

H24/3297

MONASH UNIVERSITY  
THESIS ACCEPTED IN SATISFACTION OF THE  
REQUIREMENTS FOR THE DEGREE OF  
DOCTOR OF PHILOSOPHY  
ON..... 12 April 2002 .....

.....  
Gov Sec. Research Graduate School Committee

Under the copyright Act 1968, this thesis must be used only under the normal conditions of scholarly fair dealing for the purposes of research, criticism or review. In particular no results or conclusions should be extracted from it, nor should it be copied or closely paraphrased in whole or in part without the written consent of the author. Proper written acknowledgement should be made for any assistance obtained from this thesis.

## ADDENDUM

p v line 6: insert "a" before communication.

p v line 17: delete "into" and replace with "within".

p v line 33: delete "secrecy" and replace with "unknown".

p v line 35: delete "to" and replace with "for".

p v line 36: delete "in" and replace with "within".

p vii line 4: delete "Enabling", capitalize a in "Applications".

p vii line 9: delete "here".

p vii line 20: delete "The capability to hide large amounts of data enables hidden communication." and replace with "The algorithm has the ability to hide large amounts of data that will enable hidden communications".

p vii para 3: should be read "Most of the work discussed in Chapters 4-7 has been published as papers at international conferences. The list of publications is included in this thesis".

p viii at the end of para 1 "Published papers", insert the following:

- [5] N. Abdulaziz and K. K. Pang, "Coding Techniques for Data Hiding in Images", Published at the Sixth international Symposium on Signal Processing and its Applications ISSPA 2001, 13-16 August 2001, Kuala-Lumpur, Malaysia, pp. 170-174.
- [6] N. Abdulaziz and K. K. Pang, "The Effect of Channel Coding in Data Embedding in Images," Published at Multispectral Image Processing and Pattern Recognition (MIPPR 2001), 22-24 October 2001, Wuhan, China, pp. 27-31.
- [7] N. Abdulaziz and K. K. Pang, "Wavelet Transform and Channel Coding for Data Hiding in Video," Published at the International Conference on Info-tech & Info-net, ICII2001, Beijing, Oct. 29-Nov. 1, 2001, pp.791-796.
- [8] N. Abdulaziz, A. Glass, and K. Pang, "Embedding Data in Images Using Turbo Coding," Published at the 6<sup>th</sup> International Symposium on DSP for Communication Systems (DSPCS'02), Sydney-Manly, 28-31 January 2002.

p viii delete para 2 "Accepted for publication".

p viii replace para 3 "Currently submitted papers" with:

Currently submitted papers:

- [1] N. Abdulaziz and K. K. Pang, "Coding techniques for data embedding application", submitted to SPIE Journal of Electronic Imaging (JEI).

p 177 line 8: delete "of" and replace with "for".

p 178 line 12: delete "into" and replace with "within".

p 178 line 13: add "then" at the beginning of the line.

p 179 line 4: add "then" before "MPEG".

p 180 line 8: add "other" before "than".

p 181 line 1: delete "his" and replace with "its".

p 181 line 3: delete "has" and replace with "have".

