Digital video watermarking of H.264/AVC

Zhang, Jing

2008

Zhang, J. (2008). Digital video watermarking of H.264/AVC. Doctoral thesis, Nanyang Technological University, Singapore.

<https://hdl.handle.net/10356/41405>

<https://doi.org/10.32657/10356/41405>

Digital Video Watermarking of H.264/AVC

Zhang Jing

School of Electrical and Electronic Engineering

A thesis submitted to the Nanyang Technological University
in fulfilment of the requirement for the degree of
Doctor of Philosophy

2008

Statement of Originality

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

Feb 15 2028

Date

张静 Zhang Jing

Zhang Jing

*To all in my family,
for their encouragement and love.*

Acknowledgements

I would like to take this opportunity to express my appreciation and gratitude to several individuals, without whom I would not have enjoyed my study and completed this thesis. First of all, I gratefully acknowledge my previous supervisor, Professor Ho Tung Shuen, Anthony, who had given me continuous supervision, guidance and encouragement in the past three years. I would like to thank my current supervisor Assistant Professor Pina Marziliano for her inspiring encouragement, insightful suggestions and valuable advice, which also guided me throughout my research work. I would also like to express my appreciation to Assistant Professor Guan Yong Liang, Assistant Professor Farook Sattar, Associate Professor Chua Hock Chuan, Associate Professor Anamitra Makur and Associate Professor Foo Say Wei for their invaluable support and suggestions on my research.

For plentiful discussions, cordial collaboration, and exchanges of ideas throughout the study period, I would like to express my gratitude to Mr. Qiu Gang, Mr. Chua Chin Eng and Mr. Koh Kai Meng. Special thanks are also presented to Mr. Shu Feng, Mr. Shen Jun, Mr. Niladri Bihari Puhan and Ms. Zhu Xunzhan for their kind assistance. Credits should also be given to the technicians in the Workstation Resource Lab Mr. Chu Chun Chung and Ms. Teo Boon Ping for their logistic support.

Abstract

A digital video watermark is information stored invisibly in the video content to which it belongs. This is done by slightly changing the video content according to a secret pattern. While this is imperceptible during normal viewing, a decoder can detect and retrieve the watermark by using the same secret pattern. Once applied, the watermark cannot be deleted or changed. It is inextricably bound to the video content, surviving signal processing steps or data compression, but also geometric distortions. Watermarking techniques enable a variety of content protection applications. For example, the developed robust video watermarking scheme is designed to work with the upcoming digital-cinema format, and can help content owners and distributors by tracing the origin of illegal copies of movies made in the cinema, which is an increasingly occurring practice.

Digital watermarking has first been extensively studied for still images. In this PhD thesis, a novel image-in-image watermarking algorithm is first studied for copyright protection. This algorithm is aimed to test the watermarking performance of the integer DCT transform used in the H.264/AVC standard, which is an approximation to the traditional DCT. Edge detection is used to extract the significant image feature for adaptive embedding and some effective post-processing approaches are utilized to increase the watermark robustness against common signal processing. The novel image-in-image watermarking scheme shows good performance and is benchmarked with StirMark 4.0.

Digital watermarking has recently been extended from still images to video content and the large economic stakes strongly motivates further research in this area. H.264/AVC is the latest video coding standard and the state-of-the-art methodology achieving superior performance to other existing standard. Based on the valid experimental results of the developed image watermarking algorithm, an efficient and robust video watermarking algorithm for H.264/AVC is proposed. Two different types of watermarks can be successfully embedded in the video content and extracted from the processed content. The advantage of embedding the watermark into polarities is the considerable improvement on robustness under a variety of accidental and intentional manipulations. We also propose a novel and efficient scheme of fragile video authentication for the standard H.264/AVC. The motion techniques of H.264/AVC are exhaustively investigated for the authentication strategy, such as the tree-structured motion compensation, motion estimation and Lagrangian optimization. Authentication information was embedded strictly based on these motion technologies so that if watermarked video content undergoes any spatial and temporal attacks, the scheme can detect tampering by the sensitive mode change.

Contents

Acknowledgements	i
Abstract	ii
List of Figures	xi
List of Tables	xv
List of Notations	xvii
List of Acronyms	xxiv
1 INTRODUCTION	1
1.1 Background	1
1.2 Motivation	3
1.3 Objectives	6
1.4 Major Contributions	6
1.5 Thesis Outline	7
2 WATERMARKING BASICS	11
2.1 Basic Structure of Digital Watermarking	11
2.2 Watermarking Properties	12
2.2.1 Fidelity	12
2.2.2 Embedding Capacity	13

2.2.3	Informed or Blind Detection	13
2.2.4	Robustness	14
2.2.5	Security	14
2.2.6	Complexity	15
2.3	Classification of Watermarking Techniques	15
2.4	Benchmarking for Watermarking Systems	16
2.5	Applications of Watermarking	19
2.5.1	Copyright Protection	19
2.5.2	Content Authentication	19
2.5.3	Fingerprinting	20
2.5.4	Copy Control	20
2.5.5	Broadcast Monitoring	20
3	REVIEW OF WATERMARKING TECHNIQUES	21
3.1	Image Watermarking Techniques	22
3.1.1	Spatial Domain	22
3.1.2	Transform Domain	24
3.2	Video Watermarking Techniques	27
3.2.1	Still Images Techniques	28
3.2.2	Compressed Domain Techniques	29
3.2.3	Bit Stream Techniques	31
3.3	Techniques in Authentication Applications	31
4	THE H.264/AVC STANDARD	35
4.1	Introduction	35
4.2	The H.264 Codec	36
4.3	H.264 Structure	39
4.3.1	Profiles and Levels	39
4.3.2	Video Format	40

4.3.3	Video Coding Layer and Network Abstract Layer	41
4.3.4	Reference Pictures	41
4.3.5	Slices	42
4.4	Baseline Profile	42
4.4.1	Intra Coding	42
4.4.2	Inter Coding	43
4.4.2.1	Motion Compensation in P-slices	43
4.4.2.2	Motion compensation in B-slices	46
4.4.3	In-Loop Deblocking Filter	48
4.4.4	Transform, Scaling and Quantization	48
4.4.5	Entropy Coding	49
4.4.6	Mode Decision with Lagrangian Optimization	50
4.5	Applications	51
5	PROPOSED ROBUST WATERMARKING OF DIGITAL IM- AGES	53
5.1	Introduction	53
5.2	The Approximate Integer DCT	54
5.3	Proposed Image-in-Image Method	57
5.3.1	Embedding Approach	58
5.3.1.1	Watermark Pattern	58
5.3.1.2	Cover Image	58
5.3.2	Retrieval Approach	64
5.4	Post-processing Techniques	66
5.5	Experiments and Results	67
5.5.1	Performance Measurements	67
5.5.2	Fidelity after Watermarking	69
5.5.3	Results of Robustness Tests	70

5.5.4	Performance Comparison	73
5.6	Summary	81
6	PROPOSED ROBUST WATERMARKING OF H.264/AVC STANDARD	83
6.1	Introduction	83
6.2	Methodology of Binary Sequence Watermark	84
6.2.1	Watermark Formation	84
6.2.2	Watermark Embedding	85
6.2.3	Watermark Retrieval	88
6.2.4	Watermark Embedding and Retrieval Approaches within Multiple Frames	89
6.3	Theoretical Analysis	90
6.3.1	Robust Embedding	90
6.3.2	Effective Retrieval	91
6.3.3	Optimized Rate-Distortion Control	92
6.4	Methodology of Grayscale Pattern Watermark	93
6.4.1	Classification	96
6.4.2	Watermark Pre-processing	96
6.4.2.1	4×4 Integer DCT and Zigzag-scanning	97
6.4.2.2	Normalization and Frequency Mask	98
6.4.2.3	Transformation	98
6.4.2.4	Level Reduction	100
6.4.3	Watermark Reconstruction	100
6.4.4	Watermark Embedding and Retrieval	102
6.5	Experiments and Results of Binary Sequence Algorithm	103
6.5.1	Test Conditions	103
6.5.2	Results of Robustness Tests	104

6.5.3	Algorithm Comparison	106
6.5.4	Summary	109
6.6	Experiments and Results of Grayscale Watermark Algorithm . .	110
6.6.1	Fidelity after Watermarking	110
6.6.2	Results of Robustness Tests	113
6.6.3	Summary	116
7	PROPOSED HARD AUTHENTICATION OF H.264/AVC STANDARD	117
7.1	Introduction	117
7.2	Investigation of Best Mode	118
7.2.1	Tree-structured Motion Compensation	118
7.2.2	Partition Mode Change	121
7.3	Proposed Authentication Method	123
7.3.1	Embedding	123
7.3.2	Extraction and Content Authentication	127
7.4	Theoretical Analysis	129
7.4.1	Computational Complexity	129
7.4.2	Watermarking Capacity	130
7.4.3	Sensitivity against Spatial Tampering	132
7.4.4	Sensitivity against Temporal Tampering	133
7.5	Experiments and Results	133
7.5.1	Test Conditions	133
7.5.2	Fidelity after Watermarking	134
7.5.3	Results of Authentication Tests	134
7.6	Summary	139
8	CONCLUSIONS AND RECOMMENDATIONS	143
8.1	Conclusions	143

8.2 Recommendations for Future Work	146
Author's Publications	148
Bibliography	150

List of Figures

1.1	The digital rights management (DRM) pillar model [1].	2
1.2	Progression of image and video coding standardization	4
2.1	Generic digital watermarking scheme	11
4.1	H.264 Encoder [2] (F_n is the current frame, F_{n-1} is the previous frame, F'_n is the reconstructed current frame and ME is the short for Motion Estimation, MC for Motion Compensation, T for Transformation, Q for Quantization).	37
4.2	H.264 Decoder [2] (F'_n is the reconstructed current frame, F'_{n-1} is the reconstructed previous frame and ME is the short for Motion Estimation, MC for Motion Compensation, $\mathbf{T}^{-1}$ for Inverse Transformation, $\mathbf{Q}^{-1}$ for Inverse Quantization).	38
4.3	H.264 Baseline profile, Main profile and Extended profile [2]. . .	40
4.4	Multi-picture motion compensation for a B-picture [3].	41
4.5	Labelling of Intra-prediction samples (4×4) [2].	43
4.6	4×4 luma Intra-prediction modes [2].	44
4.7	16×16 luma Intra-prediction modes [2].	44
4.8	Macroblock partition modes: 16×16 , 8×16 , 16×8 , 8×8 [3]. .	45
4.9	Sub-macroblock partition modes: 8×8 , 4×8 , 8×4 , 4×4 [3]. .	45
4.10	Residual (without motion compensation(MC)) showing choice of block sizes [2].	47

4.11 Performance of the deblocking filter. <i>Left</i> : without the filter. <i>Right</i> : with the filter	48
5.1 Watermark embedding processes of the image-in-image algorithm.	59
5.2 Original images and the corresponding Canny edges.	61
5.3 The watermark pattern (16×16 DMT), the original and water- marked images (512×512 <i>Lena</i> and <i>Baboon</i>).	65
5.4 Watermark extraction processes of the image-in-image algorithm.	66
5.5 Post-processing method again cropping.	68
5.6 The original watermark and extracted watermark (cover image: <i>Lena</i>).	71
5.7 Robustness comparison (correlation) under JPEG compression, Gaussian additive noise, frequency mode line removal and rescal- ing (cover image: <i>Lena</i>).	74
5.8 Robustness comparison (correlation) under JPEG compression (cover image: <i>Lena</i>) (IntDCT stands for the 4×4 integer DCT).	76
5.9 Robustness comparison (correlation) under Gaussian additive noise (cover image: <i>Lena</i>) (IntDCT stands for the 4×4 inte- ger DCT).	77
5.10 Robustness comparison (correlation) under line removal (cover image: <i>Lena</i>) (IntDCT stands for the 4×4 integer DCT). . . .	78
5.11 Robustness comparison (correlation) under rescaling (cover im- age: <i>Lena</i>) (IntDCT stands for the 4×4 integer DCT).	79
6.1 Watermark embedding scheme.	86
6.2 Comparison of the rate-distortion performance for H.264 en- coding without watermark (o) and with watermark (x) (512 kbits/sec, CIF-size).	94

6.3	The 30-th frame of <i>Foreman</i> and <i>Stefan</i> (512 kbits/sec, CIF-size) without watermark and with watermark.	95
6.4	Flow-chart of watermark pre-processing	97
6.5	Flow-chart of watermark Reconstruction	100
6.6	Proposed Video watermarking algorithm	103
6.7	<i>Bit error rate</i> (BER) of the watermark recovered from the P-frames for the sequences before transcoding (a) and after transcoding attacks (b) with two slide-window sizes, $T=1$ and $T=5$. . .	107
6.8	PSNR(dB) of unmarked and marked <i>Foreman</i> , <i>Stefan</i> , <i>Coastguard</i> , <i>Flower</i> , <i>silent</i> and <i>Container</i> (denoted by <i>For</i> , <i>Ste</i> , <i>Coa</i> , <i>Flo</i> , <i>Sil</i> and <i>Con</i> in the horizontal axis, respectively)(768kbits/sec, 512kbits/sec, 396kbits/sec, CIF-size).	111
6.9	The 30th frame of <i>Foreman</i> and <i>Container</i> (768 kbits/sec, CIF-size) without and with watermark.	112
6.10	The 30th frame of attacked <i>Foreman</i> (768 kbits/sec, CIF-size). .	113
6.11	Robustness under transcoding and common signal processing attacks on <i>Foreman</i> , <i>Stefan</i> , <i>Coastguard</i> , <i>Flower</i> , <i>silent</i> and <i>Container</i> (denoted by <i>For</i> , <i>Ste</i> , <i>Coa</i> , <i>Flo</i> , <i>Sil</i> and <i>Con</i> in the horizontal axis, respectively) (768 kbits/sec, 512 kbits/sec, and 396 kbits/sec, CIF-size).	115
7.1	Macroblock partition modes: 16×16 , 8×16 , 16×8 , 8×8	118
7.2	Sub-macroblock partition modes: 8×8 , 4×8 , 8×4 , 4×4	119
7.3	Examples of partition modes using the tree-structured motion compensation.	120
7.4	<i>Foreman's</i> partition mode selection: (a) of the original coded B-frame, (b) of the re-compressed B-frame.	122

7.5	Mode map of the difference between the original coded <i>Foreman</i> B-frame and the re-compressed B-frame (macroblocks with mode change are covered by transparent gray color).	123
7.6	Flow diagram of watermark embedding process (enclosed in the dashed line is the embedding process).	126
7.7	Flow diagram of watermark extraction process.	128
7.8	The number of the nonzero coefficients in the macroblocks with the mode 8×8 without and with watermark.	131
7.9	The original and watermarked <i>Foreman</i>	135
7.10	Sensitivity under transcoding on <i>Foreman</i> , <i>Mobile</i> , <i>News</i> , <i>Bus</i> , <i>Football</i> , <i>Stefan</i> and <i>Coastguard</i> (denoted by <i>For</i> , <i>Mo</i> , <i>Ne</i> , <i>Bu</i> , <i>Foo</i> , <i>St</i> and <i>Co</i> in the horizontal axis, respectively) (512 kbit/s, CIF-size).	137
7.11	Sensitivity under common signal processing on <i>Foreman</i> , <i>Mobile</i> , <i>News</i> , <i>Bus</i> , <i>Football</i> , <i>Stefan</i> and <i>Coastguard</i> (denoted by <i>For</i> , <i>Mo</i> , <i>Ne</i> , <i>Bu</i> , <i>Foo</i> , <i>St</i> and <i>Co</i> in the horizontal axis, respectively) (512 kbit/s, CIF-size).	138
7.12	Sensitivity under cropping and rotation on <i>Foreman</i> , <i>Mobile</i> , <i>News</i> , <i>Bus</i> , <i>Football</i> , <i>Stefan</i> and <i>Coastguard</i> (denoted by <i>For</i> , <i>Mo</i> , <i>Ne</i> , <i>Bu</i> , <i>Foo</i> , <i>St</i> and <i>Co</i> in the horizontal axis, respectively) (512 kbit/s, CIF-size).	139
7.13	The watermarked and the attacked <i>Bus</i> under cutting and pasting attack.	140

List of Tables

4.1	Average bit-rates savings compared with various decoding schemes [3].	36
4.2	Some significant features of the H.264 standard.	39
5.1	Visual quality comparison of watermarked image among different transforms (image <i>Lena</i> and <i>Baboon</i> embedded with <i>DMT</i>). . .	69
5.2	Correlation values of extracted watermarks under StirMark 4.0 attacks.	73
5.3	Performance comparisons (correlation) among different transform algorithms (cover image: <i>Lena</i>) (IntDCT stands for the 4×4 integer DCT, and Had for Hadamard).	80
6.1	Parameters a and d over the frequency index g for watermark pre-processing ($\mathcal{C}$, $\mathcal{L}$ and $\mathcal{D}$ denote Curves, Lines and Diagonals, respectively.)	99
6.2	Reconstructed watermarks after pre-processing and the corresponding correlations ($\mathcal{C}$, $\mathcal{L}$ and $\mathcal{D}$ denote Curves, Lines and Diagonals, respectively.)	102
6.3	The bit error rate of the recovered message after transcoding attacks.	104

6.4	The bit error rate of the recovered message after some common signal processing attacks. All three sequences are compressed at 512 kbits/sec.	105
6.5	Algorithm Comparison	109
6.6	Samples of the original, reconstructed and extracted watermarks from the 30th frame of <i>Foreman</i> (768 kbits/sec, CIF-size) and normalized correlation.	114
7.1	The average coding time (secs) per frame of the unmarked and marked <i>News</i> under different bit rates.	130
7.2	The average coding time (secs) per frame of the unmarked and marked <i>Stefan</i> under different bit rates.	130
7.3	The average PSNR comparison of the unmarked and watermarked frames.	136

List of Notations

NOTATIONS EXPLANATION

α_k	The adaptive watermark gain of the k th cover image sub-block in the embedding process
α_m	The adaptive watermark gain of the m th video sub-block in the embedding process
α_{m_f}	The adaptive watermark gain of the m_f th cover image sub-block in the embedding process
$\hat{\alpha}_{m_f}$	The adaptive watermark gain of the m_f th watermarked image sub-block in the extraction process
β	The scaling factor in the adaptive watermark gain
δ	The parameter that controls the strength of the additive noise
η	The pseudo-noise sequence used in the spread-spectrum method
$\eta(m)$	The m th component of η
θ	The Gaussian additive noise
λ	The predetermined Lagrangian multiplier
μ	The weighting factor of the adaptive watermark gain computation
σ	The variance of η

A	The 4×4 transformation matrix of the traditional DCT
$a(g)$	The additive transformation parameter of $P_l(g)$
B	One macroblock of size 16×16
$\tilde{\mathbf{B}}$	One watermarked macroblock of size 16×16
b	The bit sequence generated from $\mathbf{w}^{\mathbf{p}}$ with S
$b(m)$	The m th component of b
C	The 4×4 transformation matrix of the 4×4 forward traditional DCT
$\mathbf{C}_{\text{for}}$	The 4×4 transformation matrix of the 4×4 forward integer DCT
$\mathbf{C}_{\text{inv}}$	The 4×4 transformation matrix of the 4×4 inverse integer DCT
D	The overall distortion caused by prediction
d	One element in the 4×4 matrix C
$d(g)$	The denominator transformation parameter of $P_l(g)$
E	The 4×4 post-scaling matrix of the 4×4 forward traditional DCT
$\mathbf{E}_{\text{for}}$	The 4×4 post-scaling matrix of the 4×4 forward integer DCT
$\mathbf{E}_{\text{inv}}$	The 4×4 pre-scaling matrix of the 4×4 inverse integer DCT
e	The Canny edge matrix of size $N \times N$
$\mathbf{e}_k$	The k th 4×4 sub-blocks of the 2-D $N \times N$ Canny edge matrix in the spatial domain
F	The length of the watermark sequences w and $\mathbf{w}^{\mathbf{b}}$
f	The index of the watermark sequences w and $\mathbf{w}^{\mathbf{b}}$
G_1	The contribution to G_f from the original video data
G_2	The contribution to G_f from the watermark

G_f	The correlation sum detector for the additive embedding method
$\mathbf{G}_{\mathcal{C}}$	The set of zigzag scanning position indexes of Category $\mathcal{C}$
$\mathbf{G}_{\mathcal{L}}$	The set of zigzag scanning position indexes of Category $\mathcal{L}$
$\mathbf{G}_{\mathcal{LC}}$	The set of zigzag scanning position indexes of Category $\mathcal{LC}$
$\mathbf{G}_{\mathcal{LD}}$	The set of zigzag scanning position indexes of Category $\mathcal{LD}$
g	The zigzag scanning position index
$g(z)$	The generator polynomial of the m-sequence
$g_1, \dots, g_{q-1}$	The scaling factors in the generator polynomial of the m-sequence
H_f	The correlation sum detector for spread-spectrum watermark extraction
$\mathcal{K}$	The set of the sub-block index k , where $\mathcal{K} = \{0, 1, \dots, N^2/16 - 1\}$
$Key(g)$	The key parameter of the $P_l(g)$ in the level reduction process
k	The index of the 4×4 sub-block in the cover image and Canny edge matrix
L	The limitation term used in the adaptive watermark gain computation
N	The vetical/horizontal size of the 2-D cover image
$\mathcal{M}$	The index set of the cover image sub-blocks chosen by the m-sequence
M	The vetical/horizontal size of the 2-D watermark pattern
m	The component index of $\mathbf{w}^s$, where $m = 0, \dots, FS - 1$
m_f	The f th index in the set $\mathcal{M}$
$\mathbf{O}$	The set of all possible prediction modes
o	The current prediction mode
$\tilde{o}$	The current mode after watermarking

o^*	The best prediction mode
$\mathbf{p}$	The 2-D grayscale watermark pattern of size $M \times M$
$\hat{\mathbf{p}}$	The reconstructed 2-D grayscale watermark pattern of size $M \times M$ after extraction
$p_l(i, j)$	The l th 4×4 sub-blocks of the 2-D $M \times M$ watermark pattern in the spatial domain
$P_l(u, v)$	The l th 4×4 sub-blocks of the 2-D $M \times M$ watermark pattern in the integer DCT domain
$P_l(g)$	The g th integer DCT coefficient in $P_l(u, v)$
$P_l^n(g)$	The g th normalized integer DCT coefficient
$P_l^T(g)$	The g th transformed integer DCT coefficient
q	The length of the m-sequence generator polynomial in the binary form - 1
R	The overall consumed bit-rates of the current macroblock
S	The spreading factor in the spread-spectrum method
T	The number of video frame in the slide window
$Thresh_1$	The low sensitivity threshold for Canny edge detection
$Thresh_2$	The high sensitivity threshold for Canny edge detection
t_c	The current video frame index
t_o	The t_o th video frame index
u	The vertical index of a 4×4 matrix in the integer DCT domain
u_o	The certain vertical index of a 4×4 matrix in the integer DCT domain
$V_m(u_o, v_o)$	The original video data in the position (u_o, v_o) of the m th block
$\tilde{V}_m(u_o, v_o)$	The watermarked video data in the position (u_o, v_o) of the m th block

v	The horizontal index of a 4×4 matrix in the integer DCT domain
v_o	The certain horizontal index of a 4×4 matrix in the integer DCT domain
γ_k	The visual mask of Canny edge for the k th cover image sub-block
$\mathbf{w}$	The 1-D watermark sequence after integer DCT for embedding in Chapter 5
$\mathbf{w}_l$	The 16 integer DCT coefficients after zigzag in the l th watermark sub-block
$\hat{\mathbf{w}}$	The 1-D watermark sequence extracted from the watermarked cover image
$\mathbf{w}^b$	The 1-D binary watermark sequence in Chapter 6 and 7
$\hat{\mathbf{w}}^b$	The 1-D binary watermark sequence extracted from the watermarked video
$\mathbf{w}^p$	The bi-polar watermark sequence mapped from $\mathbf{w}^b$
$\hat{\mathbf{w}}^p$	The bi-polar watermark sequence extracted from the watermarked video
$\mathbf{w}^s$	The watermark sequence after applied the spread-spectrum method to $\mathbf{w}^p$
$\hat{\mathbf{w}}^s$	The spread-spectrum watermark extracted from the watermarked video data
$w(m)$	The m th component of the watermark sequence $\mathbf{w}$
$w_l(g)$	The g th integer DCT coefficient after zigzag scanning in the l th watermark sub-block
$w^b(f)$	The f th binary component of $\mathbf{w}^b$
$\hat{w}^b(f)$	The f th binary component of $\hat{\mathbf{w}}^b$
$w_l^b(g)$	The g th binary watermark component of the l th watermark

	sub-block
$w^p(f)$	The f th bi-polar component of $\mathbf{w}^p$
$\hat{w}^p(f)$	The f th bi-polar component of $\hat{\mathbf{w}}^p$
$w^s(m)$	The m th component of $\mathbf{w}^s$
$\hat{w}^s(m)$	The m th component of $\hat{\mathbf{w}}^s$
$\hat{w}(f)$	The extracted f th watermark component of the watermark in Chapter 5
$\mathbf{x}$	The 2-D cover image of size $N \times N$
$\hat{\mathbf{x}}$	The watermarked cover image
$\mathbf{x}_k$	The k th 4×4 sub-blocks of the 2-D $N \times N$ cover image in the spatial domain
$\mathbf{x}_{m_f}$	The m_f th 4×4 cover image sub-blocks chosen by the m-sequence
$\tilde{\mathbf{x}}_{m_f}$	The inverse integer DCT of the matrix $x_{m_f}(i, j)$
$\tilde{\mathbf{x}}_{m_f}$	The m_f th 4×4 watermarked image sub-blocks in the extraction process
$X_m(u_o, v_o)$	The element at the position (u_o, v_o) of the m th video sub-block after integer DCT during embedding
$\hat{X}_m(u_o, v_o)$	The element at the position (u_o, v_o) of the m th watermarked video sub-block after integer DCT during extraction
$X_{m_f}(u, v)$	The integer DCT of $x_{m_f}(i, j)$
$X_{m_f}(u_o, v_o)$	The element at the position (u_o, v_o) of the matrix $X_{m_f}(u, v)$
$\tilde{X}_m(u_o, v_o)$	The watermarked element at the position (u_o, v_o) of the m th video sub-block after integer DCT in Chapter 6
$\tilde{X}_{m_f}(u_o, v_o)$	The watermarked integer DCT coefficient
$\mathbf{y}$	The 4×4 input matrix to the forward traditional DCT /forward integer DCT

