types of *StirMark* attacks

Attack Types	Woman	Lena	Mandrill	Cafe
3×3 Gaussian filtering	0.2441	0.2246	0.0801	0.1602
3×3 Sharpening filter	0.8461	0.7998	0.4067	0.4316
3×3 Median filter	0.0500	0.0210	0.0288	0.0605
Adding noise 0.5	1.0000	0.9995	0.9951	0.9977
Adding noise 1	0.4898	0.5020	0.7827	0.9582
Adding noise 5	0.0277	-0.0083	0.0771	0.3359

3.5 Conclusions

In this chapter, we present a novel blind semi-fragile watermarking method for JPEG-2000 image compression standard, which increasingly gains popular due to many desirable features including high compression rates and scalability. Based on our observation on EBCOT, we introduce a new parameter *watermarking embedding bit-rate* (WEBR) to quantitatively control the watermarking strength. Given WEBR, we can find the appropriate bit-planes of wavelet coefficients for embedding and thus the quality layers above WEBR are untouched, which implies that the watermarking is distortion bounded. Then we replace them with watermark bits by taking advantage of BPCS steganography method. Experimental results show that the proposed watermarking method provides very high capacity and has stricken a good balance between robustness and invisibility. The proposed method is therefore suitable for image authentication.

Appendix 3A Test Images

The images shown in Figure 3.16 are used to test the proposed image watermarking technique. “Cafe” and “Woman” are anchor images for JPEG-2000 standard testing and are 512 by 640 pixels large; “Lena” and “Mandrill” are classic test images and are 512 by 512 pixels large.

(a) Cafe

(b) Woman

(c) Lena

(d) Mandrill

Figure 3.16: The test images.

Chapter 4

Watermarking H.264/AVC Videos

4.1 Introduction

Digital watermarking has focused on images for a long time. Nowadays the trend has changed with the development of video applications. It has become more and more popular to watch movies at home with DVD players, large screen HDTVs, or PCs with broadband connection rather than to go to the cinemas [1]. In addition, powerful replication tools enable people to share and exchange their favorite clips with ease, and at the same time almost uncontrollably. Facing such challenges, industries and researchers have scrambled to look for solutions to protect the owner's copyright and to prevent the content from tampering. Video watermarking techniques have therefore been proposed [15].

Most of the time, digital video contents, such as DVD, are sold and circulated in a standard-compliant format, in which the video is compressed. Obviously, it will be much more convenient to embed the watermark into and retrieval from the compressed video stream directly. Therefore, compressed-domain wa-

termarking methods arouse great interests among researchers. However, most of the previous works were focusing on MPEG-2 standard and therefore more suitable for high bit-rate applications like DVD and digital TV than low bit-rate videos ($1 \leq \text{Mbit/s}$). Few of them deal with the newest video coding standard H.264/AVC.

In this Chapter, we will endeavor on thoroughly investigating H.264/AVC and developing watermarking techniques. Two of the most important syntax elements of H.264/AVC, namely, *DCT coefficients* and *motion vectors*, have been exploited. First of all, we propose a *robust* watermarking method to embed watermark into the DCT coefficients. Our approach is similar with Hartung's [14] and Alattar's [41] in that we employ the spread-spectrum technique to spread the to-be-embedded message over the whole one frame. Our approach has several enhanced features. The 4×4 integer DCT transform are supported and only one of the coefficients are modified instead of all of the coefficients. Our approach also supports the Lagrangian optimization for optimal partition mode selection. The performance of watermark retrieval has been further improved by a slide window algorithm for frame accumulation. In addition, the gain of the watermark signal is based on the local luminance and activity characteristics. Experimental results show that our approach is robust against various attacks while striking a good rate-distortion balance for low bit-rate videos.

Motion vectors (MVs) are also exploited for watermarking. A novel *MV fragile* watermarking technique is proposed, which yields high embedding capacity. Traditional MV watermarking methods [33], [35], [36] modify MVs directly. However, noticing the *motion vector difference* (MVD), which is the prediction error between the current MV and its prediction from adjacent blocks, is the real syntax element of the H.264/AVC bitstream, we modify MVDs for data-hiding instead of MVs. In addition, a threshold is used to find appropriate MVDs

for watermarking. Again, the Lagrangian optimization method is featured in our approach to find the optimal MV after watermarking. Consequently, the objective quality loss (in PSNR) due to watermarking is almost negligible.

Furthermore, we propose a hybrid scheme by combining above two proposed techniques for *semi-fragile* watermarking, which is capable to detect malicious attacks, but robust to some extent of content preserving modifications, such as transcoding and recompression [100]. Our fundamental idea comes from Yin *et al.* [92] and Park *et al.* [93], in which a fragile watermark and a robust watermark are jointly used to construct a semi-fragile watermarking system for MPEG-2 video authentication. In contrast with Yin's method, our approach embeds the fragile watermark into MVs rather than in the LSBs of DCT coefficients, while the robust watermark is embedded into DCT coefficients of both I-frames and P-frames rather than I-frames alone. Therefore, our watermarking scheme can achieve greater capacity and other merits without introducing visible impairments.

An overview of H.264/AVC is presented in Section 4.2. The DCT-domain watermarking method is given in Section 4.3 and the MVD watermarking method is described in Section 4.4. The performance of the two techniques will be demonstrated by experimental results respectively. Section 4.5 presents the hybrid watermarking scheme, followed by the conclusions in Section 4.6.

4.2 Review of H.264/AVC

H.264/AVC [21] is the latest video coding standard developed by the *Joint Video Team* (JVT) of ITU-T VCEG and ISO/IEC MPEG. It has been approved by ITU-T as Recommendation H.264 and by ISO/IEC as MPEG-4 Part-10 *Advanced Video Coding* (AVC). Experiments have shown that H.264/AVC has achieved its

Figure 4.1: Block diagram of the H.264 video coding layer (VCL).

goals to provide a network-friendly ratedistortion efficient video representation addressing *conversational* (e.g. videoconferencing) and *non-conversational* (e.g. broadcast and streaming) applications ” [103]. Two major improvements are:

Improved coding efficiency Compared with existing video coding standards, H.264/AVC permits an average reduction in bit rates by average 50% or greater at most bit rates while achieves essentially the same objective PSNR reproduction quality [26] [39].

Network friendliness H.264/AVC conceptually separates into two layers: the *video coding layer* (VCL) responsible for efficiently representing the video content, and the *network adaptation layer* (NAL) responsible for packaging and conveying the data in a manner appropriate to the network on which it is used [38]. In addition, error resilience tools and other network functionalities, which have been adopted by prior standards such as MPEG-4, have been further improved [25] [38].

For a more detailed review of this state-of-the-art standard, please refer to [38],[39], and [40]. In the following, we will introduce the H.264/AVC video coding algorithm.

The H.264/AVC is similar to previous video coding standards in that it is basically a block-based motion-compensated hybrid transform coder.¹ The H.264/AVC *video coding layer* (VCL) is sketched in Figure 4.1, with which high compression ratio has been achieved. It is different in many specifics with previous standards that considerably affect the performance and complexity.

Intra Coding

In intra coding, the macroblocks (a block of 16×16 pixels) are coded without referring to the previous frames. Two types of intra coding are supported, namely, INTRA- 4×4 and INTRA- 16×16 . INTRA- 4×4 divides a macroblock into 4×4 blocks and encodes each separately. It is suitable for parts of a frame with significant details. INTRA- 16×16 , on the other hand, encodes the whole macroblock once for all and is more suitable for smooth areas. In order to improve intra coding efficiency, samples are predicted with neighboring previous encoded and reconstructed samples. In contrast to H.263++ and MPEG-4, where the prediction is carried out in the DCT domain, H.264/AVC is unique in conducting the prediction in *spatial domain*. Figure 4.2 illustrates the first six of the nine optional prediction modes for INTRA- 4×4 , where the pixels within the block $\{a, b, \dots, p\}$ are predicted by the neighboring pixels $\{A, B, \dots, M\}$. Each 4×4 block can select one of nine modes for encoding. Except for Mode-2 “DC” mode, where the entire 4×4 block is predicted by the mean of neighboring pixels, the other modes calculate prediction values according to the

¹For more information about general techniques and basic terminologies for video coding, please refer to Section 2.3.2.

Figure 4.2: First *six* of the nine INTRA-4 x 4 luma prediction modes, where the samples {a, b, ..., p} within the block are predicted by the neighboring samples {A, B, ..., M}.

different directions. For this reason, this prediction method is called *directional spatial prediction*. Similarly, INTRA-16 x 16 supports four spatial prediction modes [21].

Inter Coding

Motion estimation and compensation is performed in inter coding to predict current block from one or more previously encoded frames. The resulting MVs and prediction errors are encoded and transmitted. In contrast with prior standards like MPEG-4 and H.263++, which exploit two block-size types of 16 x 16 and 8 x 8 pixels, H.264 supports tree-structured *motion segmentation* with multiple block-size *modes*, varying from 16x16, 8x16, 16x8, 8x8, 8x4, 4x8 to 4x4 pixels, as illustrated in Figure 4.3. So up to 16 MV are available for a macroblock. Moreover, MVs are represented in quarter-pixel resolution in H.264/AVC and

Figure 4.3: Block modes available for motion compensation in H.264/AVC. For each block, a motion vector is encoded. A 16 x 16 macroblock can be divided down to 8 x 8 blocks, each of which can be further divided down to 4 x 4 blocks.

prediction values at quarter-pixel locations are interpolated from integer samples by applying a six-tap filter. As a result, the inter prediction can be more efficient than previous standards. *Multi-frame motion-compensated prediction* [27] is also supported, allowing the encoder to use more than one previously decoded picture as the reference frames.