# **Digital Watermarking and Data Hiding in Multimedia**

**Nidhal Kadhim Abdulaziz**

BSc (Eng.), MSc

A thesis submitted for the degree of

Doctor of Philosophy

in the

Department of Electrical and Computer Systems Engineering

Monash University

Clayton, Victoria 3168, Australia

June 2001

*To my children,  
Noor, Russul, and Hummam,  
and  
to my husband Abdullatif*

## CONTENTS

SUMMARY .....	v
PREFACE .....	vii
List of Publications .....	viii
Acknowledgements .....	ix
Glossary .....	x
Chapter 1 .....	1
Introduction .....	2
1.1 TERMINOLOGY .....	5
1.2 RESEARCH OBJECTIVES .....	5
1.3 ORGANIZATION OF THE THESIS .....	7
Chapter 2 .....	9
A Review of Data Embedding (Digital Watermarking) .....	10
2.1 WATERMARKING .....	10
2.2 APPLICATION AREAS .....	12
2.3 GENERAL REQUIREMENTS .....	14
2.4 RELATED WORK ON WATERMARKING SCHEMES .....	17
2.4.1 <i>Spatial Watermarking: Simple Systems</i> .....	18
2.4.2 <i>Basic Spread Spectrum Approach</i> .....	20
2.4.3 <i>Other Spatial Domain Watermarks</i> .....	21
2.4.4 <i>Fractal Operations</i> .....	22
2.4.5 <i>Pattern Overlaying</i> .....	22
2.4.6 <i>Spectral Watermarking: A Spread Spectrum Watermark Embedded in the DCT Domain</i> .....	24
2.4.7 <i>A Linear Combination of Marked and Unmarked Images</i> .....	25
2.4.8 <i>Sub-band Watermarking</i> .....	26
2.4.9 <i>Other Transform-Based Approaches</i> .....	27
2.4.10 <i>Watermarking using the Lapped Orthogonal Transform (LOT)</i> .....	30
2.5 PERCEPTUAL WATERMARKING BASED ON IMAGE-ADAPTABILITY .....	31
2.6 MARKING TEXT DOCUMENTS .....	35

2.7	COMMERCIAL SOFTWARE .....	36
2.8	SUMMARY.....	36
Chapter 3 .....		39
Image Embedding in the Wavelet Domain .....		40
3.1	DISCRETE WAVELET TRANSFORM: A BRIEF REVIEW .....	42
3.1.1	<i>Wavelets</i> .....	42
3.1.2	<i>Why use Haar Wavelets?</i> .....	44
3.1.3	<i>Haar Wavelet Transform</i> .....	45
3.2	EMBEDDING PRINCIPLE.....	48
3.3	EXTRACTION PRINCIPLE.....	52
3.4	SIMILARITY TO DATA COMMUNICATION .....	54
Chapter 4 .....		57
Data Embedding using Source and Channel Coding .....		58
4.1	DATA EMBEDDING USING VECTOR EMBEDDING.....	59
4.2	SOURCE CODING: VECTOR QUANTIZATION .....	62
4.3	BLOCK CODING.....	66
4.3.1	<i>Why use Error-Control Coding?</i> .....	67
4.3.2	<i>BCH Codes</i> .....	67
4.3.3	<i>Reed-Solomon Codes</i> .....	69
4.3.4	<i>Description of Codes by Generator Matrices</i> .....	69
4.3.5	<i>Hard-Decision Decoding</i> .....	72
4.3.6	<i>Probability of Codeword Error</i> .....	72
4.4	INTERLEAVER .....	73
4.5	IMPLEMENTATION AND EXPERIMENTAL RESULTS .....	74
4.5.1	<i>Lossless Data Embedding</i> .....	76
4.5.2	<i>Embedding Images in Images</i> .....	82
4.5.3	<i>Embedding Color Images</i> .....	86
4.6	SUMMARY .....	89
Chapter 5 .....		95
Data Embedding using Convolutional Coding .....		96
5.1	NEED FOR CHANNEL CODING IN DATA EMBEDDING.....	97
5.2	GENERAL EMBEDDING SYSTEM.....	99
5.3	CONVOLUTIONAL CODING.....	101
5.4	REPRESENTATION OF CONVOLUTIONAL CODES .....	104