$\mathbf{Y}$	The 4×4 forward traditional DCT output of y
$\mathbf{Y}_{\text{for}}$	The 4×4 forward integer DCT output of y
z	The variable used in the m-sequence polynomial presentation

List of Acronyms

ACRONYMS FULL EXPRESSIONS

AVC	Advance Video Coding
B-frame	Bi-predictive frame
BMP	Windows Bitmap
CABAC	Context-based Adaptive Binary Arithmetic Coding
CAVLC	Context-based Adaptive Variable Length Coding
Cb	One chroma sample corresponding to V in Y:U:V sampling format
CHROMA	Chrominance
CIF	Common Intermediate Format
Cr	One chroma sample corresponding to U in Y:U:V sampling format
DCT	Discrete Cosine Transform
DRM	Digital Rights Management
DSP	Digital Signal Processing/Processor
DVD	Digital Versatile/Video Disc
DWT	Discrete Wavelet Transform
ECC	Error Correcting/Correction Code
FFT	Fast Fourier Transform
GOP	Group Of Picture

HDTV	High-Definition Televisions
HVS	Human Visual System
I-frame	Intra-predicted frame
IPMP	Intellectual Property Management and Protection
IPR	Intellectual Property Right
JFCD	Joint Final Committee Draft
JM	Joint Model
JPEG	Join Photographic Experts Group
JVT	Joint Video Team
LUMA	Luminance
LSB	Least Significant Bit
MB	Macroblock
MC	Motion Composition
ME	Motion Estimation
MPEG	Moving Picture Experts Group
MSE	Mean-Square Error
MV	Motion Vector
NAL	Network Adaptation Layer
P-frame	Predicted frame
PSNR	Peak Signal-to-Noise Ratio
QCIF	Quarter Common Intermediate Format
QP	Quantization Parameter
SAD	Sum of Absolute Difference
SI-frame	Switching I-frame
SP-frame	Switching P-frame
SSD	Sum of Squared Difference
UVLC	Universal Variable-Length Coding
VCEG	Video Coding Expert Group

VCL	Video Coding Layer
VLC	Variable-Length Coding
Y	Luma sample

Chapter 1

INTRODUCTION

1.1 Background

Usage of digital video recordings has witnessed a tremendous growth during the last decade as a result of their notable benefits in efficient storage, ease of manipulation and transmission. Unfortunately, the very nature of the digital media makes the work of pirates and hackers easier, since it enables perfect copies with no loss of value. The implications of piracy of digital media could be terrible, since these media play a major role in a wide range of products and services (broadcasting, DVD, Digital Cinema, tele-teaching applications, etc). Concerns have risen about the protection and enforcement of *intellectual property rights* (IPR) for the digital media. For example, given the ease with which media files can be copied, *digital versatile disc* (DVD) burners can facilitate illegal copying. These pirated copies of the genuine articles 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. Conven-

tional DRM systems are heavily based on encryption of the content or parts in order to prevent uncontrolled access [1]. For example, cable TV signals are commonly scrambled to prevent unauthorized viewing. However, encryption alone is not sufficient. It protects contents only during the transmission from the sender to the receiver. Once receipt and subsequent decryption, the media data is no longer protected. However, current technologies already enable consumers to capture perfect copies of a media file as it plays on a computer or a TV set [4].

Figure 1.1: The digital rights management (DRM) pillar model [1].

One effective solution that is gaining popularity in protecting digital contents for copyright owners is digital watermarking. Digital watermarking complements encryption by hiding copyright information into the "essence" of the multimedia object. The hidden information should be imperceptible and robust against malicious attacks. 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. In 1993, Tirkel et al. [5] first introduced the 'electronic watermark' by proposing to hide some identification information into the least significant bits (LSB) of the cover image. This is considered as the beginning of digital watermarking. Thereafter, digital watermarking has drawn a lot of research attention and a significant

number of watermarking algorithms have been presented in the literature. Although some important issues remain unsolved so far [6], it is clear that digital watermarking has emerged as a promising candidate to solve the problem of protecting digital multimedia products.

Watermarking has been considered for many applications. For example, in copy prevention, the watermark may be used to inform software or hardware devices that copying should be restricted. In copyright protection applications, the watermark may be used to identify the copyright holder and ensure proper payment of royalties. A number of other applications include broadcast monitoring, transaction tracking, authentication, copy control and device control.

1.2 Motivation

In the literature, digital video watermarking algorithms can be sorted according to the embedding domain, such as the spatial domain, compressed domain, and bitstream domain. Compressed-domain watermarking methods arouse great interests among researchers, and Hartung et al. [7] embedded spread-spectrum watermark into the MPEG-2 bitstream with drift compensation, but the capacity is reduced. Arena et al. [8] watermarked the MPEG-2 video using interleaved coding. The differential energy watermark method has been extended to video from still images by watermarking the I-frames of an MPEG stream as in [9]. As for low bit-rate video (≤ 1 Mbit/s), Alattar *et al.* [10] extended Hartung's method for MPEG-4 video watermarking. However, the main problem of most previously proposed video watermarking techniques is that those methods are focusing on MPEG standards rather than dealing with the latest H.264/AVC video coding standard specially designed for low bit-rate videos.

Currently, two of the most preeminent standardization bodies are ISO/IEC¹

¹ISO/IEC is the International Standardization Organization and the International Electrotechnical Commission.

Figure 1.2: Progression of image and video coding standardization

and ITU-T². For gray-scale and color 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 Fig. 1.2). For example, current main-stream digital cameras use 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 this thesis, a newly developed compression standard H.264/AVC⁴ is studied for digital watermarking. The H.264/AVC standard is specially designed

²ITU-T is the International Telecommunications Union, Telecommunication Standardization Sector.

³Cameraphones are mobile phones with a built-in digital camera.

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

for low bit-rate applications. It has been shown that some core components of H.264/AVC are quite different from its predecessors. It significantly improves the coding efficiency with the help of many enhanced functional features, such as the highly flexible tree-structured motion compensation, superior rate-distortion related Lagrangian Optimization and the very efficient context-based arithmetic-coding scheme [3]. On average, H.264/AVC has achieved 50% less in overall bit-rates with the same visual quality, when compared with existing standards such as MPEG-4 Video [3]. Anyhow, a higher compression ratio leads to difficult balance among tradeoff requirements for watermarking video data [11], [12]. It also provides a "network-friendly" video representation which addresses "conversational" (video telephony) and "non-conversational" (storage, broadcast or streaming) applications. Therefore, it has significant applications in the areas of Digital Cinema, DVD and mobile communication.

While these new features enable H.264/AVC to achieve superior performance in terms of coding efficiency and video representation, they also impose great challenges for traditional watermarking methods, and thus H.264/AVC is worthwhile to investigate new watermarking approaches. Our goal is to improve the visual quality, flexibility and robustness/sensitivity of watermarking system in dealing with the low bit-rate videos.

Furthermore, H.264/AVC utilizes a 4×4 *integer Discrete Cosine Transform* with low-complexity, instead of the commonly used 8×8 DCT found in prior video coding standards. The integer DCT does not only have the same properties as the traditional DCT, but also has many advantages over DCT such as shorter processing time, higher computation accuracy and ease of hardware implementation. Therefore, to test the 4×4 integer DCT for its performance in watermarking schemes, a robust image watermarking scheme is proposed in the first chapter of this thesis. With acceptable results, video watermarking schemes are studied in the following chapters for different applications.

1.3 Objectives

The main objective of this thesis is to research, develop and analyze novel watermarking algorithms of the state-of-the-art video coding standard H.264/AVC, with feature extraction techniques, for copyright protection and authentication.

1.4 Major Contributions

The main contributions in this thesis are as follows

1. A novel proposal for image-in-image watermarking

We first investigated an image-in-image watermarking algorithm [13] by using an integer transform that is an approximation to DCT used in the H.264/AVC standard. A feature extraction technique and some post-processing techniques were proposed for better performance. The novel image-in-image watermarking scheme showed good performance and was benchmarked with StirMark 4.0. A transform performance tradeoff between various DCT transforms showed that the 4×4 integer DCT outperformed the traditional DCT under several signal processing attacks, such as JPEG compression, Gaussian additive noise and scaling.

2. Robust watermarking methods for H.264/AVC standard

An efficient and robust video watermarking algorithm for the state-of-the-art video coding standard H.264/AVC was proposed for copyright protection. A binary sequence acting as an ID number was served as watermark and embedded in the host video. Also grayscale 2-D watermark patterns such as detailed trademarks or logos were highly compressed by a proposed grayscale watermark pre-processing, and inserted in the compressed domain. The marked video sequences maintained good visual quality and the same overall consuming bit-rate. Furthermore, the

binary sequence watermark retrieval method was innovative in accumulating the watermark bits within several successive frames. The proposed algorithm can robustly survive the transcoding process and common signal processing, such as bit-rate reduction, Gaussian filtering and contrast enhancement.

3. An Effective Authentication method for H.264/AVC standard

We proposed a novel and efficient scheme of fragile video authentication. The scheme made an accurate usage of the tree-structured motion compensation, motion estimation and Lagrangian optimization of the standard. Authentication information was embedded strictly based on the best mode decision strategy in the sense that if watermarked video content undergoes any spatial and temporal attacks, the scheme can detect the tampering by the sensitive mode change. The experimental results proved the effectiveness of the algorithm against many transcoding and signal processing attacks, such as recompression with bit-rate change, Gaussian additive noise, and cutting and pasting attacks.

1.5 Thesis Outline

This thesis is organized as follows:

Chapter 1 Introduction

We introduce the motivation, objectives, main contributions, publications and organization of this thesis.

Chapter 2 Watermarking Basics

Some basic principles, such as general frameworks of watermark embedding and decoding processes, and watermarking requirements and applications are discussed in Chapter 2. Attacks and benchmarking tools are also discussed.

Chapter 3 Literature Review of Watermarking

This chapter gives a review of various watermarking technologies for images and videos, respectively.

Chapter 4 The H.264/AVC Standard

In Chapter 4, we introduce the state-of-the-art H.264/AVC standard. Its history, a brief technical report related to our research and some applications are presented.

Chapter 5 Proposed Robust Watermarking of Digital Images

In this chapter, to test the 4×4 integer DCT for watermarking performance, a robust watermarking scheme for copyright protection will be proposed for still images. Feature extraction and post-processing techniques are applied to achieve good visual quality and high robustness of the image scheme.

Chapter 6 Proposed Robust Watermarking of H.264/AVC Video

In this chapter, we extend the idea of using the 4×4 integer DCT in the image-in-image algorithm to H.264/AVC video clips in the compressed domain and develop robust watermarking schemes for copyright protection. We propose a robust watermarking method with two different watermarks. The first watermark is a binary sequence containing copyright information and the second one is a two-dimensional watermark pattern with the proposed watermark pre-processing.

Chapter 7 Proposed Hard Authentication of H.264/AVC Video

Based on the previous exhaustive investigation of the related techniques of the state-of-the-art H.264/AVC standard, we further develop a novel and efficient watermarking algorithm for fragile video authentication. This algorithm works directly in the compressed domain. The scheme also makes an accurate usage of the tree-structured motion compensation, motion estimation and Lagrangian optimization for mode decision in the H.264/AVC standard.

Chapter 8 Conclusions and Recommendations

Finally, the conclusions are drawn and some research recommendations are

suggested in Chapter 8.

Chapter 2

WATERMARKING BASICS

2.1 Basic Structure of Digital Watermarking

In digital watermarking, a low-energy signal is imperceptibly hidden in a main signal. The low-energy signal is called watermark and it depicts some data, such as security of rights information about the main signal. It can be a set of characters, a trademark image, or even a short period of video [14]. The main signal in which the watermark is embedded is referred to as a cover signal or a host signal since it covers the watermark. The cover signal is generally a still image, audio clip, video sequence or a text document in digital format.

Figure 2.1: Generic digital watermarking scheme

The digital watermarking system essentially consists of a watermark embedder and a watermark detector (see Figure 2.1). The watermark embedder inserts a watermark into the cover signal. The watermark detector detects the presence of the watermark signal and it needs to retrieve the watermark signal in some applications such as authentication. An entity called watermark key is used during the process of embedding and detecting the watermark signal. The watermark key has a one-to-one correspondence with the watermark signal (i.e., a unique watermark key exists for every watermark signal). The watermark key is private and is known only to authorized parties and it ensures that only authorized parties can detect the watermark. The communication channel can be noisy and hostile (i.e., prone to security attacks) and hence the digital watermarking techniques should be resilient to both noise and security attacks.

2.2 Watermarking Properties

Watermarking systems can be characterized by a number of defining properties. The relative importance of each property is dependent on the requirements of the application and the role of the watermark.

2.2.1 Fidelity

One of the main requirements for watermarking is the *fidelity*. The embedded watermark must be perceptually transparent. The data embedding process should not introduce any perceptive artifacts into the cover data. Distortions introduced through a watermarking process are not only annoying and undesirable, but may also reduce or destroy the commercial value of the watermarked data. It is therefore important to design a watermark which exploits characteristics of the human visual or auditory system in order to maximize the watermark energy for high robustness under the constraint of the perceptible

threshold.

2.2.2 Embedding Capacity

Embedding capacity refers to the number of bits a watermark encodes within a unit of time or within the media. For an image, the capacity would refer to the number of bits encoded within the image. For audio, capacity refers to the number of embedded bits per second that are transmitted. For video, the capacity may refer to either the number of bits per frame or the number of bits per second. Different applications require different capacities. Generally, the capacity should be as high as possible. In these cases, the algorithms must be able to embed a large amount of data, while maintaining the robustness level and the integrity of the cover signal.

2.2.3 Informed or Blind Detection

A detector is referred as an *informed detector* when it requires access to the original or some information of the original unwatermarked media. Conversely, detectors that do not require any information related to the original are referred as *blind detectors*. Whether a watermarking system employs a blind or an informed detection can be critical in determining if it can be used for a given application. Informed detection can only be used in those applications where the original data is available. Watermark recovery is usually more robust if the cover signal is available. The reason is that the cover signal allows the detection and the inversion of distortions which change the data geometry [15]. However, for some applications, like video watermarking, it may be impractical to use the cover data because a large data volume must be stored in the database.

2.2.4 Robustness

Robustness of the watermarked data is one of the key requirements in watermarking. Robustness refers to the ability to detect the watermark after common signal processing modifications and malicious attacks. Examples of common operations on images include spatial filtering, lossy compression, printing and scanning, and geometric distortion (rotation, translation, scaling, and so on). Video watermarks may need to be robust to many of the same transformations, as well as to recording on video tape and changes in frame rate, among other influences.

Not all watermarking applications require robustness to all possible signal processing operations. A robust watermark is usually used to verify the ownership of the protected work, therefore an ideal robust watermark should resist as many kinds of distortions as possible. On the other hand, an important branch of watermarking research focuses on fragile watermarks. A fragile watermark is designed for content authentication or tamper proofing, and thus it must be sensitive to any accidental or intentional manipulations. A semi-fragile watermark is also sensitive to most attacks but can survive some pre-defined attacks (e.g., lossy compression).

There are two main benchmarking tools, Stirmark [16] and Checkmark [17], available on the Internet which could be used to test the robustness of a watermarking method.

2.2.5 Security

In most applications, such as copyright protection, the secrecy of the embedded information needs to be assured. If secrecy is a requirement, a secret key has to be used for the embedding and extraction processes. A secure watermarking procedure cannot be broken unless the unauthorized user has access to a secret

key that controls the insertion of the data in the cover signal [18].

2.2.6 Complexity

Computational complexity of the watermark encoding and decoding processes is another consideration. In some real-time applications, e.g., copy control of a DVD disc, it is important that the algorithms must operate efficiently.

2.3 Classification of Watermarking Techniques

Digital watermarking techniques can be classified in a number of ways depending on different parameters. Various types of watermarking techniques exist and are enlisted with their respective applications.

- Robust, Semi-fragile and Fragile Watermarking

Robust watermarking is a technique in which modification to the watermarked content will not affect the watermark, and which is widely used for the application copyright protection. As opposed to this, fragile watermarking is a technique in which the watermark gets destroyed when the watermarked content is modified or tampered. Semi-fragile watermarking techniques are capable of tolerating some degree of change to the watermarked content. Fragile and semi-fragile techniques are suitable for the authentication application.

- Visible and Transparent Watermarking

Visible watermarks are ones which are embedded in visual content in such a way that they are visible when the content is viewed. Transparent watermarks are invisible and they cannot be detected by just viewing the digital content.

- Informed and Blind Watermarking

Informed watermarking techniques require access to the original or some information of the original data. Blind watermarking techniques do not need any information related to the original data for watermark detection. Generally, informed watermarking schemes are more robust, while blind watermarking approaches are more desirable in practice, as informed schemes need huge storage for the original data. Furthermore, in some applications (e.g., copy control), the original data is not even available or accessible.

- Spatial-domain and Transform-domain Watermarking

A watermark can be inserted either in the spatial domain or transform domain. The advantages of the spatial-domain-based methods include low computational load and negligible degradation to the watermarked signals. Anyhow, these methods are usually vulnerable to common signal processing manipulations (e.g., compression), and thus cannot meet the requirement for robustness in many cases like copyright protection. On the other hand, transform domain watermarking, which inserts the watermark into the transformation components of the cover signal, has been widely accepted for robust watermarking.

2.4 Benchmarking for Watermarking Systems

Besides designing digital watermarking methods, an important issue in developing robust digital watermarking algorithms for image and video data is to address the proper evaluation and benchmarking methods. This not only requires evaluation of the robustness, but also includes subjective or quantitative evaluation of the distortion introduced through the watermarking process. The emergence of watermarking technology stimulates some individuals to come

up with attempts to defeat its functions such as copyright protection. Such activities are broadly referred to as watermark "attacks".

Examples of such attack efforts are publicly available tools such as StirMark 4.0 [16] and Checkmark [17]. StirMark 4.0 is a utility for testing the robustness of image watermarking techniques by applying minor geometric distortions to the watermarked images. StirMark 4.0 introduces a practically unnoticeable quality loss in the image if it is applied only once. The benchmark subjects watermarked images to a variety of attacks. The resistance of different watermarking methods to the same group of attacks is then averaged to allow comparison.

Some samples of StirMark 4.0 attacks are listed below

- Centered cropping of 1%, 2%, 5%, 10%, 15%, 20%, 25%, 50% or 75%;
- Symmetric and asymmetric frequency row and column removal;
- General linear transformation;
- Scaling by factors: 0.5, 0.75, 0.9, 1.1, 1.5 or 2;
- Additive noise by factors: 0%-100%;
- Symmetric and asymmetric shearing for $a\%$ of the length in X direction and/or $b\%$ of the height in Y direction;
- Rotation by a specified angle;
- Rotation + Auto-cropping + Auto-scale (to maintain the original size of the image);

- Sharpening using spatial mask

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix};$$

- Gaussian Filtering using spatial mask

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix};$$

- Median flitering;
- JPEG compression;
- Randomization and bending;
- Random geometric distortions.

Since SitrMark 4.0 is designed for image watermarking techniques and there are no special benchmarks for video watermarking technologies to-date, StirMark 4.0 attacks were applied to each frame of the decompressed video sequence in this thesis. This way of simulating video attacks has been used by many researchers, such as Jin et al. [19]. As long as we have the same distortion parameters applied to the first frame and apply them to the other frames, we can make sure that distortions are not different for each frame. This action is highly necessary, because the distortions are different for each picture by default in StirMark 4.0.

In this thesis, several acceptable attacks for video watermarking were chosen such as Gaussian filtering, Sharpening, Additive Noise, Median Filtering and Frequency Mode Laplacian Removal. For practical applications, various attacks specified to video can take place, such as frame-ratio changing, format conversion, frame rate changing, cutting off few frames in every *group of pictures*(GOP) and so on.

2.5 Applications of Watermarking

Digital watermarking techniques have a wide range of applications. Some typical applications are described below.

2.5.1 Copyright Protection

Copyright protection, which is probably the most prominent application of digital watermarking, is to embed some identification information so that the copyright owner can prevent other parties from reclaiming the copyright on the work. The product might have been manipulated by an adversary in an effort to remove or destroy the watermark or replace it with his own. Therefore, a watermark for copyright protection requires a high level of robustness.

2.5.2 Content Authentication

Watermarks can be used to check whether the data or the meaning of the content have been altered and even provide localization information about where the content was tampered. If the content has been modified, the copyright owner may hope to know how and to what extent, the content has been impaired. This could be achieved with fragile or semi-fragile watermarks, which are able to reflect the modifications the work has undergone. Among all the possible watermarking applications, watermarks for authentication require the

lowest level of robustness.

2.5.3 Fingerprinting

Watermarks such as a user ID, or well known as *fingerprints*, are characteristics of an object that tend to distinguish it from other similar objects. It can be used to track the source of pirated copies and enable legal action to the user who breaks the licence agreement. In the scenario of fingerprinting, the distributor usually embeds different watermarks into each distributed copy of the product, in order to monitor or trace illegal copies. Multiple-bit watermarks should be employed to enable user identification.

2.5.4 Copy Control

The content provider is sometimes interested in constraining the level of control that the users may have on the purchased product. It is possible to embed a copy information into a product and use this information to indicate the copy status of the data. Depending on the information of copy status, the devices can allow or prohibit certain operations on the content. An example is that the DVD or CD player can use devices to embed or detect copy control information, and can prohibit illegal copying of the discs [20].

2.5.5 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.

Chapter 3

REVIEW OF WATERMARKING TECHNIQUES

Digital watermarking hides imperceptible but persistent pieces of information into the digital data to be protected. It has been proposed for copyright control, tracing, broadcast monitoring, authentication, to mention a few. Many digital watermarking schemes have been proposed in the literature for still images and video. Image watermarking is most widely used. It works with feature extraction techniques to utilize the significant characteristics in the host images for watermark embedding and detecting. Still image watermarking techniques can be extended and applied in video signals, since video can be considered as a sequence of still images. However, digital video is generally stored and distributed in compressed format (e.g., MPEG-X, H.26X), in which the compression algorithms take advantage of temporal and spatial redundancy in the video.

In this chapter, we begin with a discussion on watermarking techniques for still images, in the spatial domain and the transform domain, followed by

watermarking for video contents. Robust techniques for copyright protection applications are mainly concerned here. Also, digital multimedia authentication techniques have witnessed a tremendous rise in interest over the past few years. A video authentication system ensures the integrity of digital video, and verifies that the video taken into use has not been tampered. Thus, for the application of authentication, proposed techniques for images and video in the literature will be described in the later part of this chapter.

3.1 Image Watermarking Techniques

Many techniques have been developed for the watermarking of still images. For gray-level or color-image watermarking techniques, a watermark is designed to be inserted directly into the original image data, such as the luminance or color component. Or a watermark can be embedded into some transformed version of the original data to take advantage of the perceptual properties or robustness to particular signal processing manipulations. Main requirements for image watermarking include fidelity, robustness to common signal processing operations, and capacity. Common signal processing operations for which a watermark should survive include compression (such as JPEG), filtering, rescaling, cropping, additive noise and geometric distortions.

3.1.1 Spatial Domain

In early spatial-domain watermarking schemes, *Least Significant Bits* (LSB) coding is the most common method used to insert a watermark [21]. It is straightforward that the LSB of an image do not contain significant visual information. Thus, the LSB can be modified to conceal a large number of watermark bits. Tirkel *et al.* in [5] proposed an algorithm in which the watermark bits were in form of an m-sequence derived PN (pseudo-noise) code, and

embedded a watermark by either replacing or adding to the LSB. The latter embedding method required the autocorrelation function for watermark extraction. The m-sequence was chosen due to the good correlation properties so that a correlation operation for detection can be used. Then, an improved version was published in [22]. Generally, this type of technique is not robust enough. In [23], the idea of using an m-sequence and LSB coding was extended to two-dimension. A watermark is added and detected on a block-by-block basis. The block-based method has been shown to be robust to JPEG compression.

Another spatial-domain technique described as "Digital Signature of Color Images" in [24] is based on amplitude modification of the blue color component of an image in RGB format. A watermark is embedded by the addition or subtraction operation on the blue component amplitude to ensure robustness while remaining fairly insensitive to the human visual system (HVS) factors. Moreover, the original image is not necessary to extract the watermark and can be estimated by a linear combination of pixel neighbors one by one. The method has been shown to be resistant to some attacks, such as filtering and geometrical attacks.

The characteristics of the human visual system are widely exploited in designing digital watermarking schemes. Feature extraction techniques have been incorporated to utilize the significant characteristics in the host data for watermark embedding and detecting. For example, salient points in an image, which can be found by a certain feature point detector (such as Enhanced Harris Detector in [25]) can be used as reference points to achieve synchronization between watermark embedding and extraction [19].