Integer Transform

H.264/AVC is unique in that it employs a purely *integer DCT transform* (approximates the DCT). It is primarily 4 x 4 pixels in shape, as opposed to the 8 x 8 shape typically found in prior designs. The transform matrix is given as

$$\mathbf{H} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 1 & -1 \end{bmatrix}. \quad (4.1)$$

The inverse transform is also specified in terms of precise integers, rather than as a rounding tolerance relative to a real-valued ideal transform as found in most prior designs. Therefore, all operations can be carried out with integer arithmetic, such as adding and shifting. Thus, the inverse-transform mismatch issue existing in previous standards is eliminated.

Entropy Coding

Entropy coding is performed either with a single *context-adaptive variable length coding* (CAVLC) table for all syntax elements, or with the *context-based adaptive binary arithmetic coding* (CABAC). While CAVLC is easier for implementation, CABAC can typically provide 5%-15% more bit rate saving compared with CAVLC [38].

In-Loop Deblocking Filter

“Blocking artifacts” is the one of the most prominent artifacts with the present block-based video coding scheme. For this reason, an adaptive deblocking filter is used within the prediction loop to remove block-edge artifacts.

Video Coder Control with Lagrangian Optimization

From above discussion, we see that the H.263/AVC standard provides far more coding options (or *modes* in jargon) than previous standards. It is the task of video coder control to find a combination of coding options I^* out of all options $\mathcal{O}$ for a source S that can minimize the distortion D subject to a given rate R_c [39]:

$$I^* = \arg \min_{I \in \mathcal{O}} \{D(S, I)\}, \text{ subject to } R(S, I) \leq R_c. \quad (4.2)$$

To solve this optimization problem, the *Lagrangian optimization technique*

[28], [30] is adopted in H.264 due to its effectiveness and simplicity.. Given quantization parameter (QP) and Lagrange multiplier λ_{mode} , the optimal mode for a block $\mathcal{S}_k$ is found by minimizing the Lagrangian function

$$J_{mode}(\mathcal{S}_k, I_k|QP, \lambda_{mode}) = D_{rec}(\mathcal{S}_k, I_k|QP) + \lambda_{mode} R_{rec}(\mathcal{S}_k, I_k|QP, \lambda_{mode}), \quad (4.3)$$

where the block mode I_k is varied over all possible coding modes available for a particular frame type. For completeness, we list the modes used in I-frames and P-frames as follows:

- *I-frame*: INTRA-16 x 16, INTRA-4 x 4 and their prediction modes.
- *P-frame*: INTER-16 x 16, INTER-16 x 8, INTER-8 x 16, INTER-8 x 8, INTER-8 x 4, INTER-4 x 8, INTER-4 x 4, and SKIP².

Generally, the distortion $D_{rec}(\mathcal{S}_k, I_k|QP)$ is measured as the *sum of the squared differences* (SSD) between the original block s and its reconstruction $\hat{s}$ given as

$$SSD(s, \hat{s}|QP) = \sum_{(X,Y) \in \mathcal{S}_k} (s[x,y] - \hat{s}[x,y|QP])^2, \quad (4.4)$$

where the size of $\mathcal{S}_k$ depends on the chosen mode. If INTER- $A \times B$ or INTRA- $A \times B$ mode is used, $\mathcal{S}_k$ is of $A \times B$ pixels. $R_{rec}(\mathcal{S}_k, I_k|QP, \lambda_{mode})$ is the number of bits associated with the chosen *mode* and QP , including the bits for the macroblock header and all DCT blocks coefficients, as well as the motion vectors if one of INTER- X modes is used.

²For a macroblock in SKIP mode, no data is coded except its identification and, if any, motion vector.

4.3 Watermarking DCT Coefficients

As mentioned before, H.264/AVC is unique in adopting a 4×4 DCT transform. With the help of the improved prediction process for both intra and inter coding [38], the smaller block-size not only reduces the ringing artifacts but also improves the coding efficiency. However, higher compression ratio means fewer nonzero DCT coefficients. These coefficients are perceptually important so that changing it may lead to greater impairment. Therefore, the balance among capacity, robustness, and imperceptibility are more difficult to achieve in H.264/AVC watermarking [96]. Even the watermarked DCT coefficients would be easily changed due to mode changes during recompression. Therefore the class of watermarking methods which enforce a relationship between coefficients [55], [56] is not applicable for H.264/AVC.

In the proposed method, we use spread-spectrum watermarking technique for robust data-hiding. The watermark signal is generated by spread-spectrum technique from the message to be embedded and then spread over the whole frame. For each 4×4 block, only one AC coefficient is modified to embed one watermark bit. And the gain of the watermark is controlled by local characteristics analysis. In order to be compliant to the H.264 JM encoder, the Lagrangian optimization algorithm is used to select the best mode for a watermarked macroblock. Above approach is applied to not only I-frames but also P-frames to achieve greater capacity. Furthermore, frame accumulation is employed for P-frames to improve the watermark retrieval performance.

4.3.1 Watermark Formation

The direct sequence spread-spectrum technique similar with Alattar's [41] is applied to the message to-be-embedded to generate the watermark prior to

insertion into the video frames. It is well known that in spread spectrum communications, a narrow-band signal (the message in our case) is modulated with pseudo-noise sequences signal to spread the signal over a much wide frequency band. Consequently, the power of the resulting signal is much below than the noise floor and thus immune to interference and interception. Please refer to Chapter 2 for more review on spread-spectrum watermarking methods.

For each *Group of Picture* (GOP)³, we embed a 48 bits message to the I-frame and another 48 bits message to all the other P-frames. The spread-spectrum watermarking technique firstly maps the binary message $V = \{v_i\}$, $v_i \in \{0,1\}$ to bi-polar signals. That is, 0 is mapped to 1; 1 is mapped to -1. Then each message bit is multiplied by a different pseudo-random bi-polar sequence of maximum hamming distance(also known as chip sequence) to produce the watermarks $W = \{w_i\}_{i=0}^N$, $w_i \in \{-1,1\}$. As a result, each message bit is in fact represented by a pseudo-random sequence. Moreover, considering activities can vary greatly within a frame and some smooth regions almost cannot hide any bits, we further scatter watermark bits pseudo-randomly over the whole frame.

4.3.2 Perceptual Embedding

One watermark bit is embedded into a 4×4 luminance block by altering the one of mid-frequency AC coefficients, as illustrated in Figure 4.4. The original AC coefficient $X(m, n)$, where m and n indicate the position of the coefficient, is replaced by the product of the watermark w_i , the perceptual mask M_i generated by the perceptual analysis and a global gain factor a . The resulting coefficient $X^*(m, n)$ is given by

$$X^*(m, n) = \alpha M_i w_i, \quad (4.5)$$

³A set of frames from one I-frame (included) to the next I-frame (excluded) is refer to as a GOP. We only consider “IPPP” GOP structure in this chapter.

Figure 4.4: Watermark embedding scheme.

where both α and M_i are positive values and will be discussed later. The watermark bit is therefore indicated by the polarity of the watermarked AC coefficient. In our experiments, we choose the AC coefficients $X(2, 2)$ for watermarking. Our experiments show that the coefficients in the diagonal positions are stabler than the others, especially in intra modes where the directional prediction will have a greater impact for the coefficient in other locations. After watermarking, the quantization is performed as usual. The quantization may remove some watermarked coefficients and lead to watermark loss. So the higher the QP is, the fewer the watermark bits survive.

The above operations should be repeated for all the modes of I-frames and P-frames. To select the best mode for a watermarked macroblock $\mathbf{S}_k^*$, Equation 4.3 is adapted as

$$J_{MODE} = D_{REC}(\mathbf{S}_k^*, \mathbf{I}_k) + \lambda_{MODE} R_{REC}(\mathbf{S}_k^*, \mathbf{I}_k). \quad (4.6)$$

The best mode is selected by minimizing Equation 4.6.

In Equation 4.5, the tradeoff between robustness and the imperceptibility

depends on the choice of the global gain factor α and the visual mask M_i . The gain factor α is empirically decided by users for a whole video sequence. The visual mask M_i , also called local gain factor in this chapter, are generated by perceptual analysis for 4×4 blocks. It is used to adapt the strength of watermarking embedding to the local characteristics of the host video. We have found in the previous chapter that people generally hardly notice any changes within dark or textured areas. For relatively dark and ‘(busy’ (or textured) regions, the local gain factor increases the embedding power to increase the robustness. For relatively bright and smooth regions, where human eyes are more readily to spot any distortions, the local gain factor should be decreased to minimize the watermark visibility.

In our perceptual analysis process, we obtain the local gain factor on block-by-block basis and directly from the DCT coefficients $X(m, n)$ according to

where $X(0, 0)$ denotes the DC coefficient, $X(m, n)$ denotes the AC coefficient. Here, the value of the DC coefficient in fact represents the luminance of the block: A smaller DC value indicates a darker block; a higher DC value indicates a brighter block. The sum of the absolute value of the AC coefficients represents the spatial activity of the current block. They are balanced by adjusting the weighting factor μ . In Equation 4.7, it may useful to select $l \geq 2$ to avoid watermarking the blocks with strong bright and dark edges.

4.3.3 Retrieval by Frame Accumulation

We extract the watermark while decoding the watermarked, possibly distorted, video stream. For each 4×4 block containing AC coefficients, one watermark bit

is extracted according to the polarity of the quantized coefficient $X_q(m, n)$, i.e., the watermark bit $w_i = 1$ if $X_q(m, n) > 0$; $w_i = -1$ if $X_q(m, n) < 0$; otherwise w_i is deemed lost. Then, the decision of a message bit v_i from a sequence of bipolar watermark bits is made by measuring the similarity between the extracted watermark $W^* = \{w_i^*\}_{i=0}^N$ and the original spread watermark $W = \{w_i\}_{i=0}^N$. To simplify the implementation, the Hamming distance is used as the similarity measure

$$H(w, w^*) = N - \sum_{i=0}^N w_i^* \cdot w_i. \quad (4.8)$$

The message bit $v_i \in \{0, 1\}$ can be derived from the Hamming distance $H(w, w^*)$ with an appropriate threshold τ , i.e., if $H(w, w^*) \geq \tau$, the message bit is determined, otherwise the bit is deemed lost.