5.4.1	Encoder Block Diagram.....	104
5.4.2	Generator Representation.....	105
5.4.3	State Diagram Representation of Convolutional Codes.....	106
5.4.4	Trellis Representation of Convolutional Codes.....	106
5.5	DISTANCE STRUCTURE OF A CONVOLUTIONAL CODE.....	107
5.6	DECODING TECHNIQUES OF CONVOLUTIONAL CODES.....	113
5.6.1	The Viterbi Algorithm.....	113
5.7	EVALUATING ERROR PROBABILITY USING THE TRANSFER FUNCTION BOUND.....	116
5.7.1	Performance of Hard-Decision Decoding.....	116
5.7.2	Performance of Soft-Decision Decoding.....	118
5.8	CONCATENATED CODES .....	121
5.9	MODULATION CODES (TRELLIS CODES).....	122
5.10	TURBO CODES .....	127
5.10.1	Turbo Encoder.....	128
5.10.2	Decoding of Turbo Codes.....	130
5.11	IMPLEMENTATION AND EXPERIMENTAL RESULTS .....	133
5.11.1	Convolutional Codes .....	134
5.11.2	Concatenated Codes.....	142
5.11.3	Turbo codes.....	143
5.12	SUMMARY.....	145
<b>Chapter 6</b> .....		<b>149</b>
<b>Hidden Data Extraction without Original Host</b> .....		<b>150</b>
6.1	EMBEDDING TECHNIQUE .....	151
6.2	EXTRACTION TECHNIQUE .....	155
6.3	WATERMARK DETECTION.....	156
6.4	IMPLEMENTATION AND EXPERIMENTAL RESULTS .....	157
6.5	SUMMARY.....	160
<b>Chapter 7</b> .....		<b>163</b>
<b>Data Embedding in Video</b> .....		<b>164</b>
7.1	ESTABLISHED WORK .....	164
7.2	IMAGES IN VIDEO.....	167
7.3	PROPOSED TECHNIQUE .....	168
7.4	IMPLEMENTATION AND EXPERIMENTAL RESULTS .....	169
7.5	SUMMARY.....	172

Chapter 8.....	176
Conclusions and Future Work.....	177
8.1 CONCLUSIONS.....	177
8.2 SUMMARY OF CONTRIBUTIONS.....	178
8.3 DIRECTIONS FOR FURTHER RESEARCH .....	179
8.3.1 <i>Robustness to Signal Manipulation</i> .....	179
8.3.2 <i>Information-Theoretic Model</i> .....	180
8.3.3 <i>Applications</i> .....	181
References .....	183

## SUMMARY

In the past decade there has been an explosion in the use and distribution of digital multimedia data. Electronic commerce applications and on-line services are rapidly being developed. Although digital data has many advantages over analog data, service providers are reluctant to offer services in digital format because they fear unrestricted duplication and illegal publication of copyrighted material.

Data hiding may be considered as communication through a watermarking channel, or secret communication. Data embedding in multimedia could help in providing proof of origin and distribution of content. This communication could control access, provide customized delivery, and provide solutions for pay-per-view implementations. This work is concerned with the development and use of digital watermarking technology to embed signature data within multimedia content. This is done by exploring the application of channel coding techniques known for digital communication to effectively improve the performance of the data embedding system. One of the main contributions of this thesis is the development of methodologies for large quantity data embedding.

In this thesis we have developed several new techniques for robust data embedding into image and video data. These techniques enable embedding large amounts of data and facilitate signature recovery in the absence of the original host. The watermarking problem is treated as a general noisy communication channel problem. The host image constitutes the channel for transmission of the watermark data and is subject to various types of attacks. The attacks such as lossy compression, enhancements, or transformations can be treated as noise addition. The application of the well-known channel coding techniques in digital communication can be employed to effectively improve the performance of the data embedding system. The proposed methods compress the signature data before embedding using vector quantization and then apply channel coding to the data to improve the reliability of watermark detection. The channel codes used in this research comprises of block codes such as BCH and Reed-Solomon, the other codes used were the convolutional codes, trellis codes, concatenated codes, and turbo codes. The above scheme has two layers of security, which result from the variability in the source and channel codebooks chosen. It is practically impossible for unauthorized persons, even if they know the algorithm to pirate the hidden information. As these two codebooks are secrecy only to authorized persons and not embedded in the image in any form or shape.

Several interesting applications of these embedding methods to lossless text data recovery from lossy compressed images, and images in images and video hiding, are presented.