Kutter *et al.* [26] proposed a perceptual model for hiding a spread-spectrum watermark of variable amplitude and density in an image. The model takes into account the sensitivity and masking behavior of the human visual system of a local isotropic contrast measure. The experimental results showed that the

algorithm facilitates the insertion of a robust watermark in either the luminance image or blue channel while preserving the visual quality of the original image.

3.1.2 Transform Domain

Transform-domain watermarking is useful in that it takes advantage of the perceptual criteria in the embedding process, that also the watermark can be embedded directly into the compressed bit stream. Therefore the watermark can be more robust to common signal processing manipulations. Watermarks can be inserted in frequency domain by applying transforms like Discrete Cosine Transform (DCT), Discrete Wavelet Transform (DWT) and Fast Hadamard Transform (FHT), and then altering the values of selected transform coefficients to store the watermark in still images. In this section, we will introduce some typical watermarking algorithms based on several most commonly used transforms.

The DCT transform has found wide applications in image/video processing and many other fields. It has become the heart of many international standards such as JPEG, H.26X and MPEG-X family. The DCT is a robust approximation of the optimal Karhunen-Loève Transform (KLT) for a first-order Markov source with a large correlation coefficient. It has satisfactory performance in terms of energy compaction capability. Many fast DCT algorithms with efficient hardware and software implementations have been proposed.

One of the very first block-based DCT watermarking technique was proposed in [27]. The DCT is performed on 8×8 blocks of data, a pseudo-random subset of the blocks are chosen and a triplet of mid-range frequencies are slightly altered to encode a binary watermark sequence. This is a heuristic watermarking technique, because watermarks embedded in the high frequencies are vulnerable to attacks whereas those in the low frequencies are perceptually significant and sensitive to alterations. This method shows good robustness to

JPEG compression for a quality factor as low as 50%.

One of the first most influential watermarking techniques [28], [29] describes how spread spectrum principles borrowed from digital communication theory can be used in the content of watermarking. The published results show that the technique is very effective both in terms of image quality and watermark robustness to signal processing attacks and attempts to remove the watermark. The technique was motivated by both perceptual transparency and watermark robustness. One of the significant contributions in this work is the realization that the watermark should be inserted into the perceptually significant portion of the image in order to be robust to the attacks. A DCT is performed on the whole image and a watermark is inserted into the perceptually significant AC components. The watermark consists of a sequence of real numbers generated from a Gaussian distribution and is added to the DCT coefficients. The watermark is scaled according to the signal strength of the particular frequency component. This is a reasonable and simple way to introduce some type of perceptual watermark weighting. The watermark embedding algorithm can be described as

$$\tilde{X}_i = X_i(1 + \alpha w_i) \quad (3.1)$$

where $\tilde{X}_i$, X_i and w_i denote the modified frequency components, the original frequency components and the watermark, respectively, and the Gaussian distribution sequence α determines the watermark strength adaptive to different spectral components, which the authors suggested to be 0.1 to provide a good tradeoff between fidelity and robustness.

To verify the presence of the watermark, the normalized correlation between the original watermark $\mathbf{w}$ and the retrieved watermark $\tilde{\mathbf{w}}$ was computed, which

was given by

$$\text{corr}(\mathbf{w}, \tilde{\mathbf{w}}) = \frac{\tilde{\mathbf{w}} \cdot \mathbf{w}}{\sqrt{\tilde{\mathbf{w}} \cdot \tilde{\mathbf{w}}}} \quad (3.2)$$

The authors in [28], [29] claimed that the similarity measure is also normally distributed so that a high similarity value was extremely unlikely for $\mathbf{w} \neq \tilde{\mathbf{w}}$.

The presence of the watermark $\mathbf{w}$ can be determined if $\text{corr}(\mathbf{w}, \tilde{\mathbf{w}}) > Tresh$, where $Tresh$ is a threshold. Experiments showed that this non-blind watermarking scheme can resist JPEG compression at a 5% quality factor and several other kinds of attacks.

Many algorithms have been proposed to improve this method. Lu *et al.* [30] used the cocktail watermark to improve the robustness and took advantage of the human visual system to maintain high fidelity of the watermarked image. Huang *et al.* [31] embedded a watermark pattern by modifying the DC components. Malvar *et al.* [32] introduced a new watermarking modulation technique, which is referred to as the *Improved Spread Spectrum* (ISS). When compared with the traditional spread spectrum, the ISS signal does not act as a noise source, leading to significant gain. Experimental results showed that the performance improvements over the traditional spread spectrum method was 20dB in signal-to-noise ratio (SNR).

The 2D-Hadamard transform has been used extensively in image processing and image compression. The advantage of the Hadamard transform is that it contains more useful middle and high frequency bands for digital watermarking. These middle and high frequency coefficients have components equivalent to where many DCT low-frequency AC coefficients are located. It is more likely that in a very noisy environment, the Hadamard transform bands could survive different attacks as compared to other orthogonal transforms such as DCT.

The Hadamard matrix of order n is generated in terms of a Hadamard

matrix of order $n - 1$ using the Kronecker product

$$H_n = H_{n-1} \otimes H_1, \quad n = 2, 3, \dots \quad (3.3)$$

where H_1 is the core Hadamard matrix defined by

$$H_1 = \begin{bmatrix} 1 & 1 \\ 1 & -1 \end{bmatrix}. \quad (3.4)$$

Ho *et al.* [33] proposed an image-in-image watermarking approach for copyright protection, in which an 8×8 Hadamard transform was applied to both the cover image and the watermark pattern. The third order Hadamard matrix H_3 was used and the watermark was embedded by replacing 16 middle and high frequencies of the Hadamard coefficients. The watermark gain was determined from the texture by examining the AC energy, and from the edge density by employing the Canny edge detection [34]. The experimental results showed that the proposed Hadamard-based watermarking algorithm achieved a good performance of robustness against a variety of attacks, such as additive noise and median filtering.

3.2 Video Watermarking Techniques

The proliferation of digital video applications, such as videoconference, DVD, and HDTV, have aroused the attention of industries and researchers to the *intellectual property rights* (IPR) protection of video contents. Digital watermarking hides imperceptible but persistent pieces of information into the video to be protected. It thus has been proposed for copyright control, tracing, broadcast monitoring, authentication, to mention a few.

Digital watermarking for video is a fairly challenging area of research and

basically this area benefits from the results for still images [35]. Many algorithms have been proposed in the scientific literature. The most simple approach is to consider a video sequence as a series of consecutive still images. Hence still image watermarking techniques can be applied straightforward to video content in the uncompressed domain. However, digital video is usually stored in a compressed format that exploits the temporal redundancy in the video sequence to spare some storage space. The availability of the temporal dimension of video sequences provide not only new opportunities for watermarking, but also impose new challenges. A video stream can be considered as some data compressed according to a specific video compression standard. Therefore, the characteristics of such a standard can also be used to design an efficient watermarking scheme.

3.2.1 Still Images Techniques

The very first proposed algorithm for video coding was Moving-JPEG (M-JPEG) [35], which simply compresses each video frame with the image compression standard JPEG. The simplest way is to insert the same watermark as in the still image algorithms, into the video frames at a regular rate. However, this scheme has no payload because the detector of each frame can only tell if the watermark is present or not (referred as the *one-bit watermarking*). Dittmann *et al.* [36] described a method of embedding independent multi-bits watermark into each frame of the video.

Cox *et al.* [28] applied the block-based spread-spectrum watermarking method for images as well as for video data. Hartung *et al.* [7] proposed an additive spread-spectrum watermarking scheme, in which the video data is treated as a one-dimensional signal acquired by line-scanning.

3.2.2 Compressed Domain Techniques

The main drawback of considering a video as a succession of independent still images is that it does not satisfactorily take into account the new temporal dimension. Since most of the digital video commercial products are sold and circulated in compressed standard-compliant format, numerous compressed-domain methods have been proposed.

Currently, most contemporary video compression standards (such as MPEG-2, MPEG-4 and H.263) employ block-based hybrid coding methodology, which compresses video frames in either intra-coding or inter-coding in order to exploit spatial and temporal redundancy, respectively. The intra-coded frames (I-frame) is encoded by block-based DCT without referring to other pictures, and with basically the same technique of JPEG standard. The P- and B-frames, collectively known as inter-coded frames, are predicted from other frames by motion estimation and compensation processes, and thus only the prediction errors (the changes from current picture to its reference frames) and motion vectors need to be encoded. More details about video coding standards will be described in the next chapter. Therefore, a compressed-domain watermarking method often exploits the DCT coefficients (before or after quantization) and motion vectors to hide information.

Based on the spread-spectrum watermarking method in the uncompressed-domain of MPEG-2 in [7], Hartung *et al.* extended the method into compressed-domain. An MPEG-2 video is first partially decoded to obtain the DCT coefficients. These coefficients are added by the DCT transformed watermark coefficients, subject to the fidelity constraint. The total bit rate of the watermarked compressed bit stream can not exceed the total bit rate of the original unmarked bit stream and this additional constraint should be taken into account. This is an important requirement for video watermarking techniques, because for many applications, the bandwidth limitations dictate the total bit

rate possible for a given bit stream. Only nonzero DCT coefficients are marked and if a constraint bit rate is required, the DCT coefficients are marked only if the bit rate for the quantized representation is equal or less than the bit rate needed for the unmarked quantized coefficients. Although much of the video may not be marked due to this additional constraint, it is still possible to embed a few bytes of information per second. The watermark extraction is performed by correlation after the encoded video is fully decoded.

Alattar *et al.* [10] further extended this method for MPEG-4 video watermarking. A synchronization template combats geometric attacks. The method also features a gain control algorithm that adjusts the embedding strength of the watermark, depending on the local picture characteristics, increasing watermark robustness, or equivalently reducing the watermark distortion on visual quality. A drift compensator prevents the accumulation of the watermark distortion and reduces the watermark self-interference. The bit rate of the watermarked video is controlled within an acceptable limit. The watermark was evaluated to be robust against a variety of attacks, such as transcoding, scaling, rotation and noise reduction.

A method of differential energy watermarks (DEW) was initially proposed for robust image watermarking and then extended to I-frames of MPEG stream by Langelaar *et al.* [9]. The DEW algorithm embedded label bits by selectively discarding high frequency DCT coefficients in certain image regions, and introduced an energy difference between two regions in a group of 8×8 blocks. The maximal coarseness of the quantizer, the number of useful DCT blocks, and the lowest DCT coefficient were carefully studied so as to optimize the performance in robustness, capacity and fidelity. The experiments showed that this scheme is robust against attempts to remove watermark by re-encoding the JPEG or MPEG bit stream.

Other techniques proposed for video watermarking in the compressed-domain

include embedding a watermark into the motion vectors (MV). A motion vector is one of the most important syntax element of the compressed video and is generated in the motion estimation process. Jordan *et al.* [37] first proposed a video watermarking technique to hide copyright information by slightly modifying the motion vector in an MPEG video stream. By selecting the inter-frames in the MPEG sequence and according to the bitstream of copyright information, the motion vectors are regularized into a modified bitstream, from which the watermark can be easily retrieved. Zhang *et al.* [38] improved the above method by embedding the watermark bit into the motion vectors with magnitude equal to or larger than a set threshold. The experiments showed that there was little degradation in the video quality. Bodo *et al.* [39] proposed a robust motion vector watermarking method based on a hierarchical motion analysis.

3.2.3 Bit Stream Techniques

Often digital video will already be in a compressed format when watermarking is applied. It is desirable to be able to embed the watermark directly into the compressed bit stream without going through a full decoding, watermarking and re-encoding steps, which will add considerable complexity and additional delay. For example, in [40] and [41], techniques add watermark by modifying the fixed length and variable length codes in the compressed video bit stream. This allows for a computationally efficient way of real-time watermark insertion and allows for a relatively high capacity.

3.3 Techniques in Authentication Applications

Digital multimedia authentication techniques have witnessed a tremendous rise in interest over the past few years. By definition, authentication is a process whereby an entity proves the identity to another entity [42]. In a multimedia

context, video authentication aims to establish its veracity in time, sequence and content. A video authentication system ensures the integrity of digital video, and verifies that the video taken into use has not been tampered. Video authentication is important in many applications such as surveillance, journalism and video broadcast, etc.

In the past, several techniques and concepts based on data hiding or steganography, have been introduced for tamper detection in digital images and video. One class of authentication watermarks is *Hard Authentication* [43]. Hard Authentication rejects any modifications to multimedia content. The inserted watermark is so weak that any manipulations to the multimedia content disturbs its integrity.

One of the first fragile watermarking techniques was to insert key-dependent check sums of the seven most significant bits into the LSB of the pseudo-randomly selected pixels as proposed in [44] and [45]. A simple approach referred as the Yeung-Mintzer Scheme [46] enabled single pixel authentication but only half of the modified pixels on average can be detected. The scheme's security depends critically on the secrecy of the watermark logo. Another approach is to partition a multimedia signal into two disjointed parts: a signature part and an embedding part. An authenticator such as a message authentication code or a digital signature is generated from the content of the signature part and is then embedded into the embedding part. Lossless watermarking in [47] and [48] used a spatially additive, signal-independent robust watermarking to embed signal authentication data using a reversible modular addition. The watermark has to be robust enough to survive the reversible addition in the watermarking process. Thus, for an unmodified watermarked signal, the authentication data can be correctly recovered and the original watermark can be subsequently regenerated from the watermarked signal. The amount of authentication data is typically constrained by the limited embedding capacity of

the underlying robust watermark.

Chapter 4

THE H.264/AVC STANDARD

4.1 Introduction

The Joint Video Team (JVT) was formed by members of the Moving Picture Experts Group (MPEG) and the Video Coding Experts Group (VCEG), and the team has developed a new standard for the coding (compression) of natural video images that outperforms the earlier MPEG-4 and H.263 standard. The new video standard is entitled 'Advanced Video Coding' (AVC) and is published jointly as Part 10 of MPEG-4 and ITU-T Recommendation H.264 [49]. The main goals of this standardization effort are to develop a simple and straightforward video coding design, with enhanced compression performance, and to provide a "network-friendly" video representation which addresses "conversational" (video telephony) and "non-conversational" (storage, broadcast or streaming) applications. Two major improvements are

- **Improved coding efficiency**

Compared with existing video coding standards, the H.264 permits a reduction in bit rates by 50% or greater on average for a similar degree of encoder optimization at most bit rates while achieves essentially the same objective PSNR reproduction quality [50], [51], as shown in Table 4.1.

Coder	MPEG-4	H.263	MPEG-2
H.264/AVC	38.62%	48.80%	64.46%
MPEG-4		16.65%	42.95%
H.263			30.61%

Table 4.1: Average bit-rates savings compared with various decoding schemes [3].

• Network friendliness

H.264/AVC conceptually separates into two layers: the video coding layer (VLC) 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 [3]. In addition, error resilience tools and other network functionalities, which have been adopted by prior standards, such as MPEG-4 have been further improved [3], [52].

4.2 The H.264 Codec

In compliance with the early standards (MPEG-1, MPEG-2 and MPEG-4, H.261 and H.263), the H.264 draft standard does not explicitly define a CODEC (enCOder/DECoder pair), but defines the syntax of coding a video and decoding this coded video bitstream.

Figure 4.1 shows the encoder with a 'forward' coding path (left to right) and a 'backward' reconstruction path (right to left). Figure 4.2 shows the decoder with the decoding path from right to left to illustrate the similarities between the encoder and decoder.

Most of the basic functional elements are present in previous standards except for the deblocking filter, but the important changes in H.264 standard occur in the details of each functional block. Table 4.2 lists some main features of the H.264 standard. Some of these features are an extension from either

Figure 4.1: H.264 Encoder [2] (F_n is the current frame, F_{n-1} is the previous frame, F'_n is the reconstructed current frame and ME is the short for Motion Estimation, MC for Motion Compensation, T for Transformation, Q for Quantization).

Figure 4.2: H.264 Decoder [2] (F'_n is the reconstructed current frame, F'_{n-1} is the reconstructed previous frame and ME is the short for Motion Estimation, MC for Motion Compensation, T^{-1} for Inverse Transformation, Q^{-1} for Inverse Quantization).

Features	Description
Y to Cr to Cb sampling ratio	4:2:0
Size of macroblock	Tree structured motion compensation
2 new types of frame	SP, SI frame
Sub-pixel motion vectors	Quarter-pixel accuracy
Method for AC coefficients transformation	New integer Discrete cosine transformation (DCT)
Method for DC coefficients transformation	Hadamard transformation
Reference frames	5 to the previous and 5 to the future frames
Reduction of deblocking artifacts	In-the-loop deblocking filtering
Method used in Entropy encoding	Context-based adaptive arithmetic coding (CABAC); Context-based adaptive variable length coding (CAVLC)

Table 4.2: Some significant features of the H.264 standard.

H.263 or MPEG-4. Others are purely new functions that are added to H.264 standard.

4.3 H.264 Structure

4.3.1 Profiles and Levels

In a video coding design, profiles and levels specify the conformation points. These conformation points are designed to facilitate inter-operation between various applications that have similar functional requirements. A *profile* defines a set of coding tools or algorithms that can be used in generating a compliant bitstream, and a *level* places constraints on certain key parameters of the bitstream.

H.264 defines three Profiles: *Baseline Profile*, *Main Profile* and *Extended Profile* [53]. Each supports a particular set of coding functions and each speci-

Figure 4.3: H.264 Baseline profile, Main profile and Extended profile [2].

fies the requirements of an encoder or a decoder that complies with the corresponding profile. Each Profile has sufficient flexibility to support a wide range of applications. Figure 4.3 shows the relationship between the three profiles and the coding tools supported by the standard. It is clear that the Baseline profile is a subset of the Extended profile, but not of the Main profile.

Performance limits for CODECs are defined by a set of levels, each placing limits on parameters such as sample processing rate, picture size, coded bitrate and memory requirements.

4.3.2 Video Format

The H.264 standard supports coding and decoding of 4:2:0 raw video [53]. In the default sampling format, chroma (Cb and Cr) samples are aligned horizontally

Figure 4.4: Multi-picture motion compensation for a B-picture [3].

with every second luma sample and are located vertically between two luma samples.

4.3.3 Video Coding Layer and Network Abstract Layer

The H.264 standard makes a distinction between a Video Coding Layer (VCL) and a Network Abstraction Layer (NAL) [3], [54]. The VCL is specified to efficiently represent the content of the video data. The NAL is specified to format the coded data and provide header information appropriately for transmission or storage. The purpose of separately specifying the VCL and NAL is to distinguish between coding-specific features (at the VCL) and transport-specific features (at the NAL). The output of the encoding process is VCL data which are mapped to NAL units prior to transmission or storage.

4.3.4 Reference Pictures

Multi-picture motion-compensated prediction is supported in the H.264 standard [3], that is, N ($1 \leq N \leq 5$) previously coded pictures can be used as a reference for motion-compensated prediction. Figure 4.4 illustrates the concept for one B-picture.

4.3.5 Slices

A video picture can be coded as one or more slices, each containing a finite number of macroblocks (MBs) from one MB per slice to the total number of MBs in a picture. There are five types of coded slices and a coded picture may be composed of different types of slices: *I* (*Intra-predicted*), *P* (*Predicted*), *B* (*Bi-predictive*), *SI* (*Switching Intra-predicted*) and *SP* (*Switching Predicted*) [55].

4.4 Baseline Profile

The Baseline Profile supports coded sequence containing I- and P-slices, as shown in Figure 4.3, with the exception of B-slices which are supported in the Main Profile and the Extended Profile. After prediction, the residual data for each macroblock is transformed using a 4×4 integer DCT transform and quantized. Quantized transform coefficients are reordered and then entropy coded. In the Baseline Profile, the entropy coding method is a context-adaptive variable length coding scheme (CAVLC). Quantized coefficients are scaled, inverse transformed, reconstructed and filtered with a de-blocking filter before being stored for possible use in reference pictures for further intra- and inter-coded MBs.

4.4.1 Intra Coding

In all slice-coding types, two types of I-slice coding are supported, namely, INTRA- 4×4 and INTRA- 16×16 [2], for the luminance sample. INTRA- 4×4 mode divides a macroblock into 16 4×4 sub-blocks and encodes each separately. It is suitable for parts of a frame with significant details. INTRA- 16×16 mode, on the contrary, encodes the whole macroblock once and for all and this mode is more suitable for smooth areas.

M	A	B	C	D	E	F	G	H
I	a	b	c	d				
J	e	f	g	h				
K	i	j	k	l				
L	m	n	o	p				

Figure 4.5: Labelling of Intra-prediction samples (4×4) [2].

In contrast to H.263++ and MPEG-4, where the prediction is conducted in the transform domain, H.264 standard is unique in conducting the prediction in spatial domain by referring to neighboring samples of already coded blocks. As shown in Figure 4.5, the samples above the current 4×4 block to be predicted (the grey zone) and to the left have previously been encoded and reconstructed, and are therefore available to form an intra 4×4 prediction reference.

The H.264 standard supports 9 optional prediction directions for 4×4 prediction modes (Figure 4.6), 4 for 16×16 prediction modes (Figure 4.7). The arrows in figures indicate the direction of prediction in each mode. For modes 3-8 in Figure 4.6, the predicted samples are formed from a weighted average of the prediction samples A-Q. The encoder may select the prediction mode for each block that minimizes the residual between the predicted block and the original block to be encoded.

4.4.2 Inter Coding

4.4.2.1 Motion Compensation in P-slices

In addition to the Intra coding types, various predictive or motion-compensated coding types are specified for P-slices in the Baseline Profile.

Significant differences from earlier standards include the support for a range of block sizes (from 16×16 down to 4×4) for prediction and fine sub-sample motion vectors (quarter-sample resolution in the luma component).

Figure 4.6: 4 × 4 luma Intra-prediction modes [2].

Figure 4.7: 16 × 16 luma Intra-prediction modes [2].

Figure 4.8: Macroblock partition modes: 16×16 , 8×16 , 16×8 , 8×8 [3]. Figure 4.9: Sub-macroblock partition modes: 8×8 , 4×8 , 8×4 , 4×4 [3].

The luma component of each macroblock may be split up and motion compensated in four ways as shown in Figure 4.8. In cases where the 8×8 partition mode is chosen, each of the four 8×8 sub-macroblocks within the macroblock can be further split up in four ways as shown in Figure 4.9. This method of partitioning macroblocks into motion compensated sub-blocks of varying size is known as *tree-structured motion compensation* [2], [3], [56]. The resolution of each chroma component in the Macroblock is half of the luminance component.

A separate motion vector is required for each macroblock partition or sub-macroblock partition. If a large partition mode (16×16) is chosen, motion vector(s) and the type of partition are coded using a small number of bits. However, the motion compensated residual data may contain a significant amount of energy with high detail. In cases where a smaller partition mode (4×4) is selected, it may give a lower-energy residual data, but requires a larger number of bits to signal the motion vectors and the partition mode choice(s). Therefore, the choice of partition mode significantly impacts the compression performance. In general, a large partition mode is appropriate for homogeneous areas of the

frame and a small partition mode may be beneficial for detailed areas.

In addition to the motion-compensated macroblock partition modes described above, a P-slice macroblock may also be coded in the *SKIP* mode [3]. If a macroblock has motion characteristics that allow its motion to be effectively predicted from the motion of neighboring macroblocks, and it contains no non-zero quantized transform coefficients, then it is flagged as skipped. For this mode, neither a quantized prediction residual signal, nor a motion vector or reference index, has to be coded and transmitted.

Moreover, motion vectors are represented in quarter-pixel resolution in the H.264 standard and prediction values at quarter-pixel locations are interpolated from integer samples. As a result, the inter prediction can be more efficient than previous standards.

Figure 4.10 shows two adjacent luma frames and the residual frame without motion compensation. For each macroblock, the H.264 encoder selects the 'best' partition mode that minimizes the amount of information to be coded and sent. In areas where there is little change between the frames (residual appears grey), a 16×16 partition mode is chosen and in areas of detailed motion (residual appears black or white), small partitions are more efficient.

4.4.2.2 Motion compensation in B-slices

In comparison to prior video coding standards, the concept of B-slices is generalized in the Main Profile and Extended Profile of H.264. Unlike in prior standards, other pictures can reference B-pictures for motion-compensated prediction. Thus, the substantial difference between P-slices and B-slices is that: a) in B-slices, two motion vectors, representing two estimates of the motion per block are allowed for temporal prediction, and they can be from any reference picture in future or past in display order; and b) a weighted average of the pixel values in the reference pictures is used as the predictor for each sample [3], [57].

Figure 4.10: Residual (without motion compensation(MC)) showing choice of block sizes [2].

Figure 4.11: Performance of the deblocking filter. *Left:* without the filter. *Right:* with the filter

B-slices also have a special prediction mode - *Direct mode*. In this mode, no prediction error signal is coded and transmitted. It is also referred to as *B-slice SKIP mode* and can be coded very efficiently, in a similar way to the SKIP mode in P-slices [58].

4.4.3 In-Loop Deblocking Filter

"Blocking artifacts" is the one of the most prominent artifacts with the present block-based video coding schemes. For this reason, H.264 defines an adaptive deblocking filter [59] within the prediction loop to remove blocking-edge distortion, without much affecting the sharpness of the content. Consequently, the subjective quality is significantly improved, as illustrated in Figure 4.11. The coding bit rate is reduced with typically 5 – 10% while producing the same subjective quality as the non-filtered video [56].

4.4.4 Transform, Scaling and Quantization

Similar to previous coding standards, H.264 also utilizes block-based transform coding for the prediction residual. However, H.264 is unique in that it employs a purely *integer Discrete Cosine Transformation* (DCT) [60] instead of

a 8×8 DCT. This transform operates on 4×4 blocks of residual data after motion-compensated prediction or intra-prediction. This transform is based on the DCT but there are some fundamental differences [2]. A more specific description of the integer DCT will be presented in the next chapter.