The above detection process is performed on each individual frame. Obviously, the number of surviving watermark bits varied with the frames since some frames are not suitable for watermarking. Considering we have repeatedly embedded the same watermark into every P-frames in a GOP, it will be desired to exploit the temporal redundancy for better detection performance. Intuitively, because the successive frames contains some difference, the watermark hidden in the successive frames are of different strengths and consequently different survival possibilities. The union of the surviving watermark bits of multiple frames are surely larger than the surviving watermark in a single frame. Therefore, we use a slide window algorithm for watermark retrieval. The watermark signal within the slide window are accumulated; the sum of the watermark signals are then used for message extraction with the same algorithm as one frame detection. Consequently, the number of bits recovered after frame accumulation is more stable and even able to overcome some extent of burst watermark losses. This property is especially valuable for applications that require at least one

detection within any given intervals [15].

4.3.4 Experiments and Results

Test Conditions

The proposed watermarking technique has been integrated into the H.264 JM-7.6 reference software [104]. Experiments have been conducted on the video sequences : *Foreman*, *Stefan*, *Coastguard*, *Flower Garden*, *Container Ship*, and *Silent* (see Appendix 4A). All sequences are of CIF size, i.e. 352 x 288 pixels. The first 100 frames of the sequences are used for experiments and encoded at a frame rate of 15 frames/s; The GOP structure comprises one I-frame followed by 49 P-frames, corresponding to about one I-frame per three seconds. The Lagrangian optimization described in Section 4.2 is used by encoders along with the rate control to compress the sequences to 768 kbit/s, 512 kbit/s, and 396 kbit/s respectively. The above configurations and bit-rates are compliant to the *Baseline Profile* of H.264/AVC and typical for low bit-rate applications, like videoconferencing and surveillance.

In each GOP, 96 bits secret messages are embedded, within which the I-frame is embedded with 48 bits, and the other 48 bits are repeatedly embedded in each P-frame of the GOP. Every message bit are pseudc-randomly spread by spread-spectrum technique over the whole frame.

Quality and Bit-rates After Watermarking

Table 4.1 and Figure 4.5 show the experimental results on the aspects of PNSR (in dB) and bit-rates (in kbit/s). In Table 4.1, we use rate-control to achieve the target bit-rates during encoding. Thus, the bit-rate difference between unmarked and marked videos is negligible. In fact, the effect of watermarking is

Sequence	Target Bitrate (kbit/s)	Unmarked		Marked			
		PSNR (dB)	Bitrate (kbit/s)	PSNR (dB)	Bitrate (kbit/s)	Δ PSNR (dB)	Δ Bitrate (%)
Foreman	768	41.55	774.44	41.02	774.55	-0.53	0.01
	512	39.97	518.44	39.57	518.44	-0.40	0.00
	396	39.01	402.39	38.68	402.48	-0.33	0.09
Stefan	768	36.55	782.47	36.01	782.56	-0.54	0.01
	512	34.67	526.49	34.35	526.48	-0.32	-0.00
	396	33.52	410.65	33.34	410.68	-0.18	0.03
Coast- guard	768	34.92	778.58	34.71	778.49	-0.21	-0.01
	512	33.05	522.37	32.95	522.49	-0.10	0.02
	396	32.03	406.47	31.96	406.47	-0.07	0.00
Flower Garden	768	34.14	787.84	33.59	788.92	-0.55	0.14
	512	31.78	532.30	31.37	533.82	-0.41	0.29
	396	30.46	416.63	30.12	416.34	-0.34	-0.29
Silent	768	44.06	776.23	40.91	776.15	-3.15	-0.01
	512	41.82	520.18	39.56	520.23	-2.26	0.01
	396	40.39	404.21	38.68	404.20	-1.71	-0.01
Container Ship	768	43.30	776.32	39.56	776.67	-3.74	0.05
	512	41.42	520.40	38.52	520.02	-2.90	-0.07
	396	40.30	404.42	37.91	404.47	-2.39	0.05
Average	558.7	37.39	570.05	36.27	570.19	-1.12	0.02

Table 4.1: The PSNR and bit-rate for the unmarked and marked video sequences.

(a) Foreman

(b) Stefan

(c) Coastguard

(d) Flower Garden

(e) Container

(f) Silent

Figure 4.5: Comparison of the rate-distortion performance for H.264 encoding without watermarking (o) and with DCT domain watermarking (x). All the sequences are encoded with fixed QPs.

represented in the drop of the PSNR values. On average, the watermarking leads to 1.1 dB decrease in terms of the PSNR objective quality measure. The degradation differs significantly between the first four sequences and the last two sequences. While in the first four sequences, the PSNR drops by 0.3 dB on average, the watermarking degrades *Container Ship* and *Silent* by almost 3 dB. This degradation can be contributed to the fact these two sequences contain only still background, slow moving objects, and little spatial activities. Thus, the P-frames carry only visually significant features of the video. Changing these features during watermark embedding process creates greater distortion. In addition, the quality degradation is more noticeable at higher bit-rates than at lower bit-rates. It is because at high bit-rates more watermark bits can be embedded due to the low quantization. In above situations, we can decrease the global watermarking gain factor to achieve better video quality. The above observations are further validated by the rate-distortion curves shown in Figure 4.5, in which the sequences are encoded with fixed QPs. The visual quality has been demonstrated in Figure 4.6–4.8, in which the 30th frame of the marked and unmarked sequences *Foreman*, *Stefan*, *Container Ship*, and *Silent* are juxtaposed for comparison. No artifacts are visible. Even in *Container Ship*, the impairments are almost unnoticeable. This proves the effectiveness of our watermarking strength control mechanism.

Experimental Results of Robustness Tests

The robustness of the proposed watermarking technique is tested in two categories: transcoding and common signal processing. Transcoding re-compresses the watermarked video stream to a lower bit-rate or converted to another video compression format [100]. Given it is the most common process to the compressed video streams, extensive tests have been conducted over all six test

(a) Unmarked (PSNR=41.01 dB)

(b) Marked (PSNR= 40.33 dB)

Figure 4.6: The 30th frame of the *Foreman* sequence (512 kbit/s, CIF-size) without watermarking and with DCT domain watermarking.

(a) Unmarked (PSNR= 36.32 dB)

(b) Marked (PSNR= 34.36 dB)

Figure 4.7: The 30th frame of *Stefan* (512 kbit/s, CIF-size) without watermarking (a) and with DCT domain watermarking (b).

(a) Unmarked (PSNR= 42.90 dB)

(b) Marked (PSNR= 40.17 dB)

Figure 4.8: The 30th frame of *Silent* (512 kbit/s, CIF-size) without watermarking (a) and with DCT domain watermarking (b).

(a) Unmarked (PSNR= 42.20 dB)

(b) Marked (PSNR= 38.70 dB)

Figure 4.9: The 30th frame of *Container Ship* (512 kbit/s, CIF-size) without watermarking (a) and with DCT domain watermarking (b).

Sequence	Original Bitrate	No Attack		To $\frac{1}{2}$ bitrate		To $\frac{1}{3}$ bitrate	
		$N = 1$	$N = 5$	$N = 1$	$N = 5$	$N = 1$	$N = 5$
Foreman	768	0.92	0.96	0.48	0.93	-0.29	0.48
	512	0.73	0.96	-0.28	0.52	-0.83	-0.41
	396	0.48	0.95	-0.55	0.17	-0.84	-0.45
Stefan	768	0.95	0.98	0.57	0.96	-0.35	0.44
	512	0.69	0.98	-0.34	0.48	-0.85	-0.50
	396	0.36	0.97	-0.63	0.06	-0.88	-0.55
Coast-guard	768	0.84	0.98	0.31	0.92	-0.58	0.18
	512	0.30	0.94	-0.60	0.16	-0.91	-0.64
	396	-0.11	0.77	-0.76	-0.21	-0.90	-0.59
Flower Garden	768	0.99	1.00	0.72	0.99	-0.18	0.60
	512	0.91	1.00	-0.12	0.65	-0.77	-0.24
	396	0.69	1.00	-0.40	0.43	-0.74	-0.17
Silent	768	0.95	0.96	0.58	0.94	0.14	0.78
	512	0.92	0.96	0.15	0.82	-0.45	0.34
	396	0.83	0.96	-0.06	0.67	-0.41	0.35
Container Ship	768	0.96	0.97	0.70	0.96	0.39	0.89
	512	0.95	0.97	0.45	0.91	0.08	0.77
	396	0.91	0.97	0.34	0.90	0.10	0.80
Average	558.7	0.74	0.96	0.03	0.63	-0.46	0.12

Table 4.2: The robustness against transcoding attacks. Measured using normalized correlation between the recovered and embedded message.

sequences. In our experiments, the watermarked videos are re-compressed to a half and one third of the original bit-rate, respectively. Furthermore, signal processing manipulation is another source of attacks. Here we apply some common signal processing to watermarked *Foreman*, *Stefan*, and *Container Ship*. These operations include Gaussian low-pass filtering, unsharp contrast enhancement, and adding Gaussian white noise (variance= 0.001). Without special notices, the Matlab default parameters are applied. Note that these manipulations are only applicable to the frames in spatial domain. Therefore, the watermarked video streams need to be decoded firstly and compressed again after processing. As a result, the additional compression also contributes to the watermark loss.

The robustness test results are given in Table 4.2 and Table 4.3, respectively.