Maximum likelihood (ML) decoding using Viterbi algorithm is employed in extracting the hidden information for the case of convolutional codes, and a

maximum a posteriori (MAP) probability algorithm is utilized in the case of turbo codes. Results were compared to the uncoded messages and the improvement in the performance is quite noticeable. The algorithms were also tested against JPEG/MPEG compression and noise attacks. The signature data was recognizable even after high compression ratios and large noise addition.

In summary, the methods presented in this dissertation advance the current data hiding technology both in terms of the quantity of the data that can be hidden (up to 25% compared to 1% reported in the literature), and the quality of the embedded and recovered data even under significant JPEG/MPEG compression. Finally, as the work presented here is general, it can be extended to other types of host and signature data.

## PREFACE

Our main objective is to develop techniques for embedding (hiding) large amounts of multimedia data, such as text, images, audio, and video, in images and in video. Hence, the requirements are different from typical digital watermarking for data authentication only. Enabling applications for such large scale data hiding include embedded control to track the use of a video clip in pay-per-view applications, hidden communications of text (e-mail), voice and visual data, to mention a few. The capability to hide large amounts of data will also enable robust hiding of digital watermarks by introducing redundancies in the data.

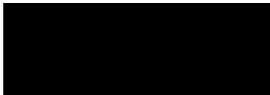
The hosts considered here in our experimental results are images and video. Generalized algorithms are presented for hiding text or an image inside a host image, as well as image/video sequences. In general, the proposed algorithms can be extended readily to other multimedia sources.

In order to accomplish the above objectives, we have developed several new techniques for robust data hiding in images and video. The main contribution of this thesis is towards enhancing the functionality of data embedding, and to the development of robust methods for hiding. The emphasis of these techniques is to improve both the quality of the embedded data and the quantity and quality of the recovered signature data. The proposed methods are based on embedding the data in a transform domain, such as the discrete wavelet transform, and the use of error-control coding for robust recovery. The capability to hide large amounts of data enables hidden communication. Compared to the techniques in digital watermarking, which can embed data at about 1%, the proposed method can embed images up to 25% of the host data size. Since we can embed a significantly larger number of bits for a given host, it is possible to make the embedding resistant to typical signal/image processing operations such as compression.

Most of the work discussed in Chapters 4 and 5 has been published as papers in international conferences. Effort has been made to publish and present the work discussed in Chapters 6 and 7. The list of publications is included in this thesis.

Finally, I declare that

This thesis contains no material which has been accepted for the award of any other degree or diploma in any university or institute, and  
To the best of my knowledge and belief, it contains no material previously published or written by another person, except when due reference is made in the text of the thesis.

  
(Nidhal K. Abdulaziz)

## List of Publications

### Published papers:

- (1) N. Abdulaziz and K. K. Pang, "Source and channel coding approach to data hiding," Proceedings of SPIE, Visual Communications and Image Processing 2000 Conference, 20-23 June 2000, Perth, Australia, pp. 1526-1535.
- (2) N. Abdulaziz and K. K. Pang, "Robust data hiding for images," Published at World Computer Congress WCC 2000, International Conference on Communications Technology (ICCT 2000), 21-25 August 2000, Beijing, China, vol. 1, pp. 380-383.
- (3) N. Abdulaziz and K. K. Pang, "Performance evaluation of data hiding system using wavelet transform and error-control coding," IEEE International Conference on Image Processing 2000 (ICIP 2000), 10-13 September 2000, Vancouver, Canada.
- (4) N. Abdulaziz and K. K. Pang, "Data embedding using trellis coding," International Conference on Communication, Computer and Power (ICCCP'01), February 12-14, 2001, Muscat, Sultanate of Oman.