For the transform coefficients, the H.264 standard uses perceptual-based scalar quantization [3], [58]. The encoder can specify a customized scaling quantization factor, for each transform block and for intra- and inter-prediction, separately. This allows tuning of the quantization fidelity according to a model of sensitivity of the human visual system to different types of residual error. It typically does not improve objective fidelity as measured by PSNR, but it does improve subjective fidelity, which is more important. The quantized transform coefficients of a block are generally scanned in a zigzag fashion and transmitted using entropy coding methods.

4.4.5 Entropy Coding

Entropy coding in previous standards such as MPEG-1, 2, 4, H.261, and H.263 is based on fixed tables of variable length codes (VLCs) [61]. These standards define sets of codewords based on the probability distributions of generic videos instead of exact Huffman code for the video sequences. However, the H.264 standard uses different VLCs in order to match a symbol to a code based on the context characteristics. The H.264 standard specifies two types of entropy coding: Context-based Adaptive Binary Arithmetic Coding (CABAC) and Context-based Adaptive Variable Length Coding (CAVLC) [2], [62]. As compared to the older version of video coding standards, CABAC is a new entropy coding scheme in addition to CAVLC. While CAVLC is easier for implementation, CABAC can typically provide 5%–15% more bit rate saving compared with CAVLC. Also, CABAC is employed in Main and High profiles,

CABAC has more coding efficiency but higher complexity compared to CAVLC.

4.4.6 Mode Decision with Lagrangian Optimization

From the above description, the H.264 standard provides far more coding options than previous standards. Thus, the optimization task is to choose, the most efficient coded representation (partition modes, prediction methods, motion vectors, quantization levels, etc.), for each picture region. As mentioned above, various partition modes are supported for the motion-compensated prediction. In the H.264 standard, mode selection can be decided using a Rate-Distortion-cost scheme. The scheme takes into consideration the distortion factor and the rate of the compressed stream.

To be more specific, to solve this optimization problem, the Lagrangian optimization technique [63], [64], [65] is adopted in the H.264/AVC due to its effectiveness and simplicity. Given the macroblock quantizer parameter QP and the Lagrangian multiplier λ , the optimal mode for a block $\mathbf{B}$ is found by minimizing the Lagrangian function within the constrained R and minimized D

$$J(\mathbf{B}, o|QP, \lambda) = D(\mathbf{B}, o|QP) + \lambda R(\mathbf{B}, o|QP, \lambda), \quad (4.1)$$

where the Lagrangian multiplier λ is related to the macroblock quantizer parameter QP by

$$\lambda = 5 \frac{QP + 5}{34 - QP} \exp\left(\frac{QP}{10}\right), \quad (4.2)$$

and the block mode o indicates the selection from the set of potential coding modes and it is varied over all possible coding modes available for a particular frame type. The distortion D is the sum of the squared differences between the original block and its reconstruction. It also takes into account the distortion in the chroma components. The number of bits R is associated with the mode o and QP , the bits for macroblock header, motion information, and all integer

transform blocks. Detailed discussion of this relationship can be found in [66] and [67].

For completeness, we list modes used in I-slices, P-slices and B-slices as follows

- I-slices: INTRA- 4×4 , INTRA- 16×16 and their prediction modes (Figure 4.6 and 4.7);
- P-slices: INTER- 16×16 , INTER- 16×8 , INTER- 8×16 , INTER- 8×8 , INTER- 8×4 , INTER- 4×8 , INTER- 4×4 , and SKIP (Figure 4.8 and 4.9);
- B-slices: INTER- 16×16 , INTER- 16×8 , INTER- 8×16 , INTER- 8×8 , INTER- 8×4 , INTER- 4×8 , INTER- 4×4 , and DIRECT.

This technique has gained importance due to its effectiveness, conceptual simplicity, and its ability to effectively evaluate a large number of possible coding choices in an optimized fashion. As a result, the computation time to test all potential modes may become the limiting factor on performance, rather than the capabilities of the syntax itself.

4.5 Applications

The increased compression efficiency of H.264/AVC, as well as simple syntax specification, adaptation to delay constraints, error resilience together with improved network friendliness, offers new application areas and business opportunities, such as the display of video on mobile devices as shown in [56], [68], [69]. It is now possible, to transmit a video signal at about 1Mbit/s with TV (PAL¹) quality, which enables streaming over xDSL² connections. Other interesting

¹PAL is the short for Phase Alternating Line, the dominant television standard in Europe.

²xDSL refers collectively to all types of digital subscriber lines, the two main categories being Asymmetric DSL (ADSL) and Symmetric DSL (SDSL). Two other types of xDSL technologies are High-data-rate DSL (HDSL) and Very high DSL (VDSL).

business areas are TV transmission over satellite, high-data-rate (HD) transmission and storage. Also in the field of mobile communication, H.264/AVC will play an important role because the compression efficiency will be doubled in comparison to the coding schemes currently specified by 3GPP for streaming, i.e. H.263 Baseline, H.263+ and MPEG-4 Simple Profile. Video transmission for mobile terminals will be a major application for H.264/AVC in the coming 3G systems. The display of video on mobile devices opens the road to the following new applications: conversational services for video telephony and video conferencing, live or pre-coded video streaming services, satellite/cable/DVD (0.5-8 Mbps) entertainments, digital cinema application, 3G video in multimedia messaging services, and so on.

Chapter 5

PROPOSED ROBUST WATERMARKING OF DIGITAL IMAGES

5.1 Introduction

In this chapter a robust watermarking scheme for copyright protection is proposed for still images, in order to test the performance of the 4×4 integer DCT in watermarking schemes. The acceptable experimental results leads us to a robust video watermarking scheme for the H.264/AVC standard in Chapter 6.

The rest of this chapter is organized as follows: Section 5.2 describes the 4×4 integer DCT of the H.264/AVC standard. Section 5.3 proposes our image-in-image algorithm for copyright protection. Section 5.4 develops the post-processing techniques to improve the performance of the image watermarking algorithm. Section 5.5 analyzes the experimental results, followed by the summary in Section 5.6.

5.2 The Approximate Integer DCT

The discrete cosine transform (DCT) has found wide applications in image and video processing and many other fields. It has become the heart of many international standards such as JPEG, H.263 and the MPEG family. The DCT is a robust approximation of the optimal Karhunen-Loève transform (KLT) for a first-order Markov source with large correlation coefficients. It has satisfactory performance in terms of energy compaction capability. Many fast DCT algorithms with efficient hardware and software implementations have been proposed [70], [71].

However, the floating-point multiplications of the fast DCTs are inevitable for implementing such transforms, which limit them from being practically used in mobile devices and lossless compression. In mobile devices, the power consumption used for computation, especially for floating-point multiplications, cannot be neglected. Since there will be errors when quantizing the transforming coefficients, it is impossible to use them for lossless compression [72]. Therefore, it is not surprising that lossless coding schemes are hardly based on the DCTs. Thus, transforms without floating-point multiplications or integer transforms have received increasing attention during recent years.

In this section, the approximate integer DCT with low-complexity used in the H.264 standard will be introduced. First consider $\mathbf{Y}$ the 4×4 traditional DCT of an input 4×4 array $\mathbf{y}$, defined as

$$\mathbf{Y} = \mathbf{A}\mathbf{y}\mathbf{A}^T, \quad (5.1)$$

$$\text{where } \mathbf{A} = \begin{pmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{pmatrix} \quad \text{with } a = 1/2, \quad b = \sqrt{1/2} \cos(\pi/8), \quad \text{and}$$

$$c = \sqrt{1/2} \cos(3\pi/8).$$

From Equation (5.1) the matrix multiplication can be factorized to the following form

$$\mathbf{Y} = \mathbf{C}\mathbf{y}\mathbf{C}^T \otimes \mathbf{E}, \quad (5.2)$$

where $\mathbf{C} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 1 & d & -d & -1 \\ 1 & -1 & -1 & 1 \\ d & -1 & 1 & -d \end{pmatrix}$ with $d=c/b$ (approximately 0.414), $\mathbf{E}$ is a

matrix of scaling factors defined as $\mathbf{E} = \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix}$, and the symbol

$\otimes$ indicates the point by point multiplication.

In Equation (5.2), $\mathbf{C}\mathbf{y}\mathbf{C}^T$ is a "core" 2-D transform. To simplify the implementation of the core transform, d is approximated by 0.5. To ensure that the transform remains orthogonal, b also needs to be modified so that

$$a = 1/2, \quad b = \sqrt{2/5}, \quad d = 1/2.$$

The 2nd and 4th rows of matrix $\mathbf{C}$ are scaled by a factor of 2 and the post-scaling matrix $\mathbf{E}$ is scaled down for compensation. This avoids multiplications by 1/2 in the "core" transform $\mathbf{C}\mathbf{y}\mathbf{C}^T$, which would result in loss of accuracy using integer arithmetic. The final forward 4×4 integer DCT is defined by

$$\mathbf{Y}_{\text{for}} = \mathbf{C}_{\text{for}}\mathbf{y}\mathbf{C}_{\text{for}}^T \otimes \mathbf{E}_{\text{for}}, \quad (5.3)$$

where $\mathbf{C}_{\text{for}}$ and $\mathbf{E}_{\text{for}}$ are defined by

$$\mathbf{C}_{\text{for}} = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{pmatrix}, \quad \mathbf{E}_{\text{for}} = \begin{pmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{pmatrix}.$$

The forward 4×4 integer DCT given by Equation (5.3) is an approximation to the 4×4 traditional DCT. In the context of the H.264 CODEC, it has almost the identical compression performance to DCT and it has a number of important advantages. The "core" part of the transform can be carried out with integer arithmetic using only additions, subtractions and shifts as the implementation of a multiplication by 2. The dynamic range of the transform operations is such that a 16-bit arithmetic may be used throughout without any overflow risk, as long as the inputs are in the range $[-255, 255]$.

The inverse transform of the integer DCT is given by

$$\mathbf{y} = \mathbf{C}_{\text{inv}}^T (\mathbf{Y}_{\text{for}} \otimes \mathbf{E}_{\text{inv}}) \mathbf{C}_{\text{inv}}, \quad (5.4)$$

where $\mathbf{C}_{\text{inv}}$ and $\mathbf{E}_{\text{inv}}$ are defined by

$$\mathbf{C}_{\text{inv}} = \begin{pmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{pmatrix}, \quad \text{and} \quad \mathbf{E}_{\text{inv}} = \begin{pmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{pmatrix}.$$

The matrix $\mathbf{Y}_{\text{for}}$ is pre-scaled by multiplying each coefficient with the appropriate weighting factor from the matrix $\mathbf{E}_{\text{inv}}$. Note the factors $\pm 1/2$ in the matrices $\mathbf{C}_{\text{inv}}$ and its transposed matrix; these can be implemented by a right-shift without a significant loss of accuracy because the coefficients $\mathbf{Y}_{\text{for}}$ are pre-scaled.

In our investigation, we first propose an image watermarking scheme. For

transform performance comparison, we also apply the other approximate transforms: Cham's 8-bit ICT, Liang et al.'s 4/8-bit BinDCT, together with the Hadamard transform applied in [13].

Cham and Yip [73], [74] developed a series of 8×8 Integer Cosine Transforms (ICTs) based on the principle of dyadic symmetry. They claimed that the implementation complexity of an ICT depends on the number of bits required to represent the magnitude of its kernel components, but Cham's integer transform still requires real number multiplication for the inverse transform.

Liang and Tran [75], [76] presented the design, implementation and application of several families of fast approximations of DCT with less multiplication, and the lifting scheme, named the BinDCT (with multiple sizes 4×4 , 8×8 and 16×16). All the lifting parameters in their design were chosen to be dyadic rational, enabling fast implementations with only shift and addition operations. The elimination of the floating-point and fixed-point multiplications enables the BinDCT to be implemented with much less computation than other fast algorithms. They once proposed their new transforms to the H.264 Draft [77], [78].

5.3 Proposed Image-in-Image Method

For the image-in-image watermarking algorithm proposed in [13], both the watermark image and the host image were transformed with the block-wise Hadamard transform of size 8×8 . A pseudo-random selection using the m-sequence was applied to find the relevant sub-blocks for embedding. Extracted features (edges and textures) adaptively determined the adaptive watermark gain. Two approaches were post-processed to increase the robustness against scaling and rotation attacks. We adapted this watermarking algorithm by using a different transform and proposed more advanced post-processing techniques.

5.3.1 Embedding Approach

Next the proposed algorithm is described step by step, and the embedding approach is illustrated in Figure 5.1.

5.3.1.1 Watermark Pattern

The watermark pattern $\mathbf{p}$ of size $M \times M$ is first decomposed into non-overlapping blocks of size 4×4 , and the l th sub-block is denoted by $\mathbf{p}_l$,

$$\mathbf{p}_l = \{p_l\{i, j\} \quad 0 \leq i, j \leq 3\}, \quad (5.5)$$

with $l = 0, 1, \dots, (M/4)^2 - 1$.

Then every watermark sub-block $\mathbf{p}_l$ are transformed into integer DCT coefficients as follows

$$\mathbf{P}_l = \text{IntDCT}\{\mathbf{p}_l\} = \{P_l\{u, v\} \quad 0 \leq u, v \leq 3\}. \quad (5.6)$$

All the frequency coefficients $P_l(u, v)$ within the l th transformed watermark sub-block are scanned in a zigzag fashion and arranged in a one-dimensional sequence, denoted by $\mathbf{w}_l = \{w_l(g), g = 0, \dots, 15\}$, where g is the zigzag position index.

Then, the 1-D watermark sequence $\mathbf{w}$ can be obtained as follows

$$\mathbf{w} = \bigcup_{l=0}^{(M/4)^2-1} \{\mathbf{w}_l\} = \{w(f), \quad f = 0, 1, \dots, (M/4)^2 - 1\}. \quad (5.7)$$

5.3.1.2 Cover Image

Consider $\mathbf{x}$ the original cover image of size $N \times N$ ($N > M$). First the *Canny Edge Detection* is applied to the whole cover image $\mathbf{x}$ to compute the adaptive watermark gain for embedding. Canny edge detection is chosen because it

Figure 5.1: Watermark embedding processes of the image-in-image algorithm.

60 Chapter 5. PROPOSED ROBUST WATERMARKING OF DIGITAL IMAGES

is able to detect weak edges by using two different thresholds (which can be decided by the user) as follows

$$\mathbf{e} = \text{edge}(\mathbf{x}, \text{'canny'}, [\text{Thresh}_1, \text{Thresh}_2]), \quad (5.8)$$

where $\mathbf{e}$ is the edge matrix of size $N \times N$, function edge is the traditional edge detection method built in Matlab, 'canny' is the edge detection parameter, and Thresh_1 and Thresh_2 specify low and high sensitivity thresholds to detect strong and weak edges, respectively. This method detects the weak edges only if they are connected to strong edges. This method is therefore less likely than the others to be fooled by noise, and more likely to detect true weak edges. In our experiments, Thresh_1 and Thresh_2 are set to 0.15 and 0.2, respectively.

Figure 5.2 shows the detected Canny edges of two test images. In Figures 5.2 (b) and (d), the white lines represent coarse texture areas with strong edges with the pixel value equal to 1, and the black areas represent smooth areas with the pixel value equal to 0.

Secondly, decompose the original cover image $\mathbf{x}$ and the edge image $\mathbf{e}$ into non-overlapping blocks of size 4×4 , and the k th sub-block is denoted by

$$\mathbf{x}_k = \{x_k\{i, j\} \mid 0 \leq i, j \leq 3\}, \quad (5.9)$$

and

$$\mathbf{e}_k = \{e_k\{i, j\} \mid 0 \leq i, j \leq 3\}, \quad (5.10)$$

with $k \in \mathcal{K} = \{0, 1, \dots, (N/4)^2 - 1\}$.

Counting the number of edge points in each sub-block, we obtain the adaptive visual mask for watermark embedding, γ_k as follows

$$\gamma_k = \sum_{0 \leq i, j \leq 3} e_k(i, j), \quad k \in \mathcal{K}. \quad (5.11)$$

(a) *Lena*

(b) Canny edges of *Lena*

(c) *Baboon*

(d) Canny edges of *Baboon*

Figure 5.2: Original images and the corresponding Canny edges.

This mask γ_k is determined by the outstanding edges in the image sub-block. A small value in this mask indicates that the corresponding sub-block is smoothly textured. A larger value indicates that the sub-block contains outstanding edges [13]. In this way, the visual quality of the watermarked image can be improved.

We adaptively control the watermark gain by using the edge-detecting mask shown in the following equation

$$\alpha_k = \beta \cdot \gamma_k, \quad (5.12)$$

where β is the scaling factor, and k indicate the positions of the cover image sub-blocks.

Thirdly, a pseudo-noise (PN) sequence pseudo-randomly selects the sub-blocks for watermark insertion among the original host image sub-blocks $\mathbf{x}_k$, so that the watermark information bits are spread around the host image. This approach can solve the problem of regional processing attacks (such as cropping). The most important class of PN sequence is the binary maximal-length sequence, or m-sequence. It is generated by the linear feedback shift registers (LFSR) and boolean Exclusive-OR gates (XOR).

The generator polynomial $g(z)$ of the m-sequence is introduced as follows

$$g(z) = 1 + g_1 z + g_2 z^2 + \cdots + g_{q-1} z^{q-1} + z^q, \quad (5.13)$$

where z is the variable for the polynomial presentation, $g_1, g_2, \dots$, and g_{q-1} are either 0 or 1, and in binary representation the generator polynomial $g(z)$ is $[1 \ g_1 \ g_2 \ \dots \ g_{q-1} \ 1]$ where $q + 1$ is the length, and q must satisfy the following condition

$$2^{q+2} - 1 > N^2/16. \quad (5.14)$$

In the example, when q is 4, and only g_1 is equal to 1, thus $g(z)$ is equal to $1 + z + z^4$, or $[1\ 1\ 0\ 0\ 1]_{\text{binary}}$, or $[25]_{\text{decimal}}$.

In our watermarking algorithm, the length of the m-sequence is decided to be the length of watermark sequence M^2 , and thus M^2 sub-blocks of the cover image will be chosen. Consider $\mathcal{M}$ the set of all the random integer numbers generated from the m-sequence as follows

$$\mathcal{M} = \{m_f, f = 0, \dots, M^2 - 1\}, \quad (5.15)$$

where m_0 is the starting value of the m-sequence pre-generated by an integer random seed, and $m_f \in \text{mathcal{K}}$ is the decimal number generated from $g(z)$, and it is also the index of the cover image sub-block chosen by the m-sequence.

Fourthly, every cover image sub-block $\mathbf{x}_{m_f}$ chosen by the m-sequence are transformed into integer DCT coefficients as follows

$$\mathbf{X}_{m_f} = \text{IntDCT}\{\mathbf{x}_{m_f}\} = \{X_{m_f}(u, v) \mid 0 \leq u, v \leq 3\}. \quad (5.16)$$

Next, within every transformed sub-block of the host image, only four middle frequency AC components ($1 \leq u_o, v_o \leq 2$) are used for embedding. The original image AC components $X_{m_f}(u_o, v_o)$ are replaced by $\tilde{X}_{m_f}(u_o, v_o)$

$$X_{m_f}(u_o, v_o) \leftarrow \tilde{X}_{m_f}(u_o, v_o) = \alpha_{m_f} \cdot w(f), \quad (5.17)$$

where $X_{m_f}(u_o, v_o)$ and $\tilde{X}_{m_f}(u_o, v_o)$ are the AC components of the original image sub-blocks before and after inserting watermark, respectively, and $w(f)$ is the f th watermark coefficient, where $f=0, 1, \dots, M^2 - 1$, and α_{m_f} is the adaptive watermark gain factor of the cover image sub-block chosen by the m-sequence.

The inverse integer DCT is then applied to obtain the luminance value of

the watermarked image sub-block $\tilde{\mathbf{x}}_{m_f}$ as follows

$$\tilde{\mathbf{x}}_{m_f} = \text{IntDCT}^{-1}\{\tilde{\mathbf{X}}_{m_f}\}. \quad (5.18)$$

The results of image-in-image watermark embedding process are shown in Figure 5.3 (original images *Lena* and *Baboon* marked with watermark image *DMT*). The PSNR of the watermarked images are as high as 30.9dB and 31.7dB, respectively.

A key file containing a number of relevant data is generated and stored. These data includes the size of the watermarked image and the size of the watermark pattern, together with relevant data of the m-sequence for pseudo-random selection.

5.3.2 Retrieval Approach

The watermarked image is stored or transmitted and would possibly be distorted by channel noise and external attacks. Thus, $\hat{\mathbf{x}}$ is used to denote the probably attacked host image with hidden watermark. At the receiver, the watermark should be detected and extracted to identify the ownership.

The data stored in the key file during embedding is needed in the blind retrieval process without the original host image. The received watermarked image $\hat{\mathbf{x}}$ is decomposed into 4×4 sub-blocks. The same m-sequence as the one in the embedding process is constructed to maintain the synchronization between embedding and retrieval. The possible embedded sub-blocks are selected by the reconstructed m-sequence and transformed into the integer DCT domain.

Take one of the integer DCT sub-blocks as an example. One watermark component $\hat{w}(f)$ is extracted from the fixed middle frequency component $\hat{x}_{m_f}(u_o, v_o)$

DMT

(a) Embedded watermark

(b) Original *Lena*

(c) Watermarked *Lena*
(PSNR=30.9dB)

(d) Original *Baboon*

(e) Watermarked *Baboon*
(PSNR=31.7dB)

Figure 5.3: The watermark pattern (16×16 DMT), the original and watermarked images (512×512 *Lena* and *Baboon*).

Figure 5.4: Watermark extraction processes of the image-in-image algorithm.

$(1 \leq u_o, v_o \leq 2)$ as follows

$$\hat{w}(f) = \hat{x}_{m_f}(u_o, v_o) / \hat{\alpha}_{m_f}, \quad (5.19)$$

where $\hat{\alpha}_{m_f}$ is the watermark gain factor obtained by the extracted feature of the watermarked image.

Inverse zigzag scanning and the inverse 4×4 integer DCT are applied to obtain the extracted watermark image, $\hat{\mathbf{p}}$.

5.4 Post-processing Techniques

Two post-processing techniques were proposed in [13] to increase the robustness against scaling and rotation geometric attacks. The geometrical transformation was estimated and the reverse transforms were applied before recovering the watermark. In the case of scaling attacks, the attacked image size, was resized to the original size. In the case of rotation attacks, in order to retrieve the information bits correctly, a post-processing method was proposed to detect and rotate the attacked image back to its original orientation. Log-Polar Mapping (LPM) of the image was used to trace the rotation angle.

Since experimental results showed that these techniques in [13] did perform well against these geometric attacks, these two post-processing methods are still used in our algorithm. In the following two paragraphs, two more post-

processing techniques are proposed to further improve the robustness of our watermarking scheme.

In the case of additive noise attacks, Gaussian white noise is added to the watermarked image pixel by pixel. We developed a post-processing method to perform 2-D adaptive noise-removal filtering. The method first detects whether the input intensity image is degraded by constant power additive noise. It is based on statistics estimated from a local neighborhood of each pixel. A threshold has been set for the detection with the value of 0.0013, based on the experiments. If additive noise was detected, it would be removed from the degraded image with a low pass filter, according to the statistics estimated from a local neighborhood of each pixel.

After undergoing cropping attacks by factor of 25%, 50% or 75%, the central part of the image is remained. Obviously, the attacked image is no longer the same size as the original un-attacked image. If the change of image size is detected, the cropped image will be scaled to the original size by interpolation. The position of the hidden bits cannot be located correctly because the synchronization between embedding and detection has been destroyed by scaling. Therefore, before scaling, a blank image of the original size is constructed and its central part is replaced by the cropped image as shown in Figure 5.5. Since the entire watermark bits are dispersed into the whole image by the m-sequence, our algorithm shows the satisfactory robustness to cropping attacks.

5.5 Experiments and Results

5.5.1 Performance Measurements

Objective image quality measures play important roles in various image processing applications. There are basically two classes of objective image visual quality or distortion assessment approaches. The first are mathematically de-

Figure 5.5: Post-processing method again cropping.

finer measures such as the widely used Peak Signal to Noise Ratio (PSNR). The second class of measurement methods considers Human Visual System (HVS) characteristics in an attempt to incorporate perceptual quality measures [79].

The PSNR value (compared between the watermarked host image and the original host image) in Table 5.1 is computed by

$$PSNR = 10 \log \frac{N^2 \max x(i, j)^2}{\sum_{i=0}^{N-1} \sum_{j=0}^{N-1} [x(i, j) - \tilde{x}(i, j)]^2}, \quad (5.20)$$

where i and j are the indexes of the image pixels, $x(i, j)$ and $\tilde{x}(i, j)$ are the gray levels of the original and the processed pixels, respectively, and $\max x(i, j)$ for an intensity image has the value of 255. One important prerequisite is that the watermarked image and the original image must have the same image size.