Sequence	Foreman				Stefan				Container			
	1	3	5	8	1	3	5	8	1	3	5	8
3x3 Gaussian Filtering	0.17	0.62	0.76	0.83	0.2	0.69	0.81	0.87	0.55	0.83	0.89	0.91
5x5 Gaussian Filtering	0.16	0.61	0.75	0.81	0.21	0.71	0.83	0.87	0.54	0.83	0.89	0.9
Unsharp Filtering	0.43	0.75	0.85	0.91	0.35	0.71	0.83	0.9	0.57	0.86	0.92	0.94
Additive Noise	-0.17	0.01	0.09	0.15	-0.04	0.24	0.37	0.49	0.14	0.46	0.62	0.73
Average	0.15	0.50	0.61	0.68	0.18	0.59	0.71	0.78	0.45	0.75	0.83	0.87

Table 4.3: The robustness against common signal processing. Measured using normalized correlation between the recovered and embedded message under different sliding-window lengths N . All the three sequences are compressed at 512 kbit/s.

The accuracy of the extracted watermark is measured by the normalized correlation between the original message and the recovered message. Note that since some of the message bits can be lost due to manipulations, we set their value to be the reverse of original value. Therefore, when the correlation value is -1 , it means all of the message bits are lost; when the correlation value is 0, half of the message is lost.

Table 4.2 shows the experimental results of transcoding attacks. We present the normalized correlation by two metrics. The first is the per-frame correlation and indicates the ratio of number of watermark bits which can be correctly extracted independently from each P-frames ($N = 1$). The other is per-five-frame correlation. The detection is performed in a sliding window of $N = 5$ frames. The watermark bits within the sliding window are accumulated and the message bits are decided by the process as described in Section 4.3.3.

Table 4.3 shows the experimental results of common signal processing attacks. In addition, to have an appreciation regarding how the sliding window length affects the watermark retrieval performance, Table 4.3 gives more results under different N values. It has demonstrated that as N increases, more watermark bits can be correctly extracted. Moreover, when $N = 5$, major gain of the

watermark retrieval performance has been achieved. While further increasing N will improve the retrieval performance a little, considering large sliding window length will increase the memory consumption and computational complexity, it seems that the results with 5-frame sliding window are good enough. It is also the reason why I only show the result with $N = 1$ and $N = 5$ in Table 4.2.

The test results in Table 4.2 and Table 4.3 demonstrate that the proposed watermarking method is fairly robust against transcoding and these common signal processing attacks. The average normalized correlation per frame is 0.74 for un-attacked watermarked video streams. After transcoding to half of the original bit-rate, the correlation is 0.03. Higher correlation is achieved by using five-frame accumulation. The average correlation value per five frames is 0.63 for the video stream with half of original bit-rates. It means that more messages bits can be correctly extracted by using the sliding-window accumulation method. This is confirmed by the correlation value curves in Figure 4.10 for sequences *Foreman*, *Stefan*, and *Silent*. Although we watermark both I-frames and P-frames in our experiments, only the correlation values for P-frames are plotted to avoid confusion. It is the reason why there is a great drop of correlation value for the 1st and the 50th frame. Both are I-frames and indicate the beginning of a new GOP. Obviously, the per-frame correlation value varied greatly according to the scenes of the sequences. In contrast, the per-five-frame detection is more stable and even is able to overcome to some extent the burst watermark losses. In addition, we also notice in Table 4.2 *Silent* and *Container Ship* give the best correlation results after transcoding among the six test sequences. It means more watermark message bits are recovered in the two sequences. These results validate the fundamental tradeoff between the quality and the watermark robustness.

(a) Foreman (512 kbit/s)

(b) Foreman (256 kbit/s)

(c) Stefan (512 kbit/s)

(d) Stefan (256 kbit/s)

(e) Silent (512 kbit/s)

(f) Silent (256 kbit/s)

Figure 4.10: Normalized correlation results of the recovered watermark from P-frames for the sequences before (left) and after transcoding attacks (right). The sequences were firstly encoded at 512 kbit/s and then were re-compressed to 256 kbit/s. The sequences were encoded in a IPPP structure. The 1st and the 50th frames are I-frames and the others are P-frames.

Figure 4.11: Fractional sample search positions.

4.4 Watermarking Motion Vectors

The utilization of *motion compensation* to removing temporal redundancies between successive frames is one of the most prominent features of modern video coding standards, which does not exist in still-image compression methods. In fact, most frames in compressed video stream are inter-coded and contains large amount of motion vectors (MVs). It is a nature choice to exploit MV to carry watermarks.

Traditional MV watermarking methods [32]-[36] modify components of MVs according to a parity rule to embed watermarks. However, in H.264/AVC MVs will be predicted by adjacent MVs and only the *motion vector differences* (MVDs) will be stored in the final compressed stream. Therefore, we propose to embed MVDs instead of MVs [102]. The novel MVD watermarking technique will be described in following sections.

4.4.1 Data Embedding in Motion Vector Differences

A video coder computes MV on a block-by-block basis by a process known as the *block matching motion estimation* (BMME) (see Section 2.3.2). For each block, the motion search proceeds first over integer-pixel positions within a searching area and then the half-pixel locations around the best integer-pixel position, followed by a refinement search proceeding over the quarter-pixel positions $\{1 \sim S\}$, as illustrated in Figure 4.11, around the best half-pixel position A. Thus, MV is in fact the relative displacement from the current to-be-encoded block to the best-matching block in the reference frames.

In H.264/AVC JM encoder [104], the BMME is initiated from the location of the predicted MV, which is the median of previous coded MVs of neighboring blocks [21]. Therefore, it is the *motion vector difference* (MVD) $\mathbf{d}$ between the current MV $\mathbf{m} = (m_x, m_y)$ and the predicted MV $\mathbf{p} = (p_x, p_y)$ that is encoded and transmitted, i.e. $\mathbf{d} = (d_x, d_y) = \mathbf{m} - \mathbf{p}$. Since MVs of neighboring blocks are highly correlated, the number of bits required to encode the MVD $\mathbf{d}$ is greatly reduced compared with that of coding the MV $\mathbf{m}$. To find the optimum MV $\mathbf{m}_i$ within the searching range W , the BMME minimizes a Lagrangian cost function

$$\mathbf{m}_i = \arg \min_{\mathbf{m} \in W} \{D_{DFD}(\mathcal{B}_i, \mathbf{m}) + \lambda_{ME} R_{ME}(\mathbf{m} - \mathbf{p})\}, \quad (4.9)$$

where D_{DFD} denotes a measure of matching error, for example *sum of absolute difference* (SAD), between the original and its reference, λ_{ME} is the Lagrangian multiplier, and $R_{ME}(\mathbf{m} - \mathbf{p})$ is the number of bits spent in encoding the MVD.

In the proposed technique, we embed two bits data into one MVD by changing the quarter-pixel refinement BMME as follows. The motion estimation is performed as usual until the optimal quarter-pixel accuracy MV $\mathbf{m}_i$ is found

Set	w_1w_0	The LSB of (d_x, d_y)	MV locations
S_0	00	0, 0	A
S_1	01	0, 1	2, 7
S_2	10	1, 0	4, 5
S_3	11	1, 1	1, 3, 6, 8

Table 4.4: The MV classification in Figure 4.11 used for data embedding.

and MVD $\mathbf{d}_i$ is known. We choose $\mathbf{d}_i$ for watermarking only if the amplitude of $\mathbf{d}_i$ is greater than a threshold τ , i.e.

$$|\mathbf{d}_i| = \sqrt{d_x^2 + d_y^2} \geq \tau. \quad (4.10)$$

The decision of τ will be discussed in next section. For a selected $\mathbf{d}_i$, we classify the optimal MV location **A** and its neighboring quarter-pixel locations (**1** ~ **8**) (see Figure 4.11) into four set $\{S_0, S_1, S_2, S_3\}$, as shown in Table 4.4. Then, according to the bits w_0w_1 to be embedded, the quarter-pixel refinement BMME is performed again over the locations specified by S_n , $n \in \{0, 1, 2, 3\}$. The optimal MV $\mathbf{d}_i$ bearing watermark bits is selected according to

$$\mathbf{m}_i^* = \arg \min_{\mathbf{m}^* \in S_n} \{D_{DFD}(\mathcal{B}_i, \mathbf{m}^*) + \lambda_{ME} R(\mathbf{m}^* - \mathbf{p})\} \quad (4.11)$$

where $\mathbf{m}^*$ denotes the candidate MVs resulted from watermarking $\mathbf{d}_i$. In fact, Equation 4.11 can be seen as a motion estimation process constrained by the watermarking rules. The number of bits and distortion introduced by watermarks have been taken into account.

The watermark extraction process is straightforward. Only entropy decoding is required to read out the MVDs from the compressed bitstream. If the MVD $|\mathbf{d}_i|$ is greater than the threshold τ , two watermark bits are extracted by examining the LSBs of (d_x, d_y) according to Table 4.4. Clearly, our water-

Figure 4.12: The number of bits consumed to encode d_x , the z-component of the motion vector difference. The y-component of the MVD has the same value.

marking method leads to a very fast implementation and no additional memory is required. In contrast, traditional MV watermarking methods, such as [35], embed the watermark bits into MVs rather than MVDs. Therefore, before watermark extraction, the MVs need to be reconstructed by adding the MVDs to their corresponding predicted MVs. This operation requires to remember all previously decoded MVs in order to compute the prediction associated with current MVD. Obviously, the memory consumption and computational complexity is much higher than our method.

4.4.2 Threshold Decision and Analysis

Researchers have observed that embedding MVs which have large magnitudes can reduce the distortion resulted from watermarking [34]. So we are inspired to introduce a threshold τ in Equation 4.10. The reason is based on the following observations. First of all, watermarking MVD with a larger magnitude will introduce little visual quality degradation. Generally, the MV can be accurately

predicted if the current block and its neighboring blocks belong to the same fast moving object or the same background with slow motions. In that case, MVDs are small. However, if these blocks are within different motion segments, it may lead to large MV prediction errors, i.e. large MVDs. In that case, high spatial activities such as object boundaries and motion blurring [101] will mask the artifacts caused by watermarking MVDs.

More importantly, an appropriate τ even helps in reducing the number of bits required to encoding the watermarked MVD. This reduction is more appreciable in low bit videos, in which MVD has higher priority in coding bits budget than the coding of the prediction errors. To have an appreciation regarding how encoding MVD consumes the bits, Figure 4.12 demonstrates an example yielded by the CAVLC entropy coding table [21], which can be used to compute the value (in terms of *bits*) of the rate term $R(\mathbf{d}) = R(d_x) + R(d_y)$ in Equation 4.11. We can see when $|\mathbf{d}_i| \leq \frac{1}{2}$ pixel, whatever changes to $\mathbf{d}$ will consume 2 or more bits. In contrast, there are some “flat” regions in the curve, which means changing d_x/d_y a little within the flat regions will not consume more bits after watermarking. Therefore, to avoid watermarking the MVDs near (0,0) can reduce the bit consumption. On the other hand, MVDs have very high concentration on (0,0), for example around 85% of MVDs in *Container* are within $(-\frac{1}{2}, \frac{1}{2})$ pixel. It means that if τ is set too high, fewer MVDs will be eligible for watermarking. Therefore, the threshold τ must be adjusted according to the capacity requirement as well as the type of video sequences. Generally, τ can be high in video sequences with large and complex motions, but reduced in video sequences with slow and gentle motions.

Figure 4.13: The number of bits that can be embedded in motion vectors when $\tau \in \{0, 0.5, 1, 2\}$ (in pixels). Both sequences are encoded at 512 kbit/s.

4.4.3 Experiments and Results

Test Conditions

Simulations have been conducted on the sequences *Foreman* and *Silent* to evaluate the performance of the MVD watermarking technique. Both sequences are encoded at a bit-rate of 512 kbit/s. fullsearch motion estimation with a range of ± 32 pixels is used by the JM encoders as well as the Lagrangian optimization to find the optimal MVs. All the other conditions are the same as those mentioned in Section 4.3.4.

Experimental Results

Figure 4.13 shows the number of bits that can be embedded per frame under different threshold values. Since I-frame contains no MVs, they are omitted here for clarity. Since two bits can be embedded in the MV of a block, the total number of bits that can be embedded is twice the INTER-coded blocks. In H.264/AVC, regions of complex motion activities generally are encoded with

Sequence	Target Bitrate	Achieved Bitrate	PNSR of Unmarked	PSNR of Marked		
				$\tau = 2$	$\tau = 1$	$\tau = 0$
Foreman	512	534.18	40.34	40.29	40.28	40.20
Stefan	512	549.02	34.87	34.84	34.84	34.81
Silent	512	543.42	43.26	43.23	43.22	43.13
Average	512	542.21	39.49	39.45	39.45	39.38

Table 4.5: PSNR of the unmarked and marked test sequences. For marked sequences, 800 bits are embedded with different thresholds τ . The target and achieved bit-rates are given in kbit/s, and the PSNR value are given in dB.

smaller block size. The number of bits that can be embedded therefore depends on the motion activities and the selected modes for the macroblocks. As a result, under the same threshold, more bits can be embedded in *Foreman* than in *Silent*, since *Foreman* contains far more dynamic motions than *Silent*. Figure 4.13 also demonstrates that increasing the threshold will result that fewer bits can be embedded. However, as seen in Figure 4.13(b), even for *Silent*, which features few motions and large still background, around 1000 bits can be embedded in each frame when the threshold $\tau = 1$ pixel. It is enough for many watermarking applications. Obviously, high watermarking capacity has been achieved by our technique.

In order to examine the objective quality of the watermarked videos, we embedded 800 bits to every P-frame of *Foreman* and *Silent*. The results are shown in Table 4.5 on the aspects of PSNR and bit-rates under different thresholds. On average, watermarking with higher τ value will achieve better PSNR. It is observed that when $\tau = 2$, there is only a 0.04 dB drop in PSNR compared with the unmarked sequences. In contrast, when $\tau = 0$ (i.e., no threshold is employed at all) the PSNR drops by 0.1 dB. In addition, the PSNR of the videos watermarked with $\tau = 1$ and that with $\tau = 2$ are almost the same. The R-D curves in Figure 4.14 further validate the above observations. Especially

Figure 4.14: Rate-distortion (R-D) curves for the H.264/AVC coded streams without watermark and with watermark at different threshold values for the sequences *Foreman* and *Silent*. The sequences are compressed with constant QPs.

in Figure 4.14(b), the R-D curve for the video marked with $\tau = 1$ and that for the unmarked videos almost overlap. These results suggest that increasing the threshold τ does not necessarily bring proportional improvement in quality. The threshold $\tau = 1$ is enough for most cases.

In order to compare the visual quality of the watermarked sequences, we present the Watermarked 60th frame of *Foreman* and *Silent* in Figure 4.15 and 4.16, respectively, as well as their respective original frames.

However, the watermark embedded in MVDs shares one characteristic of traditional MV watermarking techniques — it is fragile and easy to remove. This is demonstrated by an experiment which simply re-compresses the video with the same quantization parameters.

Figure 4.15: The 60th frame of *Foreman* (512 kbit/s, CIF-size) without watermarking (a) and with watermarking (b). (a) PSNR=39.79 dB; (b) PSNR=39.76 dB.

Figure 4.16: The 60th frame of *Silent* (512 kbit/s, CIF-size) without watermarking (a) and with watermarking (b). (a) PSNR=42.72 dB; (b) PSNR=42.72 dB.

4.5 A Hybrid Scheme for Semi-fragile Watermarking

Semi-fragile watermarking is fairly new compared with its fragile and robust counterparts. It can be fragile to any malicious attacks, but robust to some extent of content preserving modifications, such as transcoding [100]. It has been found useful in various applications. Campisi *et al.* [94][95] propose semi-fragile watermark for blind evaluation of the *quality of services (QoS)* of video streaming. For MPEG-2 video authentication, Yin and Yu [92], and Park *et al.* [93] construct semi-fragile watermarking systems by combining a robust and a fragile watermarking technique. In [92], Yin embeds the fragile watermark by modifying the LSBs of the DCT coefficients, while the robust watermark is embedded by the DCT-based spread spectrum method. Moreover, Yin's method depends on the GOP structure and only I-frames are used for watermarking. However, I-frames appear at large intervals in typical low bit-rate videos since its compression rate is the lowest in all frame types. Yin's method is therefore more suitable for high bit-rate applications other than low bit-rate ones. Moreover, to the best of our knowledge, few semi-fragile watermarking scheme for low bit-rate video has been propose. It motivates our investigation.

In this section, we propose a novel semi-fragile watermarking strategy by combining the previous proposed DCT-domain watermarking method and the MVD watermarking method, as illustrated in Figure 4.17. The design is similar with Yin's method in spirit — the watermark comprises a robust watermark W_R and a fragile watermark W_F . For typical video authentication applications, W_F can be the frame feature, frame indexes, the time-stamps, etc, and W_R is the invariant features of the video stream. However, in our method, the fragile

Figure 4.17: Watermark generation and insertion into both the DCT coefficients and the motion vectors. W_R is the robust watermark and W_F is the fragile watermark.

watermark W_F is embedded in the MVDs of P-frames rather than in the DCT coefficients. Only a small number of bits W_R is inserted in the DCT domain as a robust watermark.

As mentioned in the preceding sections, both the DCT domain watermarking technique and the MVD watermarking technique are of low computation complexity and suitable for real-time applications. Nevertheless, each technique has its own advantages and disadvantages. DCT domain watermarking inserts a watermark by modifying DCT coefficients. By adapting the embedding strength, it can be robust against recompression, transcoding and filtering. However, this approach is restrained by its low capacity compared with the MVD watermarking technique in low bit-rate applications because much fewer nonzero DCT coefficients are available for watermarking. Large amount of watermarking will lead to either the degradation of the video quality or the decrease of the robustness. On the other hand, because even the video compressed at very low bit-rate has many nonzero MVDs, the watermarking capacity can be much higher than that of the DCT domain watermarking method. However, the wa-

termark hidden in the MVDs is fragile to any manipulations such as filtering.

Our design combines the advantages of both the DCT domain watermarking method and the MVD watermarking method. Compared with Yin's method [92], our method provides more watermarking capacity for low bit-rate video ($< 1\text{Mbit/s}$). It has been demonstrated in Section 4.3.4 the proposed DCT domain watermarking method is able to robustly embed 96 bits in a GOP at a bit-rate as low as 396 kbit/s. Moreover, the experimental results in Section 4.4.3 show that the MVD watermarking method can embed 800 bits in a CIF-size frame without introducing noticeable PSNR quality degradation. Yin's method was evaluated with bit-rate 4 Mbit/s and embed average 32 bits in I-frames for robust watermark and 1.5 bits per 8×8 block for fragile watermarking [92]. Besides the watermarking capacity, the proposed scheme also enables new functionalities. Since the frame indexes or time-stamps can be inserted in P-frames, our scheme is able to detect frame-level dropping, adding and shuffling. These kinds of attacks generally do not affect the fragile watermark existing in the frames. In contrast, the method in [92] can only detect GOP-level attacks since only I-frames contain watermark.