As in most schemes such as [13], to verify the presence of the watermark, the similarity between the original watermark $\mathbf{p}$ with size $M \times M$ and the extracted watermark $\hat{\mathbf{p}}$ from the possibly attacked image $\hat{\mathbf{x}}$ is measured. The similarity measure is given by the normalized correlation coefficient as the following formula

$$Corr(\mathbf{p}, \hat{\mathbf{p}}) = \frac{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [p(i, j) \otimes \hat{p}(i, j)]}{\sqrt{\sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [p(i, j) \otimes p(i, j)] \sum_{i=0}^{M-1} \sum_{j=0}^{M-1} [\hat{p}(i, j) \otimes \hat{p}(i, j)]}}. \quad (5.21)$$

where i and j are the vertical and horizontal indices of $\mathbf{p}$ and $\hat{\mathbf{p}}$, and the

PSNR (dB)	8x8 Transforms			4x4 Transforms		
	DCT	ICT	BinDCT	DCT	IntDCT	BinDCT
Marked <i>Lena</i>	30.67	30.84	30.78	31.49	30.86	31.18
Marked <i>Baboon</i>	26.40	26.62	26.42	31.42	31.73	31.72

Table 5.1: Visual quality comparison of watermarked image among different transforms (image *Lena* and *Baboon* embedded with *DMT*).

symbol $\otimes$ indicates the point by point multiplication. The correlation result with the value 1 means that the extracted watermark is totally the same as the original one, while 0 means totally different and -1 means totally opposite.

The main tradeoff of watermarking is between the fidelity of the cover image and the robustness of the watermark. The greater PSNR value of the watermarked image shows a better visual quality. The greater correlation value between the extracted watermark and original embedded watermark represents higher robustness. If the watermark gain α_k is set to a large enough value, the watermark would obviously have higher robustness against the common signal processing and geometric attacks. However, this would introduce visible artifacts to the watermarked image.

5.5.2 Fidelity after Watermarking

Watermarked images *Lena* and *Baboon* are shown in Figure 5.3. The PSNR between the original and the watermarked images are as high as 30.9dB and 31.7dB, respectively. Furthermore, Table 5.1 concludes that the visual quality of the 4×4 transformed images are a little better than the 8×8 transformed images. This is mainly due to the fact that the 4×4 transforms disperse the energy of the watermark. Take an 8×8 sub-block from *Lena* as an example.

The experiment showed that all four DC coefficients of the 4×4 integer DCT matrices are approximately half of the value of the only one DC coefficient of the 8×8 DCT matrix. When applying the "replacing" operation during embedding, smaller values would introduce less perceptual artifacts to the cover image. However, the 4×4 transforms would reduce the robustness of the embedded watermarks.

5.5.3 Results of Robustness Tests

Figure 5.6 shows the extracted watermark from an un-attacked image. Without any attack, the extracted watermark is highly correlated to the original watermark.

Simulation results for some geometric distortion and common signal processing attacks are shown in Table 5.2. The table shows the correlation coefficients between the original watermark image and the extracted one under different attacks. As shown in Table 5.2, our scheme can resist JPEG compression for even the lowest quality factor, 10. The JPEG compression quantization step size used in StirMark ranges from 10 to 100.

Without the post-processing techniques stated in Section 5.4, our scheme cannot survive such geometric attacks as scaling, central cropping, and rotation. After applying the post-processing techniques, our scheme shows satisfactory robustness to these common attacks.

Some signal processing attacks used in StirMark 4.0 are detailed below. The 3×3 Gaussian filter matrix is

$$\begin{pmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{pmatrix},$$

A 16x16 grid of black and white pixels forming the letters 'DMT' in a bold, sans-serif font.

(a) original 16x16 watermark

A 16x16 grid of black and white pixels forming the letters 'DMT' in a bold, sans-serif font, identical to the original watermark.

(b) 16x16 watermark extracted from un-attacked *Lena*

(normalized correlation=0.98)

A 16x16 grid of black and white pixels forming the letters 'DMT' in a bold, sans-serif font, identical to the original watermark.

(c) 16x16 watermark extracted from un-attacked *Baboon*

(normalized correlation=0.99)

Figure 5.6: The original watermark and extracted watermark (cover image: *Lena*).

and the spatial sharpening filter matrix is

$$\begin{pmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{pmatrix}.$$

An additive noise attack was also applied to the watermarked image $\tilde{\mathbf{x}}$ by

$$\hat{\mathbf{x}} = \tilde{\mathbf{x}} \otimes (1 + \delta \cdot \theta), \quad (5.22)$$

where δ is a parameter that controls the strength of the additive noise, θ is noise with uniform distribution, zero mean, and unit variance, and $\hat{\mathbf{x}}$ is the attacked image. In our experiment, the additive noise is visible in *Lena* and *Baboon*, when the noise strength factor is greater than 1. When the noise strength factor is greater than 3, image quality would be terribly damaged. Even without the post-processing of de-noising, our scheme can still survive the additive uniform noise attack by the factor of 10. Our scheme also performs well under the other attacks in Stirmark 4.0, such as median filtering and Frequency Mode Laplacian Removal, shown in Table 5.2.

However, the algorithm did not survive certain Stirmark attacks. It failed for large cropping such as 25% because the remaining information was not sufficient for the decoder to extract enough hidden bits. Furthermore, the integer DCT watermarking technique was not very resistant to minor random geometric transforms, such as shearing and general linear transforms. Minor random geometric distortions could easily destroy the synchronization between the embedding and extraction processes. Since the attacks are random processes, it is difficult to perform a reverse process to reconstruct the synchronization.

Attacks	Lena	Baboon
Watermarked image without attacks	0.98	0.97
Frequency Mode Laplacian Removal factor=100	0.79	0.77
Frequency Mode Laplacian Removal factor=50	0.84	0.81
Frequency Mode Laplacian Removal factor=10	0.90	0.85
Median filter 2×2	0.91	0.80
Median filter 3×3	0.91	0.80
JPEG factor=30	0.98	0.96
JPEG factor=10	0.93	0.92
Gaussian filter 3×3	0.77	0.71
Rescaling factor=50%	0.99	0.97
Rescaling factor=90%	0.84	0.75
Rescaling factor=150%	0.89	0.84
Rescaling factor=200%	0.98	0.96
Sharpening 3×3	0.96	0.94
Additive uniform noise factor=0.1	0.98	0.97
Additive uniform noise factor=0.1	0.98	0.97
Additive uniform noise factor=0.1	0.72~0.98	0.73~0.97
Cropping>75%	0.69~0.86	0.70~0.87

Table 5.2: Correlation values of extracted watermarks under StirMark 4.0 attacks.

5.5.4 Performance Comparison

To compare the transform performance, different transforms were applied to the image watermarking scheme: 4×4 transforms - DCT, BinDCT, integer DCT and Hadamard; 8×8 transforms - DCT, BinDCT, ICT and Hadamard. When 8×8 transforms were used in the scheme, the size of sub-blocks changed to 8×8 . The same watermark image *DMT* was embedded in the host image *Lena*. The same feature extraction with one edge-detecting mask and post-processing techniques were applied.

The rough comparison results between DCTs and Hadamard transform were shown in Figure 5.7. The 4×4 integer DCT performed almost as well as 4×4 DCT, but the Hadamard transform did not show satisfactory robustness, especially when under JPEG Compression, 8×8 Hadamard performed even worse than the other 4×4 transforms. The correlation coefficients of DCTs and

(a) JPEG

(b) Gaussian Noise

(c) Line Removal

(d) Scaling

Legends:

- 1) Magenta dotted curves are for 8×8 transforms: DCT with ' $\diamond$ ', BinDCT with ' $\times$ ', and ICT with 'O';
- 2) Blue dashed curves are for 4×4 transforms: DCT with ' $\diamond$ ', BinDCT with ' $\times$ ', and integer DCT (denoted by *IntDCT*) with 'O';
- 3) Hadamard: The magenta dotted curve with ' ∇ ' for 4×4 HT, and the blue solid curve with ' ∇ ' for 8×8 HT.

Figure 5.7: Robustness comparison (correlation) under JPEG compression, Gaussian additive noise, frequency mode line removal and rescaling (cover image: *Lena*).

integer DCTs under JPEG were all greater than 0.9. The extracted watermark images were highly correlated to the original watermark. While only after the compression factor got greater than 90, did the correlation coefficient of 4×4 Hadamard become greater than 0.9. When the compression factor was in the range from 10 to 40, the correlation coefficient of 4×4 Hadamard was under 0.1. On average, The Hadamard transform is also less robust under the Gaussian additive noise, frequency mode Laplacian removal and rescaling since the fact that the integer DCT and Hadamard have different energy distribution within the sub-blocks.

Table 5.3 shows other comparison results under Gaussian Filtering, Sharpening and Median Filtering. The correlation coefficients were computed to evaluate the robustness of the different transform algorithms. From the table, the 4×4 Hadamard transform shows the least robustness among all the 4×4 transforms. While the 4×4 integer DCT sometimes performed best, for example when undergoing a 3×3 Gaussian filtering and a 3×3 sharpening. When undergoing a 3×3 Gaussian filtering and a 3×3 sharpening attack, all the 4×4 transforms except the 4×4 Hadamard transform has greater robustness than all the 8×8 transforms. However, the results were opposite when applying 4×4 and 5×5 median filtering. It is because the chosen coefficients for embedding were middle frequency coefficients. In the case of median filters, some middle coefficients were filtered, and some of the watermark information might get lost. Also the smaller size (4×4) decreased the robustness against these attacks.

For more specific comparison among all the other transforms except Hadamard, Figure 5.8 to 5.11 were generated from Figure 5.7. Robustness performance of the 4×4 integer DCT was compared to several other transforms (4×4 transforms: DCT and BinDCT; 8×8 transforms: DCT, BinDCT and ICT). It performed better than the 4×4 DCT under JPEG compression and scaling

(a) Robustness Comparison under JPEG Compression

(b) Mean of the Standard Correlation Values

Figure 5.8: Robustness comparison (correlation) under JPEG compression (cover image: *Lena*) (IntDCT stands for the 4×4 integer DCT).

(a) Robustness comparison under Gaussian Additive Noise

(b) Mean of Standard Correlation Values

Figure 5.9: Robustness comparison (correlation) under Gaussian additive noise (cover image: *Lena*) (IntDCT stands for the 4×4 integer DCT).

(a) Robustness Comparison under Line Removal

(b) Mean of Standard Correlation Values

Figure 5.10: Robustness comparison (correlation) under line removal (cover image: *Lena*) (IntDCT stands for the 4×4 integer DCT).

(a) Robustness Comparison under Rescaling

(b) Mean of Standard Correlation Values

Figure 5.11: Robustness comparison (correlation) under rescaling (cover image: *Lena*) (IntDCT stands for the 4×4 integer DCT).

80 Chapter 5. PROPOSED ROBUST WATERMARKING OF DIGITAL IMAGES

Transforms	Gaussian Filtering 3×3	Sharp- -ening 3×3	Median Filtering 2×2	Median Filtering 3×3	median Filtering 4×4	Median Filtering 5×5
4×4 IntDCT	0.77	0.96	0.84	0.84	0.08	0.08
4×4 BinDCT	0.76	0.95	0.91	0.91	0.07	0.07
4×4 DCT	0.76	0.95	0.85	0.85	0.07	0.07
4×4 Had	0.15	0.98	0.08	0.08	0.05	0.05
8×8 ICT	0.72	0.75	0.89	0.89	0.47	0.47
8×8 BinDCT	0.73	0.76	0.92	0.92	0.49	0.49
8×8 DCT	0.73	0.77	0.92	0.92	0.50	0.49
8×8 Had	0.71	0.98	0.83	0.83	0.02	0.02

Table 5.3: Performance comparisons (correlation) among different transform algorithms (cover image: *Lena*) (IntDCT stands for the 4×4 integer DCT, and Had for Hadamard).

attacks, with the correlation mean increase by 2.1% and 8.7%, respectively. Under frequency mode Laplacian removal and Gaussian additive noise, 4×4 integer DCT significantly outperformed the 4×4 DCT with correlation mean increase by 14.8% and 12.3%, respectively (when applied to the image *Lena*). It even has the closest performance to 8×8 DCT under JPEG compression. This shows that the integer DCT applied to H.264 is not only efficient for video coding, but also has a very good performance in watermarking techniques.

From Figures 5.8 to 5.11 and Table 5.3, it is easy to see that the 8×8 transform algorithms show greater robustness to some Stirmark attacks compared to the 4×4 transform algorithms. Take median filtering for example, the size of the median filters is 2×2 or 3×3 . Applying these filters to the watermarked image processed by 8×8 transforms will not affect the watermark much. However, this will destroy the watermark's robustness in the watermarked image processed by 4×4 transforms, because the size 4×4 is quite small and is very close to the size of the median filters. Similar situations happen when scaling attacks and frequency mode line removal attacks are applied.

5.6 Summary

Watermarking is a process of adding some identification information into the host data, such as images, audio, video and documents. The embedded information, known as watermark, can be extracted from the watermarked multimedia contents to provide proofs for various purposes, such as copyright protection and authentication. Investigation of video watermarking techniques is mainly concerned about compressed video of the prior successful coding standards, such as MPEG-2 and MPEG-4. However, digital watermarking has not yet been investigated for the newly developed video standard H.264.

This chapter began with an investigation of an image-in-image watermarking by using the Integer DCT transform. The main purpose of this research is to extend these techniques to the H.264 standard. It applies a new integer 4×4 transform based on the DCT. For our proposed approach, a 16×16 intensity watermark image was embedded into several standard test images based on the integer DCT transform. Feature extraction techniques were applied to make the embedding gain adaptive to the Canny edge characteristics of the test images. Several post-processing techniques were proposed to enhance the robustness of the watermarking scheme. Experimental results showed that these post-processing techniques have good performance against geometric attacks such as scaling, cropping and rotation attacks. The experimental results showed that the proposed image watermarking scheme based on the integer DCT was robust against up to 65% of the Stirmark attacks.

We studied and compared performances of the integer DCT against the Hadamard transform and various DCT transforms. Although the integer DCT is an approximate transform to DCT with acceptable errors due to approximation, it still maintains most of the properties of the DCT. The image watermarking scheme based on the integer DCT even showed to be more robust than

82 *Chapter 5. PROPOSED ROBUST WATERMARKING OF DIGITAL IMAGES*

the DCT algorithm against some attacks such as JPEG compression, Gaussian additive noise, frequency mode line removal and Gaussian filtering.

Chapter 6

PROPOSED ROBUST WATERMARKING OF H.264/AVC STANDARD

6.1 Introduction

In the previous chapter, greyscale watermark patterns were successfully embedded into host images with good visual quality. The scheme showed satisfactory robustness under StriMark attacks, and the 4×4 integer DCT outperformed the traditional DCT under several signal processing attacks. In this chapter, we extend the idea of using the 4×4 integer DCT in the image-in-image algorithm to H.264/AVC video clips and we propose robust watermarking techniques for copyright protection.

A binary sequence acting as an identification number is first embedded into the host video, as proposed in [80]. Our approach is similar to Hartung's [7] and Alattar's [10] by using the spread-spectrum technique, but with several enhanced features. Our approach supports the 4×4 integer DCT transform and the Lagrangian optimization for optimal partition mode selection used

in H.264/AVC. Furthermore, the watermark retrieval method is innovative in accumulating the watermark bits within several successive frames. Hence, we propose a novel pre-processing for grayscale watermark patterns which can efficiently compress the 2-D grayscale patterns for embedding, as proposed in [81]. This can also help to solve the low capacity problem for watermarking the H.264/AVC video due to the high compression performance of this state-of-the-art video standard.

Section 6.2 presents the watermark embedding and retrieval methods for the binary sequence watermark. Section 6.4 illustrates the watermark pre-processing and the watermarking algorithm for grayscale pattern watermark. Section 6.5 and 6.6 give the simulation results of the two algorithms with different watermark types.

6.2 Methodology of Binary Sequence Watermark

The proposed watermarking method is based on a spread-spectrum technique for robust data-hiding [7]. The watermark is inserted during the video encoding process and it can be extracted by parsing the compressed stream. In order to achieve greater capacity, the proposed method is applied to not only I-slices but also to P-slices. Furthermore, frame accumulation is employed for P-frames to improve the watermark retrieval performance.

6.2.1 Watermark Formation

Assume the copyright information is represented by a binary sequence $\mathbf{w}^b = \{w^b(f), f = 0, \dots, F-1\}$, where $w^b(f) \in \{0, 1\}$ and F is the watermark length,

which is mapped to a bi-polar vector

$$w^p(f) = \begin{cases} 1 & \text{if } w^b(f) = 0 \\ -1 & \text{if } w^b(f) = 1 \end{cases} \quad f = 0, 1, \dots, F-1. \quad (6.1)$$

The spread-spectrum watermark method is defined as follows: Consider S the spreading factor, then define a bit sequence $\mathbf{b}$ by repeating S times each bi-polar watermark information bit $w^p(f)$

$$\mathbf{b} = \underbrace{\{w^p(0), \dots, w^p(0)\}}_{S \text{ times}}, \dots, \underbrace{\{w^p(F-1), \dots, w^p(F-1)\}}_{S \text{ times}}. \quad (6.2)$$

where formally,

$$b(m) = w^p(f) \quad \text{for } fS \leq m < (f+1)S, \quad f = 0, \dots, F-1. \quad (6.3)$$

Then the spread-spectrum watermark $\mathbf{w}^s$ is defined by

$$w^s(m) = \eta(m) \cdot b(m) \quad m = 0, \dots, FS-1, \quad (6.4)$$

where $\eta(m) = \pm 1$ is a pseudo-noise bit. Note S is determined experimentally according to the video frame size and watermark size.

6.2.2 Watermark Embedding

Due to the significant compression efficiency of the H.264/AVC standard, the remaining DCT coefficients of the residual data are perceptually essential. Thus, in order to balance the tradeoff between visual quality and capacity, during the H.264/AVC encoding process, only one watermark bit $w^s(m)$ is embedded into a 4×4 luminance block, where the position of the block is denoted by index

Figure 6.1: Watermark embedding scheme.

m , $m = 0, \dots, FS - 1$, as illustrated in Figure 6.1.

One certain original AC coefficient $X_m(u_o, v_o)$ in the m th block, where u_o and v_o indicate a certain position of the coefficient, is replaced by the watermark $w^s(m)$, with the adaptive local gain α_m generated by the local character analysis and a global gain factor β . The resulting coefficient $\tilde{X}_m(u_o, v_o)$ is given by

$$X_m(u_o, v_o) \leftarrow \tilde{X}_m(u_o, v_o) = \alpha_m \cdot \beta \cdot w^s(m), \quad (6.5)$$

where both α_m and β are positive values and will be discussed later. The watermark bit is therefore indicated by the polarity of the modified AC coefficient. In our scheme, we empirically choose $X_m(1, 1)$ ($u_o=v_o=1$) for watermarking since the coefficients in central positions are less possible to be changed by different directional predictions used in intra macroblock modes. After watermark embedding, quantization is performed as usual.

To increase compression efficiency, the H.264/AVC standard supports a tree-structured motion compensation [3] with 7 Inter-prediction modes for each macroblock $\mathbf{B}$ as well as two Intra-prediction modes, from 16×16 down to 4×4 for Inter-prediction, 16×16 and 4×4 for Intra-prediction. A larger partition

size is suitable for still areas and a smaller partition size is suitable for areas with detailed motion.

In H.264/AVC standard, after embedding $w^s(m)$ into all the possible Inter/Intra coding modes (denoted as $\mathbf{O}$), the best mode $\tilde{o}$ for a watermarked macroblock $\tilde{\mathbf{B}}$ is found within the constrained R and minimized D , by using Lagrangian Optimization technique for motion compensation [3]

$$\tilde{o} = \arg \min_{o \in \mathbf{O}} \left(D(\tilde{\mathbf{B}}, o) + \lambda R(\tilde{\mathbf{B}}, o) \right), \quad (6.6)$$

where D and R represent the distortion and consumed bits for encoding the mode o respectively; λ denotes the predetermined Lagrangian multiplier.

The embedded watermark is then compressed with the other residual data in the best coding method chosen by this optimized motion compensation technique. Thus, *distortion drift* will not occur because the watermark distortion inflicted on the reference blocks will not be carried over to the predicted blocks in the video decompression operation.

In Equation (6.5), the tradeoff between robustness and the imperceptibility depends on the choice of the global gain factor β and the local gain factor α_m . The gain factor β is empirically decided by users for the whole video sequence; the local gain factor α_m is generated by characteristic analysis for each 4×4 block. It is used to adapt the strength of watermark embedding to the local motion and texture characteristics of the host video. We obtain α_m from the DCT coefficients $X_m(u, v)$ according to

$$\alpha_m = \max \left\{ 0, -X_m(0, 0) + \mu \sum_{\substack{L \leq u, v \leq 3 \\ 1 \leq L \leq 2}} |X_m(u, v)| \right\}, \quad (6.7)$$

where $X_m(0, 0)$ is the DC coefficient ($u=v=0$) representing the luminance of the m th block, and the sum of the AC coefficient $X_m(u, v)$ represents the *spatial*

activity of the block. The tradeoff between the DC and AC coefficients can be adjusted by the weighting factor μ . The number of AC coefficients is limited to $L \leq 2$ to avoid watermarking the blocks with strong bright and dark edges. If the current block is in an area with a significant amount of motion activity, the residual error should also contain motion information. Otherwise, the residual will be zeros or negligible if the block is located in the motionless background. The local gain factor can alter adaptively according to the local motion activity. The human visual system is more sensitive to the distortion in smooth areas than that in textured areas, and more sensitive to the distortion in motionless areas than areas with motion activity. The local watermark gain can be linked with perceptual masking adaptively with the visual content.

Please note that the above watermark embedding process is the same for all types of frames. The only different approach between I-frames and B-/P-frames is the computation of local gain α_m . It represents the local texture characteristic for intra-predicted frames, and local motion activities for inter-predicted frames.

6.2.3 Watermark Retrieval

In the watermark retrieval procedure, fully decoding the video stream is not necessary. Let $\hat{X}_m(u_o, v_o)$ be the AC coefficient for each 4×4 block extracted after inverse Entropy coding. One watermark bit $\hat{w}^s(m)$ is extracted using the well-known correlation detector given by

$$H_f = \sum_{m=fS}^{(f+1)S-1} \eta(m) \cdot \hat{X}_m(u_o, v_o). \quad (6.8)$$

By substituting Equation (6.4) into Equation (6.5) and then into Equation

(6.8), we get

$$\begin{aligned}
 H_f &= \sum_{m=fS}^{(f+1)S-1} \eta(m) \cdot \alpha_m \cdot \beta \cdot w^s(m) \\
 &= \sum_{m=fS}^{(f+1)S-1} \eta(m) \cdot \alpha_m \cdot \beta \cdot \eta(m) \cdot b(m) \\
 &= \alpha_m \cdot \beta \cdot \hat{w}^p(f) \sum_{m=fS}^{(f+1)S-1} \eta^2(m) \\
 &= \alpha_m \cdot \beta \cdot \hat{w}^p(f) \cdot S \cdot \sigma^2,
 \end{aligned} \tag{6.9}$$

where $\sigma^2 = 1$ is the variance of pseudo-noise sequence η . As S is a positive quantity, the sign of the correlation sum determines the estimated watermark bit

$$\hat{w}^p(f) = \text{sig}(H_f). \tag{6.10}$$

Finally, the binary message bit $\hat{w}^b(f)$ is decided by

$$\hat{w}^b(f) = \begin{cases} 1 & \text{if } \hat{w}^p(f) = -1 \quad (H_f < 0) \\ 0 & \text{if } \hat{w}^p(f) = 1 \quad (H_f > 0) \\ \text{lost} & \text{if } \hat{w}^p(f) = 0 \quad (H_f = 0) \end{cases}. \tag{6.11}$$

Because the zero value of $w^p(f)$ is not defined in the original definition (shown in Equation (6.1)), the watermark bit is lost in this case. Thus, if the extracted watermark bit is lost or not equal to the original, an error occurs.

6.2.4 Watermark Embedding and Retrieval Approaches within Multiple Frames

So far the embedding and the retrieval is performed on each individual frame. Since the number of survival bits in each frame is generally very limited, it is natural to exploit the temporal dimension of the video for greater robustness.

Similar to [82], the spread watermark bits $\mathbf{w}^s = \{w^s(m), m = 0, \dots, FS-1\}$ are not only spread in each frame but also embedded pseudo-randomly in the several subsequent P-frames in the same way. Thus, the watermark is spread over different scenes within the successive frames. The survival probability is increased because the union of the survival bits of multiple scenes are surely larger than the survival bits in a single scene. Therefore, before watermark retrieval, the watermark bits within a slide window are accumulated according to

$$w^s(m) = \sum_{t_c=t_o}^{t_o+T-1} w_{t_c}^s(m), \quad (6.12)$$

where t_c denotes the current frame index, t_o denotes the index of the t_o th frame and T denotes the number of frames in the slide window, and $w_{t_c}^s(m)$ corresponds to the $w^s(m)$ component at the t_c frame. The sum of the watermark signals are then used for message extraction with the same procedure stated in Section 6.2.3. Consequently, the number of bits recovered after accumulation is more stable and even able to overcome to some extent burst watermark losses [82]. This property is especially valuable for applications that requires at least one detection within a given interval [35].

6.3 Theoretical Analysis

6.3.1 Robust Embedding

As mentioned in Section 6.2.1, the watermark component $w^s(m)$ that is embedded into video data is a bipolar component. Thus, from Equation (6.5), the watermark information is embedded into the polarity of the frequency component by the replacing operation, instead of modulating the amplitude in

traditional addition embedding methods

$$\text{sig}(\tilde{X}_m(u_o, v_o)) = w^s(m). \quad (6.13)$$

During the video processing and the transcoding process, the amplitude of the H.264/AVC compressed video data can be altered in a very wide dynamic range. Therefore, embedding watermark components into the data polarity can help increase the robustness of the watermarking scheme.