4.6 Conclusions

A set of watermarking techniques have been presented in this chapter for the newest H.264/AVC video coding standard. Two indispensable syntax elements of the standard — *DCT coefficients* and *motion vectors* — are exploited respectively for robust watermarking and fragile watermarking. Combining these two watermarking techniques, a novel hybrid semi-fragile watermarking technique is also presented in this chapter. Compared with traditional watermarking methods, our methods have achieved higher capacity especially in low bit-rate video

applications. They also consume little computation and memory, which make them especially suitable for real-time video applications. In addition, the optimal prediction mode/motion vector selection in H.264/AVC is supported by slightly adapting the Lagrangian optimization functions in order to determine the best tradeoff between the objective quality and the bit-rate after watermarking. The experimental results show that our DCT domain watermarking method is capable to withstand transcoding and filtering and at the same time cause almost unnoticeable quality degradation; the MVD watermarking method achieves high capacity with only negligible PSNR loss.

Appendix 4A Test Video Sequences

Six video sequences are used to test our proposed video watermarking techniques: *Foreman*, *Stefan*, *Coastguard*, *Flower Garden*, *Container Ship*, and *Silent*. Figure 4.18 shows the first frame of each sequence. All video sequences are of CIF-size, i.e. 352 x 288 pixels, and 300 frames in length.

Foreman, *Stefan*, *Coastguard* and *Flower Garden* contain either fast motions or high spatial and color details; *Container Ship*, and *Silent* are slow motion sequences with still background, which make them more difficult to achieve watermark capacity and imperceptibility.

(a) Foreman

(b) Stefan

(c) Coastguard

(d) Flower Garden

(e) Container Ship

(f) Silent

Figure 4.18: Video sequences used in the experiments.

Chapter 5

Conclusions and Recommendations

5.1 Major Contributions and Conclusions

In this thesis, new watermarking techniques have been investigated, developed and implemented for the two advanced image and video coding standards — the *JPEG-2000 image coding standard* and the *H.264/AVC video coding standard*, respectively. The contributions are concluded as follows:

- **Novel proposal of watermarking method for JPEG-2000**

JPEG-2000 is the second-generation still-image compression standard from JPEG. It is highly promising that JPEG-2000 will become a popular image coding standard in the coming future. Naturally, our research endeavor is first focusing on developing novel digital watermarking method aligned with JPEG-2000 standard.

We presented a novel blind watermarking method based on our observation of the EBCOT algorithm, which is the foundation of JPEG-2000

encoder, as well as the wavelet coefficient representation. First, we introduced a new parameter *watermarking embedding bit-rate* (WEBR) to quantitatively control the watermarking strength. Given WEBR, we found a particular bit-plane of the wavelet coefficients for watermarking. Thus, the distortion resulted by the proposed watermarking method can be estimated by EBCOT algorithm in advance even before watermarking. In contrast, in conventional watermarking methods, the watermarking strength is controlled empirically with a gain factor. As a result, the distortion could only be known after watermarking. Second, we inserted the watermark bits by taking advantage of BPCS (Bit-Plane Complexity Segmentation) steganography method in order to achieve high watermark capacity. BPCS method inserts a watermark in forms of noise-like binary patches into the regions with complex textures. Third, we also proposed a novel method to select appropriate regions for watermarking. The average value and the complexity were jointly considered, which led to a very fast but efficient implementation. Experimental results showed that the proposed watermarking method achieves high capacity while striking a good balance between robustness and invisibility.

- **Novel proposal of a set of watermarking methods for H.264/AVC**

H.264/AVC is the newest video coding standard and the state-of-the-art coding methodology achieving superior performance to other existing standards. It is desired to investigate new real-time watermarking approaches for H.264/AVC to improve the visual quality, flexibility and robustness in dealing with low bit-rate videos (≤ 1 Mbit/s).

First, a robust watermarking method had been proposed. The watermark bits were inserted on 4×4 block basis. Direct sequence spread spectrum

method was employed to generate the watermark signal and spread it over the whole frame. In order to be compliant to the H.264 JM encoder, the Lagrangian optimization algorithm was used to select the best mode for the watermarked macroblocks. The above strategy was applied to both I-frames and P-frames to achieve greater capacity. Furthermore, a sliding-window algorithm for frame accumulation was used to further improve the performance of watermark retrieval for P-frames.

Second, a novel *motion vector* (MV) watermarking method had been proposed. Traditional MV watermarking methods modify motion vectors, but given that it is the *motion vector difference* (MVD) that is the real syntax element stored in the final compressed stream, we chose to hide data in the MVDs rather than in the MVs. Consequently, the watermark could be read out without fully decoding the compressed stream, which makes it suitable for real-time applications. Furthermore, based on our observation of the motion estimation process of the H.264/AVC JM encoder, a threshold is derived so that only the MVDs whose amplitudes are greater than the threshold will be embedded. Experiment results showed that the proposed scheme only introduced very small distortions into the video. The PSNR dropped only average 0.05 dB after watermarking while maintain high watermarking capacity.

Finally, we also present a hybrid semi-fragile watermarking scheme for video authentication. By combining the robust DCT domain watermarking technique and the fragile MV watermarking technique, Our scheme outperforms other semi-fragile video watermarking schemes with higher watermarking capacity and the capability to detect frame-level attacks.

5.2 Recommendations for Future Work

o On the new trend in watermarking

While this thesis mainly deals with blind watermarking methods, which treat the cover signal as noise and do not exploit any knowledge of the cover signal, the *informed embedding watermarking* [108] represent a trend in the watermarking research field. The *Quantization Index Modulation* technique, proposed by Chen and Wornell [108] is one of the representatives. This class of techniques embeds the watermark in a cover signal through quantization; a different quantization vector is used to embed a different watermark value. It would be interesting to further our research on the ways of applying the QIM method on JPEG and MPEG encoded images and videos, respectively.

o On watermarking JPEG-2000 compressed images

The proposed method has achieved high embedding capacity and robustness within the JPEG-2000 structure by using the BPCS steganography for data hiding. However, BPCS method inserts data by adding noise-like binary blocks, which makes it vulnerable to smoothing operations, such as low-pass filtering and median filtering. It is also vulnerable to JPEG compression, in which the high-frequency DCT coefficients are regarded as insignificant and quantized with relative large quantization parameters. Error correction code (ECC) or spread-spectrum technique can be adopted to improve the robustness.

o On watermarking H.264/AVC compressed videos

For H.264/AVC video, the DCT domain watermarking is the most challenge but intriguing part. In our DCT-domain watermarking technique,

the message is spread by the spread-spectrum technique over the whole of the frame and repeatedly embedded into successive frames. Consequently, the watermark capacity is limited by the frame size and temporal redundancy is not fully exploited. It would be desirable to spread the message pseudo-randomly over a period of frames to increase the capacity and the robustness. The 3-D interleaving technique proposed by Shi *et al.* [106], [105] is a promising solution. Experimental results in [106] show that the robustness of the embedded data has been dramatically improved by interleaving the message in a three dimensional way. Further investigation on this technique by incorporating it into the H.264/AVC watermarking method should be meaningful.

Bibliography

- [1] “The movie business: romancing the disc,” *The Economist*, vol. 370, no. 8361, pp. 55-56, Feb. 7th, 2004.
- [2] F. Hartung and F. Ramme, “Digital rights management and watermarking of multimedia content for M-commerce applications,” *IEEE Commun. Mag.*, pp. 78-84, Nov. 2000.
- [3] E.A. Taub, “A TV that prints still photos”, *The New York Times*, sec. G, pp. 3, col. 3, Jan. 8th, 2004.
- [4] D. Kahn, *The Codebreaker: The Comprehensive History of Secret Communication from Ancient Times to the Internet*, Scribner, Dec. 1996.
- [5] M. Drosnin, *The Bible Code*, Touchstone Books, Apr. 1998.
- [6] S.D. Zhang, D.Y. Pang, R.S. Zhen, et al. *General History of Chinese Printing (Zhonguo yin shua tong shi)*, [On-line]. Available in Chinese: <http://www.cgan.net/book/books/print/g-history/>, 1999.
- [7] J.R.H. Martin and M. Kutter, “Information retrieval in digital watermarking,” *IEEE Commun. Mag.*, pp. 110-116, Aug. 2001.

-
- [8] Z.C. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 2, pp. 11-912 - II-915, 2003.
 - [9] M.D. Swanson, B. Zhu, and A.H. Tewfik, "Data hiding for video-in-video," in *Proc. IEEE Int Conf. Image Processing* vol. 2, pp. 676 - 679, 1997.
 - [10] F. Hartung and M. Kutter, "Multimedia watermarking techniques," *Proc. IEEE*, vol. 87, no. 7, pp.1079 - 1107, Jul. 1999.
 - [11] F.A.P. Petitcolas, R.J. Anderson, and M.G. Kuhn, "Information hiding—A survey," *Proc. IEEE*, vol. 87, no. 7, pp. 1062-1078, Jul. 1999.
 - [12] G. Langelaar, I. Setyawan, and R. Lagendijk, "Watermarking digital image and video data: a state-of-the-art overview," *IEEE Signal Processing Mag.*, vol. 17, pp. 20-46, Sept. 2000.
 - [13] C. Podilchuk and E. Delp, "Digital watermarking: algorithms and applications," *IEEE Signal Processing Mag.*, vol. 17, pp. 33-46, Sept. 2001.
 - [14] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing*, vol. 66, no. 3, pp. 283-301, May 1998.
 - [15] G. Doërr and J-L. Dugelay, "A guided tour of video watermarking," *Signal Processing: Image Commun.*, vol. 18, pp. 263-282, 2003.
 - [16] *Video Codec for Audio-Visual Services at 64 kbit/s*, ITU-T Rec. H.261, 1993.
 - [17] *Video Coding for Low Bit-Rate Communication*, ITU-T Rec. H.263; Version 1: Nov. 1995; Version 2: Jan. 1998; Version 3: Nov. 2000.

-
- [18] *Information Technology— Coding of Moving Pictures and Associated Audio for Digital Storage Media at up to 1.5 Mbit/s: Video*, ISO/IEC 11172-2 (MPEG-1 Video), 1993.
 - [19] *Information Technology— Generic Coding of Moving Pictures and Associated Audio Information: Video*, ISO/IEC 13818-2 (MPEG-2 Video), ITU-T Rec. H.262, 1995; 2nd ed., ISO/IEC 13818-2:2000/2000.
 - [20] *Information Technology— Generic Coding of Audio- Visual Objects*, ISO/IEC JTC 1 14496-2, (MPEG-4 Visual), Version 1: Apr. 1999, Amendment 1 (Version 2): Feb. 2000, Amendment 4 (streaming profile): Jan. 2001.
 - [21] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC),” Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, doc. JVT-G050, Mar. 2003.
 - [22] B.G. Haskell et al., “Image and video coding: Emerging standards and beyond,” *IEEE Trans. Circuits Systems for Video Technology*, Nov. 1998, 8(7):814-37.
 - [23] Y. Wang, J. Sstermann and Y.Q. Zhang, *Video Processing and Communications*, Prentice Hall, USA, 2001.
 - [24] ITU-T/SG 16/Q.6 (VCEG), “Advanced video coding: Background and Justification,” <http://www.itu.int/itu-t/studygroups/com16/sg16-q6.html>.

-
- [25] G.J. Sullivan, T. Wiegand and T. Stockhammer, "Using the draft H.26L video coding standard for mobile applications," in *Proc. IEEE Int. Conf. Image Processing*, Thessaloniki, Greece, Sep. 2001, invited paper.
 - [26] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand and G.J. Sullivan, "Performance comparison of video coding standards using Lagrangian coder control," *Proc. IEEE Int. Conf. Image Processing*, Rochester, NY, USA, Sept. 2002.
 - [27] T. Wiegand, B. Girod. *Multi-Frame Motion-Compensation Prediction For Video Transmission*, Kluwer Academic Publishers, USA, 2001.
 - [28] T. Wiegand, "Rate-distortion optimized mode selection for very low bit rate video coding and the emerging H.263 standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 6, pp. 182-190, Apr. 1996.
 - [29] G.J. Sullivan, "Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-Scan Source Material," document VCEG-N81, Sept. 2001.
 - [30] G.J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, Nov. 1998.
 - [31] D. Simitopoulos, S.A. Tsiftaris, N.V. Boulgouris, and M.G. Strintzis, "Compressed-domain video watermarking of MPEG streams," in *Proc. IEEE Int'l Conf. Multimedia and Expo*, vol.1, pp. 569 -572, Aug. 2002.
 - [32] F. Jordan, M. Kutter, and T. Ebrahimi, "Proposal of a watermarking technique for hiding/retrieving Data in compressed and decompressed Video," ISO/IEC JTC1/SC29/WG11/MPEG97/M2281, 1997.

-
- [33] J. Song and K.J.R. Liu, "A data embedding scheme for H.263 compatible video coding," in *Proc. IEEE Int. Symp. Circuits and Systems*, vol. 4, pp. 390 -393, June 1999.
 - [34] J. Zhang, J. Li, and L. Zhang, "Video watermark technique in motion vector," in *XIV Brazilian Symp. on Computer Graphics and Image Processing*, pp. 179-182, Oct. 2001.
 - [35] J. Song and K.J.R. Liu, "A data embedded video coding scheme for error-prone channels," *IEEE Trans. Multimedia*, vol. 3, no. 4, pp. 415-423, Dec. 2001.
 - [36] Y. Bodo, N. Laurent, and J.-L. Dugelay, "[Watermarking] video, hierarchical embedding in motion vectors," in *Proc. IEEE Int. Conf. Image Processing*, no. 2, pp. 739-742, Sept. 2003.
 - [37] C.-S. Lu, H.-Y.M. Liao, and C.-J. Sze, "Combined watermarking for image authentication and protection," in *Proc. IEEE Int'l Conf. Multimedia and Expo*, vol. 3, pp. 1415 -1418, Aug. 2000.
 - [38] T. Wiegang, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560 -576, Jul. 2003.
 - [39] T. Wiegang, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and compression of video video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688 -703, Jul. 2003.
 - [40] M. Wien, "Variable block-size transforms for H.264/AVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 604 -613, July 2003.

-
- [41] A. M. Alattar, E. T. Lin, and M. U. Celik, "Digital watermarking of low bit-rate advanced simple profile MPEG-4 compressed video," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 787-800, Aug. 2003.
 - [42] C.-Y. Lin and S.-F. Chang, "Issues and solutions for authenticating MPEG video," *SPIE*, vol. 3657, 1999.
 - [43] K. Tanaka, Y. Nakamura, and K. Matsui, "Embedding secret information into dithered multi-level image," in *Proc. 1990 IEEE Military Commun. Conf.*, pp. 216-220, 1990.
 - [44] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq, "Invisibility and application functionalities in perceptual watermarking—an overview," *Proc. IEEE*, vol. 90, no. 1, pp. 64-77, Jan. 2002.
 - [45] A. Tirkel, G. Rankin, R. van Schyndel, W. Ho, N. Mee, and C. Osborne, "Electronic water mark," in *Proc. DICTA 1993*, pp. 666-672, Dec. 1993.
 - [46] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. Int. Conf. Image Processing*, vol. 2, pp. 86-89, 1994.
 - [47] J. Fridrich and M. Goljan, "Protection of digital images using self embedding," in *Proc. Symp. Content Security and Data Hiding in Digital Media*, New Jersey Institute of Technology, Mar. 1999.
 - [48] W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "Techniques for data hiding," *IBM Systems Journal*, vol. 35, no. 3 & 4, pp. 313-336, 1996.
 - [49] K. Matsui and K. Tanaka, "Video-steganography: how to secretly embed a signature in a picture," in *Proc. IMA Intellectual Property Project*, vol. 1, pp. 187-206, Jan. 1994.

-
- [50] M. Kutter, F. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," in *Proc. Electronic Imaging 1997 (EI 97)*, Feb. 1997.
- [51] B. Macq, J.-F. Delaigle, and C. De Vleeschouwer, "Digital watermarking," *SPIE Proc. 2659: Optical Security and Counterfeit Deterrence Techniques*, pp. 99-110, Mar. 1996.
- [52] J. Tian, "Reversible data embedding using a difference expansion ICompress Technologies," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 8, pp. 890- 896, Aug. 2003.
- [53] C. De Vleeschouwer, J.-F. Delaigle, and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Tran. Multimedia*, vol. 5, pp. 97-105, Mar. 2003.
- [54] I.J. Cox, J. Kilian, F.T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, vol. 6, pp. 1673-1687, Dec. 1997.
- [55] E. Koch and J. Zhao, "Toward robust and hidden image copyright labeling," in *Proc. Workshop Nonlinear Signal and Image Processing*, pp. 452-455, Jun. 1995.
- [56] J. Zhao and E. Koch, "Embedding robust labels into images for copyright protection," in *Proc. Int. Congr. Intellectual Property Rights for Specialized Information, Knowledge and New Technologies*, pp. 242-251, Aug. 1995.
- [57] C.-T. Hsu and J.-L. Wu, "Hidden signatures in images," in *Proc. IEEE Int. Conf. Image Processing*, vol. 3, pp. 223-226, Sept. 1996.

- [58] C.S. Lu, S.K. Huang, C.J. Sze, and H.Y.M. Liao, "Cocktail watermarking for digital image protection," *IEEE Trans. Multimedia*, vol. 2, pp. 209-224, Dec. 2000.
- [59] J. Huang, Y.Q. Shi, and Y. Shi, "Embedding image watermarks in DC components," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 10, pp. 974-979, Sept. 2000.
- [60] B. Sklar, *Digital Communications: Fundamentals and Applications*, 2nd Ed., NJ: Prentice-Hall, 2001.
- [61] C. Busch, W. Funk, and S. Wolthusen, "Digital watermarking: From concepts to real-time video applications," *IEEE Comput. Graphics Applicat.*, pp. 25-35, Jan. 1999.
- [62] T. Kalker, G. Depovere, J. Haitsma, and M. Maes, "A video watermarking system for broadcast monitoring," in *Proc. SPIE, Security and Watermarking of Multimedia Content*, San Jose, CA, vol. 3657, pp. 103-112, 1999.
- [63] M. Maes, T. Kalker, J.-P. Linnartz, J. Talstra, G. Depovere, and J. Haitsma, "Digital watermarking for DVD video copy protection," *IEEE Signal Processing Mag.*, pp. 47-57, Sept. 2000.
- [64] A.M. Alattar, E.T. Lin, and M.U. Celik, "Watermarking low bit-rate advanced simple profile MPEG-4 bitstreams," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol. 3, pp. 513-16, Apr. 2003.
- [65] G.C. Langelaar, R.L. Lagendijk, and J. Biemond, "Real-time labeling of MPEG-2 compressed video," *J. Vis. Commun. Image Represent.*, vol. 9, no. 4, pp. 256-270, 1998.

-
- [66] G.C. Langelaar and R.L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Image Processing*, vol. 10, pp. 148-158, Jan. 2001.
 - [67] J.-P.M.G. Linnartz and J.C. Talstra, "MPEG PTYMarks: Cheap Detection of embedded Copyright Data in DVD-Video," *Computer Security — ESORICS'98: Proc. 5th Euro. Symp. Res. In Comput. Security, Lecture Notes in Computer Science*, Springer, vol. 1485, pp. 221-240, 1998.
 - [68] *Information Technology — JPEG — Digital Compression and Coding of Continuous-Tone Still Image - Part 1: Requirements and Guidelines*, ISO/IEC 10918-1 and ITU-T Rec. T.81, 1994.
 - [69] *Information Technology — JPEG 2000 — Image Coding System - Part 1: Core Coding System*, ISO/IEC 15444-1, 2000.
 - [70] *Information Technology — JPEG 2000 — Image Coding System - Part 8: Secure JPEG 2000 (JPSEC Committee Draft - Version 1.0)*, ISO/IEC CD 15444-8, ISO/IEC JTC1/SC29/WG1 N3297, Apr. 2004.
 - [71] D. Taubman, "High performance scalable image compression with EBCOT," *IEEE Trans. Image Processing*, vol. 9, no. 7, pp. 1158-1170, July 2000.
 - [72] D.S. Taubman, M.W. Marcellin, "JPEG2000: standard for interactive imaging," *Proc. IEEE*, vol. 90, no. 8, pp. 1336-1357, Aug. 2002.
 - [73] M. D. Adams, "The JPEG-2000 still image compression standard Tech. Rep. distributed with the JasPer JPEG-2000 software," Tech. Rep. ISO/IEC JTC1/SC29/WG1/N2412 [Online]. Available: <http://www.ece.ubc.ca/~mdadams/jasper>, Sept. 2001.