6.3.2 Effective Retrieval

For a clearer illustration, a traditional additive embedding method is compared to our algorithm. The watermark $w^s(m)$ is added to the original video data $V_m(u_o, v_o)$ as

$$\tilde{V}_m(u_o, v_o) = V_m(u_o, v_o) + \alpha_m \beta w^s(m). \quad (6.14)$$

Suppose that there are no attacks after embedding. Thus, the corresponding correlation sum G_f for watermark extraction should be

$$\begin{aligned} G_f &= \sum_{m=fS}^{(f+1)S-1} \eta(m) \cdot [V_m(u_o, v_o) + \alpha_m \beta w^s(m)] \\ &= \underbrace{\sum_{m=fS}^{(f+1)S-1} \eta(m) \cdot V_m(u_o, v_o)}_{G_1} + \underbrace{\alpha_m \beta \sum_{m=fS}^{(f+1)S-1} \eta^2(m) \cdot b(m)}_{G_2}. \end{aligned} \quad (6.15)$$

The terms G_1 and G_2 denote the contributions to the correlation sum from the host data signal and the watermark signal, respectively. Only for large values of S , G_2 can be much larger than G_1 because the video signal is statistically uncorrelated with the pseudo-noise sequence η . Thus, when S is not large enough, G_1 will be an interference in the watermark extraction. However, in

our algorithm because of the replacing method, there is no interference in the watermark retrieval (Equation (6.9)), no matter how small S is.

What's more, we even proposed the watermark accumulation through several sequential frames. This error correction approach can further increase the robustness of the embedded watermark. The experimental results based on our proposed algorithm will be shown in the next section.

6.3.3 Optimized Rate-Distortion Control

Measuring the bitrate and the PSNR at a range of quantizer settings provides an estimate of compression performance that is numerically more accurate than subjective comparisons. Figure 6.2 compares the rate-distortion performance of the H.264 encoder without watermarking and with watermarking. All tests are performed with fixed quantization parameters.

As proposed in Section 6.2.1, the binary copyright information is first mapped into a bi-polar sequence, and then spread-spectrum method is applied to it to gain a pseudo-random sequence which is S times longer than the original information. In Equation (6.2), the same component is repeated S times to increase the redundancy for more robustness. By repeating the same watermark component, the probability of obtaining the correct extracted watermark is increased. This is because the correlation detector given in Equation (6.8) can be served as an error correction in our algorithm. Thus, in our algorithm, in order to achieve high robustness, S is set to be enough high so that the total length of the spread-spectrum watermarked component may be much longer than the suitable integer DCT coefficient of the video residual data. This may happen more often if the target over all consumed bit-rate is relatively low.

In Figure 6.2, When the consumed bit-rate is targeted as low as 200 kbits/sec, most video data is highly compressed. There are few suitable integer DCT coefficients to be embedded. Especially when the encoder finds that there is not

enough information to be coded in the current macroblock, the skip mode [3] of inter-coded current macroblock will be used by the H.264/AVC. For this mode, neither a quantized prediction residual signal, nor a motion vector or reference index, has to be coded and transmitted. It contains no non-zero quantized transform coefficients even if the watermark component has been inserted into this macroblock. Therefore, not all spread-spectrum watermarked components can be embedded into the same frame with low target over all bit-rates. As the consumed bit-rate increases, more integer DCT coefficients are produced and thus can be used for watermark embedding. Consequently, more distortion will be introduced. In Figure 6.2, the gap between the PSNR values of the original frames and the watermarked frames become larger as the bit-rate increases, and the decreased PSNR value due to watermark embedding becomes higher. However, on average under the same coding bit-rates, there is only approximately 0.9% and 0.6% PSNR decrease on average for *Foreman* and *Stefan*, respectively. Figure 6.3 juxtaposes one of the unmarked and marked frames of *Foreman* and *Stefan* to demonstrate the visual quality after watermarking. No artifacts are visible. This proves the effectiveness of our watermark gain control mechanism and the optimized rate-distortion control of the H.264/AVC.

6.4 Methodology of Grayscale Pattern Watermark

The purpose of using images as watermarks is to increase the robustness of the watermarking scheme. When a watermark image undergoes attacks, the chances of recovery is higher than compared to other kinds of watermarks. When the watermark image gets distorted after the video gets attacked, it can still be recognized. Also the advantage of the H.264 video coding techniques has led to a significant decrease in redundancy for watermark embedding. In the

(a) Foreman

(b) Stefan

(c) Flower Garden

(d) Coastguard

Figure 6.2: Comparison of the rate-distortion performance for H.264 encoding without watermark ($\circ$) and with watermark ($\times$) (512 kbits/sec, CIF-size).

(a) Unmarked (41.01dB)

(b) Marked (40.33dB)

(c) Unmarked (34.67dB)

(d) Marked (34.35dB)

Figure 6.3: The 30-th frame of *Foreman* and *Stefan* (512 kbits/sec, CIF-size) without watermark and with watermark.

example, in bi-predicted slices of a video clip with low motion activity at a low coding bitrate, most of the coefficients after motion compensation and motion estimation are zero [12]. Accordingly, the 2-D grayscale watermark should be pre-processed to decrease the data size for embedding. A grayscale watermark pre-processing is proposed to deal with this problem. For a grayscale image, the raw value of one pixel can vary from 0 to 255. After the pre-processing approach proposed in Section 6.3, the processed data is simplified into a binary sequence, which will be used as watermark to embed.

6.4.1 Classification

The preliminary analysis for watermark pre-processing has been focused on grayscale patterns with characters. Therefore, we need to classify the 26 capital letters of the English alphabet and 10 numerical numbers for the adaptive process, such as *Transformation*, which will be explained in Section 6.3.2. Let us define four categories as follows

- *Category C*: C, J, O, Q, S, U, 0, 3, 6, 8 and 9;
- *Category L*: E, F, H, I, L, T and 1;
- *Category L + C*: B, D, G, P, R, 2, 4 and 5;
- *Category L + D*: A, K, M, N, V, W, X, Y, Z and 7.

where $\mathcal{C}$, $\mathcal{L}$ and $\mathcal{D}$ denote curves, lines and diagonals, respectively.

6.4.2 Watermark Pre-processing

This process is conducted before the watermark formation mentioned in Section 6.2.1, and can be carried out in the five main approaches: (1) 4×4 integer DCT and zigzag scanning, (2) coefficient normalization, (3) adaptive frequency masking, (4) coefficient transformation and (5) level reduction. Figure 6.4 shows

Figure 6.4: Flow-chart of watermark pre-processing

the flow-chart of the watermark pre-processing steps. Watermark patterns can then be reconstructed from the 1-D output sequence obtained from the pre-processing.

6.4.2.1 4×4 Integer DCT and Zigzag-scanning

The grayscale pattern $\mathbf{p}$ of size $M \times M$ is first decomposed into non-overlapping 4×4 sub-blocks, denoted by $\mathbf{p}_l$, where $\mathbf{p}_l = \{p_l\{i, j\}, \quad 0 \leq i, j \leq 3\}$, and $l = 0, 1, \dots, (M/4)^2 - 1$. Then, each sub-block $\mathbf{p}_l$ is transformed into the integer DCT domain as follows

$$P_l(u, v) = \text{IntDCT}\{p_l(i, j)\} \quad 0 \leq u, v \leq 3, \quad (6.16)$$

where $\text{IntDCT}\{\cdot\}$ represents the 4×4 integer DCT with low-complexity and higher accuracy and is uniquely used in H.264/AVC [3]. This transform is approximated to the traditional DCT used in all prior video standards. However, it has many fundamental differences which contribute to a better performance for video coding [3] and digital watermarking [11].

The coefficients $P_l(u, v)$ of the 2-D 4×4 integer DCT transformed matrix are zigzag-scanned into a 1-D sequence of 16 integer DCT coefficients $P_l(g)$, where $g \in [0, 15]$ indicates the position in zigzag-scanning.

6.4.2.2 Normalization and Frequency Mask

After integer DCT and zigzag-scanning, several lowest frequencies possess the significant energy with sufficient information for the pattern reconstruction. To achieve higher efficiency of the pre-processing, the remaining high frequencies are discarded. To effectively discard unnecessary coefficients, after performing a DCT transformation in Equation (6.16), all 16 coefficients $P_l(g)$ in each 4×4 block are normalized as follows

$$P_l^n(g) = \frac{|P_l(g)|}{\sum_{k=0}^{15} |P_l(g)|} \quad 0 \leq g \leq 15. \quad (6.17)$$

From the experimental results, the top four normalized coefficients $P_l^n(g)$ with the largest values are at different zigzag scanning positions g for patterns with characters from various categories as follows

- *Category C*: $g \in \mathbf{G}_C = \{0, 1, 2, 5\}$;
- *Category L*: $g \in \mathbf{G}_L = \{0, 1, 3, 5\}$;
- *Category L + C*: $g \in \mathbf{G}_{LC} = \{0, 1, 2, 5\}$;
- *Category L + D*: $g \in \mathbf{G}_{LD} = \{0, 1, 5, 6\}$.

Therefore, four different sets $\mathbf{G}_C$, $\mathbf{G}_L$, $\mathbf{G}_{LC}$ and $\mathbf{G}_{LD}$ are generated from the normalized coefficients $P_l^n(g)$. Next, only the DCT coefficients $P_l(g)$ with the index g belonging to the sets $\mathbf{G}_C$, $\mathbf{G}_L$, $\mathbf{G}_{LC}$ or $\mathbf{G}_{LD}$ are kept, and the coefficients with g not belonging to the these sets are discarded. This technique is known as *Adaptive Frequency Masking*, and these sets $\mathbf{G}_C$, $\mathbf{G}_L$, $\mathbf{G}_{LC}$ or $\mathbf{G}_{LD}$ act as the frequency masks.

6.4.2.3 Transformation

After masking, there is a significant difference between the dynamic range of every two adjacent coefficients. For example, the DC coefficients always have

Table 6.1: Parameters a and d over the frequency index g for watermark pre-processing ($\mathcal{C}$, $\mathcal{L}$ and $\mathcal{D}$ denote Curves, Lines and Diagonals, respectively.)

Frequency Index g	$\mathcal{C}$		$\mathcal{L}$		$\mathcal{L} + \mathcal{C}$		$\mathcal{L} + \mathcal{D}$	
	$a(g)$	$d(g)$	$a(g)$	$d(g)$	$a(g)$	$d(g)$	$a(g)$	$d(g)$
0	0	166	0	140	0	195	0	157
1	500	150	430	140	440	145	420	150
2	100	40	250	100	400	100	300	100
3	50	40	150	60	100	60	160	60
4	200	60	250	80	300	100	200	50
5	300	100	400	120	350	120	450	200
6	300	70	400	70	120	50	260	80

larger values than AC coefficients, and lower frequency coefficients are always larger than higher frequency coefficients. The dynamic ranges of these coefficients must be narrowed down, to avoid undesirable distortion during watermarking. Thus, the remaining coefficients $P_l(g)$ are transformed as follows

$$P_l^T(g) = \frac{P_l(g) + a(g)}{d(g)} \quad 0 \leq g < 15, \quad (6.18)$$

where the value of transformed result $P_l^T(g)$ has been decreased with in the range from 1 to 7, and $a(g)$ and $d(g)$ represent the corresponding additive and denominator parameters to the zigzag scanning position g , and the values are decided according to experiments.

In our experiments, in order to narrow all the different dynamic ranges down to this unique range from 1 to 7, $a(g)$ and $d(g)$ both change adaptively to zigzag position g of the corresponding coefficients, and also vary to different categories of characters as mentioned. Only the pairs corresponding to the seven lowest frequencies are shown in Table 6.1.

Figure 6.5: Flow-chart of watermark Reconstruction

6.4.2.4 Level Reduction

The coefficients $P_l^T(g)$ have multiple values, $P_l^T(g) \in [1, 7]$. They can be further simplified to be binary, denoted by $w_l(g)$ as

$$w_l^b(g) = P_l^T(g) \bmod 2, \quad (6.19)$$

$$Key(g) = P_l^T(g)/2, \quad (6.20)$$

where g is in the sets $\mathbf{G}_C$, $\mathbf{G}_L$, $\mathbf{G}_{LC}$ or $\mathbf{G}_{LD}$ and the sequence Key can be stored in a key parameter file for future reconstruction.

Then, 1-D binary watermark sequence $\mathbf{w}^b$ can be obtain as follows

$$\mathbf{w}^b = \bigcup_{l=0}^{M^2/16-1} \{w_l^b(g), \quad g \in \mathbf{G}_C, \mathbf{G}_L, \mathbf{G}_{LC} \text{ or } \mathbf{G}_{LD}\}. \quad (6.21)$$

6.4.3 Watermark Reconstruction

Watermark image can be obtained by reconstructing the 1-D binary sequence output $\mathbf{w}^b$ of the watermark pre-processing. The reconstruction is the inverse approach of our proposed watermark pre-processing, as shown in Figure 6.5.

The first step is an inverse approach of level reduction. The binary bit $w_l^b(g)$ is processed as below to obtain the coefficient with the value range from 1 to 7

$$P_l^T(g) = w_l^b(g) \times 2 + Key(g). \quad (6.22)$$

Next, in the de-transformation approach, the inverse method of Equation (6.18) is applied to the output of Equation (6.22)

$$P_l(g) = P_l^T(g) \times d(g) - a(g). \quad (6.23)$$

Then, according to the frequency masking approach in the pre-processing method, the de-transformed coefficients are mapped into their corresponding frequency positions by referring to sets $\mathbf{G}_C$, $\mathbf{G}_L$, $\mathbf{G}_{LC}$ or $\mathbf{G}_{LD}$.

For example, if $P_l(g)$ are the coefficients for a watermark pattern which in in $\mathbf{G}_C = 0, 1, 2, 5$, then each 4 components from $P_l(g)$ are mapped into frequency positions 0,1, 2 and 5. The value of other frequency coefficients are set to be 0.

The inverse integer DCT is applied to each 4×4 reconstructed blocks to generate the 2-D grayscale image.

Table 6.2 shows two groups of watermark samples reconstructed directly after pre-processing without watermarking, and the corresponding normalized correlation. All patterns have the same size $M \times M$, where M is equal to 32. A grayscale pattern (*NTU*) with a hybrid combination from different character categories and an *EEE* logo are also illustrated. In the case of *NTU*, the pattern can be separated into three parts and each part undergoes the pre-processing separately.

Before pre-processing, the dynamic range of all the pixels of the $M \times M$ patterns is from 0 to 255. After the proposed method, the values of all processed DCT coefficients have been reduced to be binary.

In Group A of Table 6.2, the 7 lowest frequencies ($g \in [1, 7]$) in each 4×4 block are remained without adaptive frequency masking. In Group B, normalization and adaptive frequency masking are conducted, and only four coefficients are remained after masking. The number of coefficients for watermarking has been reduced to 43.8% in Group A and 25.0% in group B.

Table 6.2: Reconstructed watermarks after pre-processing and the corresponding correlations ($\mathcal{C}$, $\mathcal{L}$ and $\mathcal{D}$ denote Curves, Lines and Diagonals, respectively.)

Pattern	$\mathcal{C}$	$\mathcal{L}$	$\mathcal{L} + \mathcal{C}$	$\mathcal{L} + \mathcal{D}$	Others	
Original	689	ILT	BRG	WXY	NTU
Group A	689	ILT	BRG	WXY	NTU
Correlation A	0.97	0.99	0.98	0.99	0.99	0.96
Group B	689	ILT	BRG	WXY	NTU
Correlation B	0.92	0.99	0.95	0.97	0.98	0.91

The reconstructed grayscale watermarks are highly correlated to the original as given in Table 6.2, which demonstrates and supports the feasibility of the proposed pre-processing. The watermark patterns with more straight lines, such as in Category $\mathcal{L}$, can obtain better reconstruction than patterns in Category $\mathcal{C}$ and $\mathcal{L} + \mathcal{D}$. This is due to the fact that watermarks with more curves always have more energy in higher DCT frequencies, while most of the high frequencies are discarded during the pre-processing. For higher robustness in terms of higher correlation to the original, the transformation parameter $d(g)$ can be slightly larger and more frequencies can be retained during frequency dropping. While for higher capacity of watermarking scheme, more low and medium frequencies can be remained which could obviously trade for lower robustness.

6.4.4 Watermark Embedding and Retrieval

The proposed grayscale watermark pre-processing can change a 2-D 8-bit watermark pattern ($M \times M$) into a binary sequence $\mathbf{w}$ with the shortest length $M^2/4$.

The embedding and retrieval processes are the same as mentioned in Sec-

Figure 6.6: Proposed Video watermarking algorithm

tions 6.2.1, 6.2.2 and 6.2.3.

After extracting the binary sequence $\hat{\mathbf{w}}^b$ from the compressed H.264 video, the inverse approach of the watermark pre-processing (as mentioned in Section 6.3.3) reconstructs the extracted grayscale watermark.

6.5 Experiments and Results of Binary Sequence Algorithm

6.5.1 Test Conditions

The proposed watermarking technique has been integrated into the H.264 JM-8.6 reference software [83] and evaluated in terms of rate-distortion performance, objective quality, and robustness. The first 100 frames of the video sequences: *Foreman*, *Stefan*, *Coastguard*, and *Flower Garden* are used in the experiments. All sequences are of CIF resolution (352×288 pixels) and encoded at the frame rate 15 frames/sec with the bit-rate 768 kbits/sec, 512 kbits/sec, and 396 kbits/sec, respectively. The GOP structure comprises IPP..., which is compliant to the *Baseline Profile* [3] (first frame coded as an I-slice, subsequent frames coded as P-slices, one reference frame used for inter prediction, UVLC/CAVLC entropy coding). Due to the limitation caused by small CIF frame size and highly efficient compression, we only embed a 96-bit message to

every frame. With more watermark bits, it will be more difficult to balance the tradeoff among robustness, fidelity and capacity.

6.5.2 Results of Robustness Tests

Two categories of video attacks have been applied to the watermarked video to test the robustness of the watermarking algorithm: transcoding and common signal processing operations.

In the first group, we utilized the bit-rate reduction. During watermarking in the H.264/AVC encoding process, the Lagrangian Optimization technique targets the overall consuming bit-rate at 768 kbits/sec, 512 kbits/sec and 396 kbits/sec. When transcoding is applied after watermarking, the bit-rates have been reduced to approximately 1/2 and 1/3 of the original bit-rates. During re-encoding with the bit-rate reduction, most high frequency components which represent detailed texture will be discarded, especially when the bit-rates have been decreased to the 1/3. Our robust method can still robustly survive under such strong attacks.

Table 6.3: The bit error rate of the recovered message after transcoding attacks.

Sequence	Original Bitrate	No Attack		To $\frac{1}{2}$ bitrate		To $\frac{1}{3}$ bitrate	
		$T = 1$	$T = 5$	$T = 1$	$T = 5$	$T = 1$	$T = 5$
Stefan	768	0.03	0.01	0.22	0.02	0.67	0.28
	512	0.15	0.01	0.67	0.26	0.93	0.75
	396	0.32	0.02	0.82	0.47	0.94	0.77
Coastguard	768	0.08	0.01	0.35	0.04	0.79	0.41
	512	0.35	0.03	0.80	0.42	0.96	0.82
	396	0.56	0.11	0.88	0.61	0.95	0.79
Foreman	768	0.04	0.02	0.26	0.04	0.65	0.26
	512	0.14	0.02	0.64	0.24	0.92	0.71
	396	0.26	0.02	0.77	0.41	0.92	0.73
Flower	768	0.00	0.00	0.14	0.00	0.59	0.20
	512	0.04	0.00	0.56	0.18	0.89	0.62
	396	0.15	0.00	0.70	0.29	0.87	0.59
Average	558.7	0.13	0.02	0.49	0.19	0.73	0.44

After decoding the marked bitstream, common signal processing attacks are applied to the raw video frame by frame, including 5×5 Gaussian low-pass filtering, circular average filtering, unsharpened contrast enhancement, and additive Gaussian noise (mean=0, variance= 0.001). Note that the watermarked video streams need to be decoded first and compressed again after processing. Consequently, the additional compression also contributes to the watermark loss.

The results are given in Table 6.3 and Table 6.4. When $T = 1$, the watermark is extracted in every frame; when $T = 5$, the detection with frame accumulation is performed in a sliding window of 5 frames in the time axis. In Table 6.3, the three rows of each test sequence are at 768 kbits/sec, 512 kbits/sec and 396 kbits/sec, respectively.

The watermark robustness is measured in *Bit Error Rate* (BER), which is the ratio of the number of error bits to the total number of the embedded watermark bits. Note that different from conventional definition of BER, *error bits* in our scheme include not only the bits that are opposite to the original but also those deemed lost according to Equation (6.11).

Table 6.4: The bit error rate of the recovered message after some common signal processing attacks. All three sequences are compressed at 512 kbits/sec.

Sequence	3 × 3 Gaussian Filtering		5 × 5 Gaussian Filtering		Unsharp Filtering		Additive Noise	
	$T = 1$	$T = 5$	$T = 1$	$T = 5$	$T = 1$	$T = 5$	$T = 1$	$T = 5$
Stefan	0.39	0.08	0.38	0.07	0.31	0.07	0.51	0.31
Coastguard	0.56	0.17	0.57	0.17	0.41	0.11	0.57	0.39
Foreman	0.41	0.11	0.41	0.12	0.28	0.06	0.57	0.44
Flower	0.17	0.02	0.18	0.02	0.27	0.04	0.34	0.09
Average	0.34	0.08	0.35	0.08	0.28	0.06	0.48	0.27

Table 6.3 and Table 6.4 demonstrate that the proposed watermarking method is fairly robust against transcoding and these common signal processing attacks.

The average BER per frame is 0.13 for un-attacked watermarked video streams. The higher the video is coded at, the lower the BER gets. After transcoding to half of the original bit-rate, about half of the message bits can not be correctly extracted due to the host data lost. By using the sliding window accumulation, more message bits can be correctly extracted with the BER drops to 0.19 on average. From these two tables, accumulation increased the robustness by 53% BER decrease on average. This shows the effectiveness of the proposed frame accumulation method. Moreover, the fact that the watermark components are embedding into the polarities of the DCT coefficients by replacing the original data, instead of modifying the amplitude (such as additive methods) also plays an important role in better watermark extraction and better robustness against attacks. The overall robustness can be further improved in the algorithm mentioned in Section 6.3 with the 2-D grayscale watermark format.

To have an appreciation regarding how the slide-window accumulation performs, we take *Foreman* (originally coded at 512kbits/sec with watermark) as an example and plot the BER curves with recompression at the same bit-rate in Figure 6.7(a) and the BER curves after transcoding attack of bit-rate change (to 1/2) in Figure 6.7(b). In both figures, the BER with $T = 5$ is always lower than that with $T = 1$. It further corroborates that the watermark retrieval performance has been significantly improved by the slide-window accumulation. Note that in Figure 6.7, only the BER values for P-frames are plotted. It leads to the significant BER increase for the 1st and the 50th frame, which are I-frames.

6.5.3 Algorithm Comparison

Although the H.264/AVC standard has been finalized since 2004, only recently researchers have begun to publish research in data hiding in H.264 [84]. Among the very first few publications about H.264 watermarking algorithms is the one

Figure 6.7: *Bit error rate* (BER) of the watermark recovered from the P-frames for the sequences before transcoding (a) and after transcoding attacks (b) with two slide-window sizes, $T=1$ and $T=5$.

by T.-H. Chen et al. (8/2005) [85] who proposed a binary data hiding algorithm based on the stability of the direct current coefficient values in blocks transformed by DCT. The frame is randomly selected to embed a watermark and then its luma is put in 8-by-8 blocks and transformed by DCT. The watermark is embedded in the high frequency coefficients by addition, which can be simplified by the following equation,

$$X' = X + \alpha W, \quad (6.24)$$

where W is the watermark bit, X and X' are the video DCT coefficients before and after watermarking respectively, and α is the scaling factor determined by the algorithm design.

During the detection process, the correlation between the watermark and the high frequency coefficients is computed, However, this algorithm can not extract the embedded watermark sequence from the watermarked frames. It can only detect whether the frame has been embedded with a watermark or not.

M. Yang et al. (8/2005) [86] developed a high bit-rate information hiding algorithm for digital video content under the H.264/AVC compression and 1 bit is hidden with each 4-by-4 DCT block. Low frequency coefficients are chosen for information hiding due to their relatively large amplitude and the corresponding small step sizes in the quantization matrix. The embedding process is simplified as follows,

$$X' = \begin{cases} (1 + \alpha)X & \text{if } W = 1 \\ (1 - \alpha)X & \text{if } W = 0 \end{cases}. \quad (6.25)$$

During watermark extraction, the original video data is compared with the watermarked coefficients to extract the binary watermark bit. This algorithm can survive H.264 compression itself by the BER equal to 0.2 under no other

attacks. In [86], the authors mentioned that the algorithm is not robust against signal processing attacks, because any simple signal processing operation, such as filtering, changes the prediction modes and subsequently the residual data of H.264. Our proposed replacement embedding algorithm has overcome this drawback.

Table 6.5: Algorithm Comparison

	Ours	Yang's	Chen's
Watermark type	binary	binary	binary
Embedding	Replacement	addition	addition
Retrieval	Yes	Yes	No(only detection)
Robustness (no attacks)	BER = 0	BER = 0.2	N/A
Robustness (w/ attacks)	BER = 0.1	N/A	N/A

In Table 6.5, our proposed algorithm is compared with Chen's and Yang's algorithms. Both Chen's and Yang's algorithms embed the watermark by addition, which in turn modulated the amplitude of the video coefficients. Our algorithm embeds the watermark into the polarity of the frequency component by the replacing operation. The robustness comparison in Table 6.5 proves that the replacement embedding in our algorithm is more robust than addition embedding in Chen's and Yang's algorithms. During the video processing and the transcoding process, the amplitude of the H.264/AVC compressed video data can be altered in a very wide dynamic range. Therefore, embedding watermark components into the data polarity can help increase the robustness of the watermarking scheme.

6.5.4 Summary

A robust watermarking technique has been presented for the state-of-the-art H.264/AVC video coding standard, based on the previous investigation of the image-in-image watermarking algorithm. The optimal prediction mode selec-

tion 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.

6.6 Experiments and Results of Grayscale Watermark Algorithm

The same test video sequences are used in the grayscale pattern algorithm with the same test conditions mentioned in Section 6.5.1. A small 16×16 grayscale watermark is used as the watermark pattern for the grayscale watermark algorithm.

6.6.1 Fidelity after Watermarking

Figure 6.8 shows the average PSNR comparison results (dB). The unmarked and marked video clips are reconstructed from the compressed data without and with watermarks, respectively. The average PSNR values are computed by comparing the reconstructed video frames to the original raw video frames. On average, the watermarking leads to a decrease of approximately 1.1 dB, equal to the average PSNR drop in 6.3.3. This is mainly due to the Lagrangian Optimization in H.264/AVC and the proposed watermark gain factors. The payload of every frame in both algorithms are the same. Figure 6.9 shows both *Foreman* and *Container* as examples of video clips with significant motion and with minimal motion, respectively. In the experiments, no visible artifacts can be observed in all of the test video sequences.

(a) 768kbits/sec

(b) 512 kbits/sec

(c) 396kbits/sec

Figure 6.8: PSNR(dB) of unmarked and marked *Foreman*, *Stefan*, *Coastguard*, *Flower*, *silent* and *Container* (denoted by *For*, *Ste*, *Coa*, *Flo*, *Sil* and *Con* in the horizontal axis, respectively)(768kbits/sec, 512kbits/sec, 396kbits/sec, CIF-size).

(a) Original(41.55dB)

(b) Watermarked(41.00dB)

(c) Original(43.30dB)

(d) Watermarked(39.54dB)

Figure 6.9: The 30th frame of *Foreman* and *Container* (768 kbits/sec, CIF-size) without and with watermark.

(a) To 1/3 bitrate(35.11dB)

(b) 5×5 Gaussian filter(35.73dB)

(c) Contrast enhancement(22.91dB)

(d) Gaussian noise(31.04dB)

Figure 6.10: The 30th frame of attacked *Foreman* (768 kbits/sec, CIF-size).

6.6.2 Results of Robustness Tests

The same video attacks in Section 6.5.2 are applied again. Figure 6.10 shows the examples of attacked *Foreman* (768 kbits/sec, CIF-size). The visual quality decreases significantly, especially under contrast enhancement and Gaussian noise.

Table 6.6 shows the reconstructed watermark without watermarking and the retrieved samples from the marked and attacked *Foreman* (768 kbits/sec, CIF-size). These samples are highly correlated to the original pattern. The proposed

watermarking algorithm together with the watermark pre-processing can still survive strong attacks, such as bit-rate reduction, contrast enhancement and Gaussian noise.

Table 6.6: Samples of the original, reconstructed and extracted watermarks from the 30th frame of *Foreman* (768 kbits/sec, CIF-size) and normalized correlation.

Original Watermark	K
Reconstructed after WM Pre-processing	K ($\rho = 0.97$)
Extracted from watermarked video	K ($\rho = 0.93$)
Extracted after transcoding (to 1/3 Bit-rate)	($\rho = 0.50$)
Extracted after 5x5 Gaussian Filtering	($\rho = 0.72$)
Extracted after Circular Filtering	K ($\rho = 0.87$)
Extracted after Contrast Enhancement	($\rho = 0.83$)
Extracted after Gaussian Noise	($\rho = 0.75$)

The robustness is measured by the correct bit rate between the original grayscale watermark and the extracted watermark as illustrated in Figure 6.11. The average correct bit rate per frame for all watermarked video streams at 768 kbits/sec is approximately 0.90. After transcoding to 1/3 of the original bit-rate, the average rate drops to approximately 0.58. After contrast enhancement, the average rate only decreases by approximately 12%.

Moreover, among the six test sequences, *Silent* and *Container Ship* give the best robustness after transcoding and signal processing attacks. This leads to the conclusion that more correct watermark bits are recovered in the two sequences. This is due to the fact that there is only minimal motion in the two sequences. Most of the watermark bits are automatically embedded into the

(a) No attacks

(b) To 1/3 bitrate

(c) 5 x 5 Gaussian filter

(d) Circular filter

(e) Contrast enhancement

(f) Gaussian noise

Figure 6.11: Robustness under transcoding and common signal processing attacks on *Foreman*, *Stefan*, *Coastguard*, *Flower*, *silent* and *Container* (denoted by *For*, *Ste*, *Coa*, *Flo*, *Sil* and *Con* in the horizontal axis, respectively) (768 kbits/sec, 512 kbits/sec, and 396 kbits/sec, CIF-size).

textured background with high gain. These bits are relatively more robust, even though the introduced distortion is higher than the other four sequences with significant motion. These results validate the fundamental tradeoff between the fidelity of the video data and the watermark robustness.

Figure 6.11 shows that the proposed watermarking algorithm together with the watermark pre-processing can survive strong attacks, such as bit-rate reduction, contrast enhancement and Gaussian noise. The watermark patterns can tolerate the distortion introduced by some incorrect extracted watermark components, and perform better than binary watermark sequences. Moreover, the fact that the watermark components are embedded into the polarities of the DCT coefficients of the residual data, instead of modifying the amplitude also plays an important role in increasing the robustness of the algorithm. Furthermore, by applying the spread-spectrum to the pre-processed watermark components before embedding repeatedly spreads the watermark components into every macroblock of each frame. This also increases the robustness of our algorithm.

6.6.3 Summary

The proposed grayscale watermark pre-processing with the presented video watermarking algorithm for copyright protection technique performed well in terms of robustness and data capacity. The watermarked H.264/AVC video clips maintained the good visual quality and almost the same Bit-rate. Moreover, detailed copyright information, such as textured company trademarks or logos can also be used as watermark to further protect ownership and defend against illegal attacks.

Chapter 7

PROPOSED HARD AUTHENTICATION OF H.264/AVC STANDARD

7.1 Introduction

With the exhaustive investigation of the related techniques of the state-of-the-art H.264/AVC standard, we develop a novel and efficient watermarking algorithm for authentication. This algorithm works directly in the compressed domain. The scheme makes an accurate usage of the tree-structured motion compensation, motion estimation and Lagrangian optimization for mode decision. By making use of this feature to our advantage, a careful and detailed study was done on how this feature can be exploited in the implementation of a digital watermarking scheme. The authentication information is represented by a binary watermark sequence and embedded into video frames. The experimental results prove the effectiveness of the algorithm.

The rest of this chapter is organized as follows. Section 7.2 provides related techniques of the H.264/AVC standard. Section 7.3 gives a detailed explanation

Figure 7.1: Macroblock partition modes: 16×16 , 8×16 , 16×8 , 8×8 .

of the H.264 hard authentication algorithm. Analysis of the proposed algorithm is given in Section 7.4. Some experimental results of the developed algorithm are presented in Section 7.5. Section 7.6 concludes this chapter with the outline of future work.

7.2 Investigation of Best Mode

7.2.1 Tree-structured Motion Compensation

Significant differences of the H.264/AVC standard from earlier standards include the support for a range of block sizes (from 16×16 down to 4×4) for prediction and fine sub-sample motion vectors (quarter-sample resolution in the luma component), as mentioned in Chapter 3.

The luma component of each macroblock may be split up and motion compensated in four ways as shown in Figure 7.1. In cases where the 8×8 partition mode is chosen, each of the four 8×8 sub-macroblocks within the macroblock can be further split up in four ways as shown in Figure 7.2. This method of partitioning macroblocks into motion compensated sub-blocks of varying sizes is known as *tree-structured motion compensation* [3]. Therefore, the choice of partition mode significantly impacts the compression performance. In general, a large partition mode is appropriate for homogeneous areas of the frame and a small partition mode may be beneficial for detailed areas. Figure 7.3 shows

Figure 7.2: Sub-macroblock partition modes: 8×8 , 4×8 , 8×4 , 4×4 .

Figure 7.3: Examples of partition modes using the tree-structured motion compensation.

four macroblocks with different partition modes.

In addition to the motion-compensated macroblock partition modes described before, a P-slice macroblock may also be coded in the *SKIP* mode [87]. If a macroblock has motion characteristics that allow its motion to be effectively predicted from the motion of neighboring macroblocks, and it contains no or very few non-zero quantized transform coefficients, then it is flagged as skipped. For this mode, neither a quantized prediction residual signal, nor a motion vector or reference index, has to be coded and transmitted. B-slices also have a special prediction mode - *Direct mode*. In this mode, no prediction error signal is coded and transmitted. It is also referred to as *B-slice SKIP mode* and can be coded very efficiently, in a similar way to the *SKIP* mode in P-slices. In this situation, no watermark components are embedded, thus, the capacity decreases. This has been considered in the proposed authentication scheme, as shown in Figure 7.6.

The best mode o^* for a macroblock B is selected by minimizing the expression in Equation (7.1) within the constrained R and minimized D , using Lagrangian optimization technique [50] of H.264/AVC, among all possible modes

denoted by O

$$o^* = \arg \min_{o \in \mathcal{O}} \left(D(\mathbf{B}, o) + \lambda R(\mathbf{B}, o) \right), \quad (7.1)$$

where λ denotes the predetermined Lagrangian multiplier for mode choice, and D and R represent the distortion and consumed bits for encoding the current mode o , respectively.

Figure 7.4(a) shows one frame of *Foreman* and the partition mode selection map. In areas where there is little change between the frames, a 16×16 partition mode is chosen and in areas of detailed motion, small partitions are more efficient.

7.2.2 Partition Mode Change

For each macroblock, the H.264 encoder selects the best partition mode among all possible partition modes that minimizes the amount of information to be coded and sent. During transmission, once the coded stream undergoes any transcoding process (such as recompression, bit-rate change, and frame-rate change), the information may alter significantly. Thus the best partition modes for some macroblocks may be different. If more malicious signal processing attacks are applied, more macroblocks may have different best modes.

Figure 7.4(b) shows the same frame as in Figure 7.4(a) and the new partition mode selection map after recompression. From these two figures, the best partition modes of many macroblocks are not the same. The mode map of difference between Figures 7.4(a) and (b) is shown in Figure 7.5, as the macroblocks with mode change are covered by transparent gray color. For example, for the fourth macroblock in the second row, due to recompression, its best mode changes from 8×16 to 16×16 . For the eighth macroblock in the fifth row, its original best mode is 8×16 . After recompression the new best mode is 8×8 . Each 8-by-8 sub-macroblock also has a best sub-mode: 8×8 ,

(a) mode map of the original coded B-frame

(b) mode map of the re-compressed B-frame

Figure 7.4: *Foreman's* partition mode selection: (a) of the original coded B-frame, (b) of the re-compressed B-frame.

Figure 7.5: Mode map of the difference between the original coded *Foreman* B-frame and the re-compressed B-frame (macroblocks with mode change are covered by transparent gray color).

4×8 , 8×8 , 4×8 . For some macroblocks, even if their best modes are the same, the information may be different. Thus the DCT coefficients may not be the same, when compared to the coefficients before recompression.

7.3 Proposed Authentication Method

7.3.1 Embedding

In a H.264/AVC video sequence, there are a total of five different types of slices: Intra (I-), Predicted (P-), Bi-predictive (B-), Switching Intra (SI-) and Switching Predictive (SP-). Our algorithm is developed for the inter-predicted slices: P- and B-slices.

As mentioned before, the encoder needs to find the best partition mode for each macroblock, as different modes will produce different sets of bitrate and distortion to the video stream. The encoder will go through the motion estimation and compensation, transformation, quantization and entropy coding

for all possible partition modes, and the Lagrangian optimization technique determines which partition mode has the lowest rate-distortion related cost in Equation (7.1). Only when the minimum cost is attained, the encoder will allocate the corresponding partition mode as the best mode to the macroblock. Through careful observation of the mode decision scheme, it can be certain that in a region where there is no motion (such as background), a partition mode of 16×16 is chosen by the encoder. In areas where there is a lot of detailed motion, smaller partition modes prove to be more efficient.

Therefore, by using the mode decision scheme of the encoder, we could implement our watermarking algorithm targeting at higher motion activities macroblocks with the best mode 8×8 (with four sub-modes chosen from 4×4 , 8×4 , 8×4 and 8×8). By choosing these smaller partition modes, it is difficult for the Human Visual System to detect the differences, or the distortions introduced by the watermark embedding scheme. Therefore, watermark components are only embedded in the macroblocks of best mode 8×8 with all possible sub-modes (4×4 , 8×4 , 8×4 and 8×8).

In the iterative procedure of the mode decision, the watermarking scheme needs to follow the video coding flow to embed the watermark. After some components of the watermark sequence are embedded in a macroblock, the macroblock best mode is then determined but it may not match what is required. Hence, the macroblock best mode is checked at the end of every encoding run of a macroblock to ensure that the watermark sequence is properly embedded. If the best mode is not what is needed, the embedding of the watermark sequence will have to be restarted.

For example, consider $\mathbf{w} = \{w(f), f=0, \dots, F-1\}$ the binary authentication watermark (a pseudo-random sequence), where F is the number of watermark coefficients. Thus, in one run of the iterative mode decision procedure for a macroblock $\mathbf{B}$, if the current prediction mode o is 8×8 (with four

sub-modes 4×4 , 8×4 , 8×4 and 4×8) (Figure 7.3(d)), let's suppose that the watermark components are embedded starting with the 10th coefficient. As the coefficients represent the residual to code, all zero coefficients are avoided to prevent the video from getting badly distorted. The nonzero quantized DCT coefficients $X_f(u, v)$ are replaced by

$$X_f(u, v) \leftarrow \tilde{X}_f(u, v) = w(f) + 1 \quad \text{if} \quad X_f(u, v) \neq 0. \quad (7.2)$$

In the H.264/AVC standard, the encoder will first carry out the motion estimation and mode decision for modes 16×16 , 16×8 and 8×16 , for the unitary macroblock, and compute the corresponding rate-distortion related costs in Equation (7.1). These functional blocks appear in the blue line block in Figure 7.6. After big partition modes, the encoder will apply the motion estimation and mode decision for mode 8×8 with the sub-modes 4×4 , 8×4 , 8×4 and 8×8 , for four 8 by 8 sub-macroblocks, as shown in the red dotted block. The proposed authentication scheme embeds the watermark components in this phase, as shown in the blocks with red words in Figure 7.6. Assume that six watermark components are embedded into the current macroblock, so the last watermark to be embedded is the 15th coefficient.

After embedding the watermark component $w(f)$ into all the available modes (denoted by $\mathcal{O}$), the best mode $\tilde{o}$ for the marked macroblock $\tilde{\mathbf{B}}$ is selected by minimizing the expression in Equation (7.1)

$$o^* = \arg \min_{\tilde{o} \in \mathcal{O}} \left(D(\tilde{\mathbf{B}}, \tilde{o}) + \lambda R(\tilde{\mathbf{B}}, \tilde{o}) \right), \quad (7.3)$$

where λ , D and R are defined as in Equation (7.1).

After all the possible modes are tested for the minimum rate-distortion cost of the current macroblock, the encoder is checking which partition mode has the minimum cost. If partition mode 8×8 is the best partition mode o^* ,

Figure 7.6: Flow diagram of watermark embedding process (enclosed in the dashed line is the embedding process).

a counter of watermark sequence will be updated. Since the assumption is that the watermark is embedded into the current macroblock starting from the 10th component and total 6 components are inserted, then the algorithm will embed watermark into the next macroblock starting from the 16th watermark component.

However, if the rate-distortion cost of the partition mode 8×8 is not the minimum in the current macroblock, the watermark sequence will have to be re-embedded into the next macroblock. That is, the algorithm will have to embed the watermark components into the next macroblock starting from 10th coefficient again, when the encoder checks the rate-distortion cost of the all possible partition modes. Hence, the algorithm will be more complex and the sequence to embed needs to be scrutinized.

7.3.2 Extraction and Content Authentication

In the decoding process of the video stream, the retrieval of the watermark will be performed. Compared to the embedding process, the watermark extraction is rather straight forward, as shown in Figure 7.7.

During the decoding process of a macroblock, if the best mode of the current macroblock is 8×8 , all the nonzero quantized DCT coefficients (level values of entropy coding) are extracted to form the watermark sequence for authentication in a 1-D order. If the best mode of the current macroblock is not mode 8×8 , the extraction process skips the current macroblock, and tries to extract watermark from the next macroblock.

The extracted 1-D watermark sequence $\hat{\mathbf{w}} = \{\hat{w}(f), f=0, \dots, \hat{F} - 1\}$ is compared to the original authentication watermark information $\mathbf{w} = \{w(f), f=0, \dots, F - 1\}$ to check if the video has been tampered, by using *Correct Bit Rate* as follows

Figure 7.7: Flow diagram of watermark extraction process.

when $\hat{F} \geq F$:

$$CBR(f) = \begin{cases} 1 & \text{if } \hat{w}(f) = w(f) \\ -1 & \text{if } else \end{cases} \quad f = 0, 1, \dots, F-1. \quad (7.4)$$

Or when $\hat{F} < F$:

$$CBR(f) = \begin{cases} 1 & \text{if } \hat{w}(f) = w(f) \\ -1 & \text{if } else \end{cases} \quad f = 0, 1, \dots, \hat{F}-1. \quad (7.5)$$

$$CBR(f) = -1 \quad f = \hat{F}, \hat{F}+1, \dots, F-1. \quad (7.6)$$

Equation (7.5) and Equation (7.6) can be simplified as follows

when $\hat{F} < F$:

$$CBR(f) = \begin{cases} 1 & \text{if } \hat{w}(f) = w(f) \text{ and } 0 \leq f \leq \hat{F}-1 \\ -1 & \text{if } else \end{cases} \quad f = 0, 1, \dots, F-1. \quad (7.7)$$

This correct bit rate has dynamic range from -1 to 1. 1 means the retrieved watermark is exactly the same as the embedded watermark. The lower the rate is achieved, the less correct bits are extracted.

7.4 Theoretical Analysis

7.4.1 Computational Complexity

Initial concern is the increased time involved in watermarking. On the surface, it seems that the time needed will increase. The extra time needed is small compared to the time taken to encode a frame, as shown in Table 7.1 for the fairly motionless *News* and Table 7.2 for the *Stefan* with significant motion

activity. The average increases are less than 5%. This is because, during the encoding process, most of the time is taken up by the motion estimation and motion compensation. The number of computations involved in these functions takes up most of the encoding time. Hence, we can ignore this time factor.

Table 7.1: The average coding time (secs) per frame of the unmarked and marked *News* under different bit rates.

Time(s)	Unmarked	Marked	Difference	Increase
256kbps	20.7	21.3	0.6	3%
512kbps	21.0	21.2	0.2	1%
768kbps	21.6	23.1	0.5	2%

Table 7.2: The average coding time (secs) per frame of the unmarked and marked *Stefan* under different bit rates.

Time(s)	Unmarked	Marked	Difference	Increase
256kbps	21.3	22.1	0.8	4%
512kbps	18.9	19.6	0.7	4%
768kbps	21.3	22.2	0.9	4%

7.4.2 Watermarking Capacity

The watermark capacity per frame is mainly determined by: (a) the motion activity, (b) the distortion introduced by watermark and (c) the *Direct/Skip* prediction modes in B-/P-slices.

All the watermark components are embedded in the macroblock with the best mode 8×8 . The nonzero coefficients in the remaining macroblocks are untouched. Therefore, the watermark capacity is the number of the nonzero coefficients of the macroblocks with the best mode 8×8 . Figure 7.8 shows the capacity of the videos *Foreman* and *News*. The lines with $\circ$ are the number of the nonzero coefficients without the watermark, and the red lines $\times$ are the number of the nonzero coefficients with the watermark.

(a) Foreman

(b) News

Figure 7.8: The number of the nonzero coefficients in the macroblocks with the mode 8×8 without and with watermark.

For the video sequence with significant motion activities, many macroblocks are allocated with the mode 8×8 . Therefore, a significant amount of suitable nonzero coefficients are produced. While if the video contains moderate motion activities, the number of suitable nonzero coefficients decrease. Less watermark components can be embedded into these video sequences. This is illustrated in Figure 7.8 by comparing the capacity of two different video sequence with watermark (red lines). The mean capacity of *Foreman* (69 binary bits per frame) is almost 3 times to the mean capacity of *News* (26 binary bits per frame).

The distortion introduced by the watermark embedding process also affects the watermark capacity. For example, in the current frame if more distortion is introduced by watermarking, after motion estimation and compensation, in the next frame there may be less suitable nonzero coefficients for watermark embedding. This is due to the fact that the other larger partition modes (such as 16×8) may produce the less rate-distortion cost. For example, in Figure 7.8(a), the potential watermark capacity mean per frame without watermark is approximately 9.8 times to the mean capacity with watermark. Thus, decreasing the watermark distortion (such as by using watermark gain adaptive to the local characteristics) can increase the watermark capacity.

7.4.3 Sensitivity against Spatial Tampering

An important and inherent feature of our algorithm is that the embedding location selection is tightly dependent on the mode decision scheme. In our method, since the embedding location selection is based on the best mode of macroblocks (8×8), a small change in the content of a frame can lead to a new mode allocation. That is, the original best mode 8×8 may change to other big partition modes or even direct/skip mode, and the locations of suitable nonzero coefficients change in the video. Even one change of one macroblock's

best mode affects the correct extraction of the embedded watermark in this macroblock. This change is propagated from this period of extracted watermark to the following watermark components, which eventually destroys the extraction synchronization of the entire watermark sequence. Consequentially less correct watermark components can be achieved. Thus, our proposed algorithm can be more sensitive and effective in content tampering detection.

7.4.4 Sensitivity against Temporal Tampering

Another main feature of the H.264/AVC coding standard which is made use of in our algorithm is that the standard supports *Multi-picture Motion-compensated Prediction* [67]. That is, more than one previously coded pictures (≤ 5) can be used as a reference for motion-compensated prediction. Even previously coded B-slices can be references. Therefore, if any of the frames in the video sequence changes (content-wise or position-wise), it is reflected not only in itself, but also in all the subsequent frames. For instance, any alteration such as dropping and reordering of the f th frame is reflected on the f th frame, just as mentioned in Section 7.4.3, that directly changes the subsequent frames which take the f th frame as the prediction reference. Eventually all the sequence frames are affected directly or indirectly. So our algorithm is very secure against all the temporal attacks.

7.5 Experiments and Results

7.5.1 Test Conditions

The proposed watermarking technique has been integrated into the H.264 JM-9.0 reference software [83]. The seven test video sequences: *Foreman*, *Stefan*, *Coastguard*, *Mobile*, *Bus*, *Football* and *News* are used in the experiments. All

video clips are coded in CIF resolution (352×288 pixels) at the frame rate 30 frames/sec and a bit-rate of 512 kbits/sec. The GOP structure comprises IBPBP..., compliant to the *Main Profile* (first frame coded as an I-slice, subsequent frames coded as B- and P-slices, five reference frames used for inter prediction, CABAC entropy coding). A binary watermark sequence is used as the authentication information for our experiments. Generally, a large amount of watermark bits can be inserted into frames with significant movement (such as over 2000 bits for some of the frames in the video sequence *football*). It happens that no bit can be embedded into some of the motionless frames (such as *News*). The capacity varies for different video clips, but for fair performance comparison, the watermark sequence with 150 bits is used in our experiments.

7.5.2 Fidelity after Watermarking

Figure 7.9 shows the frame samples from the unmarked and marked video clips with PSNR (dB). The unmarked and marked video clips are reconstructed from the compressed data without and with the watermarks, respectively. The average PSNR values are computed by comparing the reconstructed video frames to the original raw video frames.

On average, the watermark insertion leads to a decrease of approximately 0.04 dB for all video clips coded at 512 kbits/sec, as shown in Table 7.3. In the experiments, no visible artifacts can be observed in all of the test video sequences.

7.5.3 Results of Authentication Tests

Two categories of video attacks have been applied to the marked video to test the sensitivity of the authentication algorithm: transcoding and common signal processing operations.

In the first group, we utilized the bit-rate reduction, frame-rate reduction,

(a) Unmarked(39.5dB)

(b) Watermarked(39.4dB)

Figure 7.9: The original and watermarked *Foreman*.

Table 7.3: The average PSNR comparison of the unmarked and watermarked frames.

PSNR(dB)	Watermarked	Unmarked	Difference
Foreman	39.4	39.5	0.1
Mobile	37.4	37.4	0.0
News	41.0	41.1	0.1
Bus	37.7	37.8	0.1
Football	39.2	39.2	0.0
Stefan	38.8	38.8	0.0
Coastguard	37.4	37.4	0.0

frame reordering, and frame replacing. During watermarking in the H.264/AVC encoding process, the Lagrangian Optimization technique targets the overall consuming bit-rate at 512 kbits/sec. When transcoding is applied after watermarking, the bit-rates have been reduced to approximately 1/2 of the original bit-rates. During re-encoding with the bit-rate reduction, most high frequency components which represent detailed texture will be discarded. To apply the transcoding of frame-rate reduction, the frame-rate is changed from 30 to 15 frames/sec and the video is re-compressed.

After decoding the marked bitstream, common signal processing attacks are applied to the raw video frame by frame, including 3×3 Gaussian low-pass filtering, circular averaging filtering, unsharpened contrast enhancement, additive Gaussian noise (mean=0, variance= 0.001), cropping (central 85% remains) and rotation (-1 degree). Then the attacked raw video is coded again. Thus, there are actually two attacks applied each time: decoding and re-encoding, and common signal processing attacks.