-
- [74] EPFL, Ericsson, and Canon Research France, "Java implementation of the JPEG 2000 reference software," [Online]. Available: <http://jpeg2000.epfl.ch>.
 - [75] K. Wallace, "The JPEG still picture compression standard," *IEEE Trans. Consumer Electron.*, vol. 38, no. 1, pp. xviii-xxxiv, Feb. 1992.
 - [76] X.G. Xia, C.G. Boncelet, and G.R. Arce, "A multisolution watermark for digital images," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, vol. 1, pp. 548-551, 1997.
 - [77] H.-J. Wang, P.-C. Su, and C.-C.J. Kuo, "Wavelet based blind watermark retrieval technique," in *SPIE Photonics East Symposium on Voice, Video, and Data Communications*, 1998.
 - [78] H.-J. Wang and C.-C.J. Kuo, "Image protection via watermarking on perceptually significant wavelet coefficients," in *Proc. IEEE Int. Conf. Multimedia Signal Processing*, pp. 279-284, 1998.
 - [79] P.-C. Su and C.-C.J. Kuo, "Steganography in JPEG2000 compressed images," *IEEE Trans. Consumer Electronics*, vol. 49, no. 4, pp. 824-832, Nov. 2003.
 - [80] P.-C. Su, H.-J. M. Wang and C.-C.J. Kuo, "An integrated approach to image watermarking and JPEG-2000 Compression", *Journal of VLSI Signal Processing*, no. 27, pp.35-53, 2001.
 - [81] R. Grosbois, T. Ebrahimi, "Watermarking in the JPEG 2000 domain," *IEEE Fourth Workshop on Multimedia Signal Processing*, pp. 339-344, Oct. 2001.

-
- [82] T.-S. Chen, J. Chen, J.-G. Chen, "A simple and efficient watermarking technique based on JPEG2000 codec," *Fifth Int. Symp. Multimedia Software Engineering, 2003. Proceedings*, pp. 80-87, 2003.
 - [83] M. Rabbani and R. Joshi, "An overview of the JPEG2000 still image compression standard," *Signal Processing: Image Communication*, vol. 17, pp. 3-48, Jan. 2002.
 - [84] A. Skodras, C. Christopoulos, and T. Ebrahimi, "The JPEG 2000 still image compression standard," *IEEE Signal Processing Mag.*, vol. 18, pp. 36-58, Sept. 2001.
 - [85] Z.S. Zhang, G. Qiu, Q.B. Sun, X. Lin, Z.C. Ni, and Y.-Q. Shi, "An unified authentication framework for JPEG2000," ISO/IEC JTC1/SC29/WG1/N3211, Mar. 2004.
 - [86] Q.B. Sun, S.-F. Chang, M. Kurato, and M. Suto, "A quantitative semi-fragile JPEG2000 image authentication system," in *Proc. Int. Conf. Image Processing*, vol. 2, pp. 921-924, Sept. 2002.
 - [87] E. Kawaguchi and R.O. Eason, Principle and applications of BPCS-steganography, in *Proc. SPIE*, vol. 3528, pp. 464-473, 1998.
 - [88] H. Noda, J. Spaulding, M.N. Shirazi, and E. Kawaguchi, "Application of bit-plane decomposition steganography to JPEG2000 encoded images," *IEEE Signal Processing Lett.*, vol. 9, no. 12, pp. 410-413, Dec. 2002.
 - [89] H. Noda, J. Spaulding, M. N. Shirazi, M. Niimi, and E. Kawaguchi, "Bit-plane decomposition steganography combined with JPEG2000 compression," in *5th Workshop on Information Hiding*, pp. 295-309, Oct. 2003.

-
- [90] J. Spaulding, H. Noda, M.N. Shirazi, and E. Kawaguchi, "BPCS steganography using EZW lossy compressed images," *Pattern Recognition Letters*, no.13, Nov. 2002.
 - [91] F.A.P. Petitcolas, StirMark Benchmark. [Online] Available: <http://www.petitcolas.net/fabien/watermarking/stirmark/>.
 - [92] P. Yin and H.H. Yu, "Semi-fragile watermarking system for MPEG video authentication," in *Proc. IEEE Int'l Conf. Acoustics, Speech, and Signal Processing*, vol.4, pp. 3461-3464, 2002.
 - [93] J. Y. Park, J. H. Lim, G. S. Kim, and C. S. Won, "Invertible semi-fragile watermarking algorithm distinguishing MPEG-2 compression from malicious manipulation," in *Int'l Conf. on Consumer Electronics*, Digest of Technical Papers, pp. 18-19, Jun. 2002.
 - [94] P. Campisi, G. Giunta, and A. Neri, "Object-based quality of service assessment using semi-fragile tracing watermarking in MPEG-4 video cellular services," *IEEE Int'l Conf. Image Processing*, vol. 2, pp. 881-884, Sept. 2002.
 - [95] P. Campisi, M. Carli, G. Giunta, and A. Neri, "Blind quality assessment system for multimedia communications using tracing watermarking," *IEEE Trans. Signal Processing*, vol. 51, no. 4, pp. 996- 1002, Apr. 2003.
 - [96] J. Zhang and A.T.S. Ho, "An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform (IntDCT)," in *IEEE Joint Conf. of 4th Int. Conf. Info., Commun. and Signal Processing and 4th Pacific-Rim Conf. Multimedia*, Dec. 2003.

-
- [97] F.A.P. Petitcolas, R.J. Anderson, and M. G. Kuhn. "Attacks on copyright marking systems," in *Information Hiding Second International Workshop*, David Aucsmith (Ed), Proceedings LNCS 1525, Springer-Verlag, pp. 219-239, Apr. 1998.
 - [98] F.A.P. Petitcolas, "Watermarking schemes evaluation," *IEEE Signal Processing*, vol. 17, no. 5, pp. 58C64, Sept. 2000.
 - [99] J. Lacy, N. Rump, and P. Kudumakis, "MPEG-4 intellectual property management & protection (IPMP) overview and applications," ISO/IEC/JTC1/SC29/WG11/N2614, Dec. 1998.
 - [100] A. Vetro, C. Christopoulos, and H. Sun, "Video transcoding architectures and techniques: an overview," *IEEE Signal Processing*, vol. 20, no. 2, pp.18-29, Mar. 2003.
 - [101] P. Marziliano, F. Dufaux, S. Winkler, and T. Ebrahimi, "A no-reference perceptual blur metric," in *Proc. Int. Conf. Image Processing*, vol. 3, pp. 57-60, Jun. 2002.
 - [102] G. Qiu, P. Marziliano, Anthony T.S. Ho, D.J. He, and Q.B. Sun, "A hybrid watermarking scheme for H.264/AVC video," in *the 17th International Conference on Pattern Recognition*, Cambridge, UK, Aug. 2004.
 - [103] ITU-T/SG 16/Q.6 (VCEG) and ISO/IEC JTC 1/SC 29/WG11 (MPEG), "Terms of Reference for Joint Video Team (JVT) Activities." Annex 1, VCEG-054, Pattaya, Thailand, 5 Dec., 2001.
 - [104] K. Huhning, E.d., H.264/AVC Joint Model 7.6 (JM-7.6) Reference Software. [Online]. Available: ftp://imtc.org/jvt-exports/reference_software/.

-
- [105] Y.Q. Shi, X.M. Zhang, Z.C. Ni, and N. Ansari, "Interleaving for combating bursts of errors," *IEEE Circuits and Systems Mag.*, vol. 4, no. 1, pp. 29 - 42, 2004.
- [106] Y.Q. Shi, Z.C. Ni, N. Ansari, and J.W. Huang, "2-D and 3-D successive packing interleaving techniques and their applications to image and video data hiding," in *Proc. IEEE Int'l Symp. Circuits and Systems*, pp. 924 - 927, vol.2, May 2003.
- [107] I.J. Cox, M.L. Miller, and A.L. McKellips, "Watermarking as communications with side information," *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1127-1141, July 1999.
- [108] B. Chen and G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Trans. Info. Theory*, vol. 47, no. 4, May 2001.
- [109] M. Barni, "What is the future for watermarking? (Part II)," *IEEE Signal Processing Mag.*, pp. 53-55, Nov. 2003.
- [110] J.J. Eggers and B. Girod, "Informed watermarking," Kluwer Academic Publishers, 2002.
- [111] M. Barni and F. Bartolini, "Watermarking system engineering: enabling digital assets security and other applications," Marcel Dekker, Feb. 2004.

Publications

[1] G. Qiu, P. Marziliano, Anthony T.S. Ho, D.J. He, and Q.B. Sun, “A hybrid watermarking scheme for H.264/AVC video,” in *the 17th International Conference on Pattern Recognition*, Cambridge, UK, Aug. 2004.

[2] Z. S. Zhang, G. Qiu, Q. B. Sun, X. Lin, Z. C. Ni, and Y. Q. Shi, “A unified authentication system for JPEG2000,” in *2004 IEEE International Conference on Multimedia and Expo*, 2004.