The authentication security of our watermarking algorithm is represented by the sensitivity against attacks. The *Correct Bit Rate* (as proposed in Equation (7.4) and Equation (7.7)) of the original and extracted watermark sequences are measured and shown in Figures 7.10, 7.11 and 7.12, with the dynamic range from -1 to 1. 1 means the retrieved watermark is exactly the same as the

(a) Bit-rate/frame-rate/bit-rate + frame-rate reduction

(b) Frame Reordering and replacing

Figure 7.10: Sensitivity under transcoding on *Foreman*, *Mobile*, *News*, *Bus*, *Football*, *Stefan* and *Coastguard* (denoted by *For*, *Mo*, *Ne*, *Bu*, *Foo*, *St* and *Co* in the horizontal axis, respectively) (512 kbit/s, CIF-size).

(a) 3×3 Gaussian filter and Circular average filter

(b) Contrast enhancing filter and Gaussian noise

Figure 7.11: Sensitivity under common signal processing on *Foreman*, *Mobile*, *News*, *Bus*, *Football*, *Stefan* and *Coastguard* (denoted by *For*, *Mo*, *Ne*, *Bu*, *Foo*, *St* and *Co* in the horizontal axis, respectively) (512 kbit/s, CIF-size).

Figure 7.12: Sensitivity under cropping and rotation on *Foreman*, *Mobile*, *News*, *Bus*, *Football*, *Stefan* and *Coastguard* (denoted by *For*, *Mo*, *Ne*, *Bu*, *Foo*, *St* and *Co* in the horizontal axis, respectively) (512 kbit/s, CIF-size).

embedded watermark. The lower the rate is, the less correct watermark bits are extracted. From these two figures, almost all the correlation values are less than 0. Thus, our proposed fragile authentication algorithm is effective to detect content tampering.

One special attack, the cutting and pasting attack is also applied to test the security of our authentication system. For example, the *DVC* characters in the area of size 8×48 over the bottom right corner of the *Bus* frame is cut and replaced as shown in Figure 7.13. Even though the square is so small (compared to the frame size 352×288), our algorithm shows the significant sensitivity from the average correlation value equal to -0.261.

7.6 Summary

The proposed hard authentication algorithm performed well in terms of sensitivity against transcoding and common signal processing attacks. The watermarked H.264/AVC video clips maintained the good visual quality and almost

(a) Watermarked

(b) undergone cutting and pasting

Figure 7.13: The watermarked and the attacked *Bus* under cutting and pasting attack.

the same bit-rate. However our algorithm lacks the ability to provide further information necessary to characterize the attack. Therefore, the future work will focus on enhancing the proposed algorithm by localizing the attacked areas.

Chapter 7. PROPOSED HARD AUTHENTICATION OF H.264/AVC

142

STANDARD

Chapter 8

CONCLUSIONS AND RECOMMENDATIONS

8.1 Conclusions

Watermarking is a process of adding some identification information into the host data, such as images, audio, video and documents. The embedded information, known as watermark, can be extracted from the watermarked multimedia contents to provide proofs for various purposes, such as copyright protection and authentication.

In this thesis, new watermarking techniques have been investigated, developed and implemented for still images and the state-of-the-art video coding standard H.264/AVC. The contributions are concluded as follows:

- **A novel proposal for robust image-in-image watermarking**

First we investigated an image watermarking by using the 4×4 integer DCT followed by video watermarking technology. The main purpose of this investigation is to test the performance of this novel transform and extend these techniques to the H.264 standard.

For our proposed approach, a 32×32 intensity watermark image was

embedded into several standard test images based on the integer DCT transform. Feature extraction techniques were applied to make the embedding gain adaptive to the Canny edge characteristics of the test images. Several post-processing techniques were proposed to enhance the robustness of the watermarking scheme. Experimental results showed that these post-processing techniques did have good performance against geometric attacks such as scaling, cropping and rotation attacks.

We studied and compared performances of integer DCT against the Hadamard transform and the various integer DCTs. Although the integer DCT is an approximate transform to the traditional DCT with acceptable errors due to approximation, it still maintains most of the properties of DCT. The image watermarking scheme based on the integer DCT even showed to be more robust than the DCT algorithm against some attacks such as JPEG Compression, Additive Noise and Gaussian Filtering. Moreover, the integer DCT transform offers a significant advantage in shorter processing time and ease of hardware implementation than commonly used DCT techniques.

- **Robust watermarking methods for H.264/AVC**

Investigation of video watermarking techniques is mainly concerned about compressed video of the prior successful coding standards, such as MPEG-2 and MPEG-4. However, digital watermarking has not yet been investigated for the newly developed video coding standard H.264/AVC which is currently the most efficient video coding standard.

In this thesis, a robust video watermarking algorithm for the H.264/AVC coding standard was proposed for copyright protection. A binary sequence acting as an ID number was served as watermark. Also grayscale 2-D watermark patterns such as detailed trademarks or logos were highly

compressed by a proposed grayscale watermark pre-processing before embedding. The watermark bits were inserted on a 4×4 integer DCT block basis. The spread spectrum method is employed to spread watermark bits all over the whole spectrum frame. Watermark embedding gain adaptively changes according to the local motion activities. To be compliant to the H.264 JM encoder, the Lagrangian optimization was used to select the new best mode for the watermarked macroblocks. A slide-window approach for frame accumulation is applied to further improve the robustness of watermark extraction for P-slices. The proposed algorithm can robustly survive the transcoding process and common signal processing attacks, such as bit-rate reduction, Gaussian filtering and contrast enhancement.

- **An Effective Authentication method for H.264/AVC videos**

Authentication is a process whereby an entity proves the identity to another entity. A video authentication system ensures the integrity of digital video, and verifies that the video taken into use has not been tampered. Video authentication is important in many applications such as surveillance, journalism and video broadcast, etc.

For the application of video authentication, we proposed a novel and efficient scheme of fragile video watermarking. In the literature, most fragile video authentication methods were extended from image algorithms to video frame by frame. Our proposed algorithm is more efficient as motion and spatial redundancy have been fully considered. This scheme made an accurate usage of best mode decision strategy based on the tree-structured motion compensation, motion estimation and Lagrangian optimization of the standard. Authentication information was embedded strictly based on the best mode in the sense that if watermarked video content undergoes

any spatial and temporal attacks, the scheme can detect the tampering by the sensitive mode change. The experimental results proved the effectiveness of the algorithm against many transcoding and signal processing attacks.

8.2 Recommendations for Future Work

- **On robust image-in-image watermarking**

For the proposed image algorithm, feature extraction techniques with consideration of the Human Visual System (HVS) are an important issue for our future work. For example, if more efficient feature extracting techniques can be applied to the selecting location step in the watermark embedding, our algorithm can be more efficient, and can survive more attacks such as rotation and translation. This would bring about more computation complexity.

- **On robust watermarking methods for H.264/AVC videos**

For H.264/AVC coding standard, the compression performance is so efficient that there is not much temporal and spatial redundancy in the residual error after prediction, which is usually suitable for watermark insertion. Thus watermark capacity is the main issue for the H.264/AVC watermarking algorithm. In our DCT-domain watermarking scheme, the watermark bits were spread over the whole frames and repeatedly embedded into successive frames. Consequently, the watermark capacity is limited by the frame size and redundancy after prediction. The 3-D interleaving technique proposed by Shi *et al.* [88], [82] is a promising solution. It is desired to spread the bits pseudo-randomly over a period of frames to increase the capacity and the robustness. Their experimental results show that the performance of embedded data has been dramatically improved

by interleaving the data in a three dimensional way. Further investigation on this technique to incorporate into the H.264/AVC watermarking methods should be greatly meaningful.

- **On fragile authentication method for H.264/AVC videos**

Our authentication method performed well in terms of sensitivity against transcoding and common signal processing attacks. However our algorithm lacks the ability to provide further information necessary to characterize the attack or to localize the tampered areas and further self-recover these areas. Better tamper localization without sacrifice of security remains an issue to be studied. In the video authentication algorithm proposed by Mobasser *et al.* [89], the scheme was not only able to detect tampering, but also capable of identifying cuts and splices both in length and duration. Winne *et al.* [90] also developed an MPEG-4 authentication system that allowed tamper characterization of time-based attacks. Therefore, the future work will focus on enhancing the proposed algorithm by localizing the attacked areas.

Publication List

1. J. Zhang and A. T. S. Ho, “An efficient digital image-in-image watermarking algorithm using the integer discrete cosine transform,” *Watermarking and Security II, IEEE ICICS-PCM2003*, Singapore, Dec 15-18, 2003.
2. J. Zhang and A. T. S. Ho, “Robust digital image-in-video watermarking for the emerging H.264/AVC standard,” *IEEE 2005 Workshop on Signal Processing Systems (SIPS’05)*, Athens, Greece, Nov. 2-4, 2005.
3. J. Zhang and A. T. S. Ho, “Efficient robust watermarking of compressed 2-D grayscale patterns for H.264/AVC,” *International Workshop on Multimedia Signal Processing (MMSP05)*, Shanghai, China, Oct. 30-Nov. 2, 2005.
4. J. Zhang and A. T. S. Ho, “An efficient authentication method for H.264/AVC,” *2nd European Workshop on the Integration of Knowledge, Semantic and Digital Media Technologies (EWIMT2005)*, London, Nov. 30 - Dec. 1, 2005.
5. J. Zhang, A. T. S. Ho, G. Qiu and P. Marziliano, “Robust video watermarking of H.264/AVC,” *IEEE Trans. On Circuits and Systems Part II*, Vol. 54, no. 2, pp.205 - 209, Feb. 2007.
6. J. Zhang and A. T. S. Ho, “Efficient video authentication for H.264/AVC,” *2006 International Conference on Innovative Computing, Information and Control (ICICIC2006)*, 2006.

Bibliography

- [1] 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.
- [2] Iain E. G. Richardson, *H.264 and MPEG-4 video compression: video coding for next-generation multimedia*, Wiley, England, 2003.
- [3] T. Wiegang, G. J. Sullivan, G. Bjontegaard, 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, July 2003.
- [4] E. A. Taub, "A TV that prints still photos," *The New York Times*, vol. 3, Jan. 2004.
- [5] A. Z. Tirkel, G. A. Rankin, R. M. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne, "Electronic watermark," in *Digital Image Computing, Technology and Applications (DICTA'93)*, Macquarie University, Sydney, 1993, pp. 666–673.
- [6] I. J. Cox and M. L. Miller, "The first 50 years of electronic watermarking," *Journal of Applied Signal Processing*, pp. 126–132, 2002.
- [7] F. Hartung and B. Girod, "Watermarking of uncompressed and compressed video," *Signal Processing (Special issue on Watermarking)*, vol. 66, no. 3, pp. 283–301, May 2003.
- [8] S. Arena, M. Caramma, and R. Lancini, "Digital watermarking applied to MPEG-2 coded video sequences exploiting space and frequency masking," in *IEEE International Conference on Image Processing (ICIP 2000)*, Sep. 2000, pp. 1987–1990.
- [9] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of DCT encoded images and video," *IEEE Trans. Image Processing*, pp. 148–158, Jan. 2001.
- [10] 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.

- [11] 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.
- [12] G. Qiu, P. Marziliano, A. T. S. Ho, D. J. He, and Q. B. Sun, "A hybrid watermarking scheme for H.264/AVC video," in *Proc. the 17th International Conference on Pattern Recognition*, Cambridge, UK, August 2004.
- [13] A. T. S. Ho, J. Shen, and S. H. Tan, "Robust digital image-in-image watermarking algorithm using the fast Hadamard transform," in *Proceedings of SPIE*, Seattle, Washington, USA, July 2002, vol. 4793.
- [14] M. D. Swanson, B. Zhu, and A. H. Tewfik, "Data hiding for video-in-video," in *Proceedings of International Conference on Image Processing 1997*, Oct. 1997, vol. 2, pp. 676–679.
- [15] C.-Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui, "Rotation, scale, and translation resilient watermarking for images," *IEEE Trans. Image Processing*, vol. 10, no. 5, pp. 767–782, 2001.
- [16] "Stirmark. [online] available: <http://www.petitcolas.net/fabien/watermarking/stirmark/>," .
- [17] "Checkmark. [online] available: <http://watermarking.unige.ch/checkmark/>," .
- [18] I. J. Cox, M. L. Miller, and J. A. Bloom, *Digital Watermarking*, Academic Press, USA, 2002.
- [19] C.-W. Tang and H.-M. Hang, "A feature-based robust digital image watermarking scheme," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 950–959, Apr. 2003.
- [20] M. Yeung, B.-L. Yeo, and M. Holliman, "Digital watermarks: Shedding light on the invisible," *Micro, IEEE*, vol. 18, no. 6, pp. 32–41, 1998.
- [21] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-embedding and watermarking technologies," *Proceedings of the IEEE*, vol. 86, no. 6, pp. 1064–1087, 1998.
- [22] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A digital watermark," in *Proc. Int. Image Processing (ICIP)*, 1994, pp. 86–89.
- [23] A. Z. Tirkel, R. M. van Schyndel, and C. F. Osborne, "A two-dimensional watermark," in *Digital Image Computing, Technology and Applications (DICTA'93)*, 1993.

- [24] M. Kutter, F. Jordan, and F. Bossen, "Digital signature of color images using amplitude modulation," in *Proc. Electronic Imaging 1997 (EI 97)*, Feb. 1997, vol. 7, pp. 326–332.
- [25] P. Bas, J.-M. Chassery, and B. Macq, "Geometrically invariant watermarking using feature points," *IEEE Trans. Circuits Signal Processing*, vol. 11, no. 9, pp. 1014–1028, Sept. 2002.
- [26] M. Kuttur and S. Winkler, "A vision-based masking model for spread-spectrum image watermarking," *IEEE Trans. Image Processing*, vol. 11, no. 1, pp. 16–25, Jan. 2002.
- [27] E. Koch and J. Zhao, "Towards robust and hidden image copyright labeling," in *Nonlinear Image Processing Workshop*, Thessaloniki, Greece, 1995.
- [28] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for images, audio and video," in *Proc. IEEE Int. Conf. Image Processing (ICIP 96)*, Lausanne, Switzerland, September 1996, pp. 666–673.
- [29] I. J. Cox, J. Kilian, T. Leighton, and T. Shamoan, "Secure spread spectrum watermarking for multimedia," *IEEE Trans. Image Processing*, pp. 1673–1687, Dec. 1997.
- [30] 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.
- [31] 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.
- [32] H. S. Malvar and D. A. F. Florencio, "Improved spread spectrum: a new modulation technique for robust watermarking," *IEEE Transactions on Signal Processing*, vol. 51, no. 4, pp. 898–905, April 2003.
- [33] A. T. S. Ho, J. Shen, and S. H. Tan, "Robust digital image-in-image watermarking algorithm using the fast Hadamard transform," in *SPIE's 47th Annual Meeting on Optical Science and Technology*, Seattle, USA, 2002.
- [34] R. C. Gonzalez and R. E. Woods, *Digital image processing*, Prentice Hall, 2002.
- [35] G. Doerr and J.-L. Dugelay, "A guided tour of video watermarking," *Signal Processing: Image Commun.*, vol. 18, no. 7, pp. 263–282, 2003.

- [36] J. Dittmann, M. Stabenau, and R. Steinmetz, "Robust MPEG video watermarking technologies," in *In Proc. of ACM Multimedia*, 1998, pp. 71–80.
- [37] F. Jordan, M. Kutter, and T. Ebrahimi, "Proposal of a watermarking technique for hiding/retrieving datas in compressed and decompressed video," in *in ISO/IEC Doc. JTC1/SC29/WG11/MPEG97/M2281*, July 1997.
- [38] J. Zhang, J. Li, and L. Zhang, "Video watermark technique in motion vector," in *in XIV Brazilian Symp. on Computer Graphics and Image Processing*, Oct. 2001, pp. 179–182.
- [39] Y. Bodo, N. Laurent, and J.-L. Dugelay, "Watermarking video, hierarchical embedding in motion vectors," in *Proc. IEEE Int. Conf. Image Processing (ICIP'03)*, Sept 2003, vol. 2, pp. 739–742.
- [40] G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Real-time labeling of MPEG-2 compressed video," *J. Vis. Commun. Image Represent.*, , no. 4, pp. 256–270, 1998.
- [41] G. C. Langelaar, "Real-time watermarking techniques for compressed video data," in *PhD. thesis*, Delft Univ. of Technology, 2000.
- [42] P. K. Atrey, W.-Q. Yan, E.-C. Chang, and M. S. Kankanhalli, "A hierarchical signature scheme for robust video authentication using secret sharing," in *in Proc. the 10th International Multimedia Modelling conference*, 2004.
- [43] B. B. Zhu, M. D. Swanson, and A. H. Tewfik, "When seeing isn't believing," *IEEE Signal Processing Magazine*, vol. 2, pp. 988–991, May 2004.
- [44] P. W. Wong and N. Memon, "Secret and public key image watermarking schemes for image authentication and ownership verification," *IEEE Trans. Image Processing*, vol. 10, no. 10, pp. 1593–1601, 2001.
- [45] S. Walton, "Information authentication for a slippery new age," *Dr. Dobbs Journal*, vol. 20, no. 4, pp. 18–26, 1995.
- [46] M. M. Yeung and F. Mintzer, "An invisible watermarking technique for image verification," in *in Proc. IEEE Int. conf. Image Processing*, 1997, pp. 680–683.
- [47] C. W. Honsinger, P. W. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," *U.S. Patent*, , no. 6,278,791, 2001.
- [48] J. Frodroch, M. Goljan, and R. Du, "Security watermarking of multimedia contents," in *In Proc. SPIE of ACM Multimedia*, San Jose, CA, 2001, vol. 3971, pp. 197–208.

- [49] "Draft ITU-T recommodation and final draft international standard of joint video specification (ITU-T Rec. H.264 ISO/IEC 14 496-10 AVC) joint video team (JVT)," Tech. Rep. JTV-G050, Mar. 2003.
- [50] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G.J.Sullivan, "Performance comparison of video coding standards using Lagrangian coder control," in *Proc. IEEE Int. Conf. Image Processing*, 2002, vol. 4, pp. IV-3728-IV-3731.
- [51] T. Wiegand, H. Schwarz, A. Joch, F. Kossentini, and G. J. Sullivan, "Rate-constrained coder control and compression of video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688-703, July 2003.
- [52] 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, September 2001.
- [53] A. Joch, F. Kossentini, and P. Nasiopoulos, "A performance analysis of ITU-T draft H.26L video coding standard," in *Packet Video 2002*, Pittsburgh, PA, April 2002.
- [54] T. Stockhammer, M. Hannuksela, and S. Wenger, "H.26L/JVT coding network abstraction layer and IP-based transport," in *IEEE ICIP2002*, Rochester, New York, September 2002, pp. 485-488.
- [55] M. Karczewicz and R. Kurceren, "The SP- and SI-frames design for H.264/AVC," *IEEE Transaction on circuits and systems for video technology*, vol. 13, no. 7, pp. 637-644, July 2003.
- [56] T. Halbach, "The H.264 video compression standard," in *NORSIG-2003*, Bergen, Norway, October 2003.
- [57] M. Flierl and B. Girod, "Generalized B picture and the draft H.264/AVC video-compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 587-597, July 2003.
- [58] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: overview and introduction to the fidelity range extensions," in *SPIE Conference on Applications of Digital Image Processing XXVII*, August 2004, pp. 53-74.
- [59] P. List, A. Joch, J. Lainema, G. Bjontegaard, and M. Karczewicz, "Adaptive deblocking filter," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 614-619, July 2003.

- [60] A. Hallapuro, M. Karczewicz, and H. Malvar, "Low complexity transform and quantization - part I: basic implementation," in *JVT document JTV-B038*, Geneva, February 2002.
- [61] S. W. Golomb, "Run-Length encoding," *IEEE Trans. on Information Theory*, vol. IT-12, pp. 399–401, December 1966.
- [62] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, July 2003.
- [63] 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, April 1996.
- [64] G. J. Sullivan and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, November 1998.
- [65] A. Joch, F. Kossentini, H. Schwarz, T. Wiegand, and G. Sullivan, "Performance comparison of video coding standards using Lagrangian coder control," in *Proc. IEEE Int. Conf. Image Processing (ICIP)*, Rochester, NY, Oct. 2002, pp. 501–504.
- [66] C. Sullian and T. Wiegand, "Rate-distortion optimization for video compression," *IEEE Signal Processing Mag.*, vol. 15, pp. 74–90, Nov. 1998.
- [67] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coder control," Thessaloniki, Greece, Oct. 2001, vol. 3, pp. 542–545.
- [68] R. Schafer, T. Wiegand, and H. Schwarz, "The emerging H.264/AVC standard," in *NORSIG-2003*, Bergen, Norway, October 2003.
- [69] F. Pan and K. P. Lim, "H.26L-the newest standard in digital video compression," in *ItsAsia 2003-A Tale of 2 Cities*, Singapore, Feb. 2003.
- [70] S. C. Chan and K. L. Ho, "A new two-dimensional fast cosine transform algorithm," *IEEE Transactions on Signal Processing*, vol. 39, no. 2, pp. 481–485, February 1991.
- [71] C. Loeffler, A. Ligtenberg, and G. S. Moschytz, "Practical fast 1-D DCT algorithms with 11 multiplications," *Proceedings of IEEE International Conference on Acoustics, Speech and Signal Processing 1989*, vol. 2, pp. 988–991, May 1989.
- [72] T. D. Tran, "The BinDCT: Fast multiplierless approximations of the DCT," *IEEE Signal Processing Letters*, September 2001.

- [73] W. K. Cham, "Development of integer cosine transforms by the principle of dyadic symmetry," *IEE Proceedings*, vol. 136, no. 4, pp. 276–282, August 1989.
- [74] W. K. Cham and P. C. Yip, "Integer sinusoidal transforms for image processing," *International Journal of Electronics*, vol. 70, no. 6, pp. 1051–1030, June 1991.
- [75] T. D. Tran, "The BinDCT: Fast multiplierless approximations of the DCT," *IEEE Signal Processing Letters*, vol. 7, no. 6, pp. 141–144, June 2000.
- [76] J. Liang and T. D. Tran, "Fast multiplierless approximations of the DCT with the lifting scheme," *IEEE Transactions on Signal Processing*, vol. 49, no. 12, pp. 3032–3044, December 2001.
- [77] J. Liang, T. D. Tran, and P. Topiwala, "A 16-bit architecture for H.26L, treating DCT transforms and quantization," in *VCEG-M16.doc*, 13th Meeting of ITU-T VCTEG, Austin, Texas, USA, April 2001.
- [78] J. Liang, T. D. Tran, and P. Topiwala, "FastVDO's unified 16-bit transform/quantization approach," in *JVT-B018.doc*, 2nd Meeting of JVT of ISO/IEC MPEG & ITU-T VCTEG, Geneva, CH, January 2002.
- [79] Z. Wang and A. C. Bovik, "A universal image quality index," *IEEE Signal Processing Letters*, vol. 9, no. 3, pp. 81–84, March 2002.
- [80] G. Qiu, P. Marziliano, J. Zhang, and A. T. S. Ho, "Robust watermarking of H.264/AVC video for copyright protection (Part I)," (resubmitted) *IEEE Trans. on Circuits and Systems II: Express Briefs*, 2005.
- [81] J. Zhang, A. T. S. Ho, G. Qiu, and P. Marziliano, "Robust watermarking of H.264/AVC video for copyright protection (Part II)," (resubmitted) *IEEE Trans. on Circuits and Systems II: Express Briefs*, 2005.
- [82] Y. Q. Shi, X. M. Zhang, Z. C. Ni, and N. Ansari, "Interleaving for combating bursts of errors," *IEEE Circuits and System Mag.*, vol. 4, no. 1, pp. 29–42, 2004.
- [83] "H.264/AVC joint model 7.6 (JM-7.6) reference software.[online] available:ftp://imtc.org/jvt-exports/reference_software/," .
- [84] S. Bhattacharya, T. Chattopadhyay, and A. Pal, "A survey on different video watermarking techniques and comparative analysis with reference to H.264/AVC," in *IEEE Tenth International Symposium on Consumer Electronic, ISCE '06*, 2006, pp. 1–6.

-
- [85] T.-H. Chen, S.-H. Liu, H.-X. Yao, and W. Gao, "Robust video watermarking based on DC coefficients of selected blocks," in *Proc. of the 4th Int. Conf. on Machine Learning and Cybernetics*, Guangzhou, P. R. China, August 2005.
 - [86] M. Yang and N. Bourbakis, "A high bitrate information hiding algorithm for digital video content under H.264/AVC compression," in *48th Midwest Symposium on Circuits and Systems*, August 2005, vol. 2, pp. 935–938.
 - [87] T. Wiegang, H. Schwarz, A. Joch and F. Kossentini, and G. Sullivan, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 688–703, July 2003.
 - [88] 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 *in Proc. IEEE Int'l Sump. Circuits and Systems (ISCAS'03)*, May 2003, vol. 2, pp. 924–927.
 - [89] B. G. Mobasser, M. J. Sieffert, and R. J. Simard, "Content authentication and tamper detection in digital video," in *in Proc. IEEE International Conference on Image Processing, 2000*, Sept. 2000, vol. 1, pp. 458–461.
 - [90] D. A. Winne, H. D. Knowles, D. R. Bull, and C. N. Canagarajah, "An investigation of an MPEG-4 embedded spatial digital watermark developed for tamper detection and characterization," in *in Proc. IEEE International Conference on Consumer Electronics, 2003*, June 2003.