### Accepted for Publication:

- (1) N. Abdulaziz and K. K. Pang, "Coding Techniques for Data Hiding in Images", Accepted for publication, Sixth international Symposium on Signal Processing and its Applications ISSPA 2001, 13-16 August 2001, Kuala-Lumpur, Malaysia.
- (2) N. Abdulaziz and K. K. Pang, "The Effect of Channel Coding in Data Embedding in Images," Accepted for publication at Multispectral Image Processing and Pattern Recognition (MIPPR 2001), 22-24 October 2001, Wuhan, China.

### Currently submitted papers:

- (1) N. Abdulaziz and K. K. Pang, "Source and Channel Coding Techniques for Data Hiding Applications," Submitted to the Journal of Signal processing, Image Communications.
- (2) N. Abdulaziz and K. K. Pang, "Wavelet Transform and Channel Coding for Data Hiding in Video," International Conference on Info-tech & Info-net, ICI2001, Beijing, Oct. 29-Nov. 1, 2001.
- (3) N. Abdulaziz, A. M. Glass, and K. K. Pang, "Digital Watermarking and Data Embedding in Images and Video," The Third Middle East Symposium on Simulation and Modeling (MESM'2001), September 3-5, 2001, Amman, Jordan.

## Acknowledgements

I would like to sincerely thank my supervisor, Reader Dr. K. Khee Pang, for his guidance, feedback, and financial support during the development of this work. His confidence in my abilities as a researcher is greatly appreciated.

My appreciation to A/Professor W. A. Brown and Dr. T. Czaszcejko for their encouragement and administrative support during my candidature in the Department of Electrical and Computer Systems Engineering at Monash University.

I wish also to acknowledge the Head of the Department, Professor Greg Egan, for providing me with financial support during my study, and for suggesting to work with Dr. Khee Pang. I would like also to thank A/Professor Henry Wu from the School of Computer Science and Software Engineering, for bringing the problem of digital watermarking to my attention.

I am most grateful to my previous supervisors, Professor J. Godfrey Lucas from University of Western Sydney (Nepean), and A/Professor Irena Cosic from the Department of Electrical and Computer systems Engineering, Monash University. Thanks also go to the administrative assistance, Ms. Sonja Aherns, the Business Manager of Ctie and Ms Roslyn Varley, Administration Officer, for their friendship during the course of my studies.

Finally, I am grateful to God for blessing me throughout my life, especially during my candidature here. Last but certainly not least, my appreciation to my children, Noor, Russul, and Hummam, and to my husband Abdullatif, for their love and understanding and constant encouragement throughout this research.

I would also like to thank both the Department of Electrical and Computer Systems Engineering, Monash University, and the Australian Telecommunications Cooperative Research Center (ATCRC) for the scholarships which made this study possible.

## Glossary

### Abbreviations

APP	A posteriori probability
AWGN	Additive White Gaussian Noise
BCH	Bose-Chaudhuri-Hocquenghem
BN	Bounded Normal
BSC	Binary Symmetric Channel
DCT	Discrete Cosine Transform
DFT	Discrete Fourier Transform
DHWT	Discrete Haar Wavelet Transform
DVD	Digital Versatile Disk
DWT	Discrete Wavelet Transform
FFT	Fast Fourier transform
GF	Galois Field
HD	Hamming Distance
HVS	Human Visual system
IA-DCT	Image Adaptive Discrete Cosine Transform
IA-W	Image Adaptive Wavelet Transform
IDWT	Inverse Discrete Wavelet Transform
IFPI	International Federation of the Phonographic Industry
IP	Intellectual property
JND	Just Noticeable Difference
JPEG	Joint Photographic Experts Group
LOT	Lapped Orthogonal Transform
MAP	Maximum A posteriori Probability
ML	Maximum Likelihood
MPEG	Moving Pictures Experts Group
MSE	Mean Square Error
pdf	Probability Density Function
PNN	Pairwise Nearest Neighbor
PSNR	Peak Signal-to-Noise Ratio
SDMI	Secure Digital Music Initiative
RS	Reed Solomon
RSC	Recursive Systematic Convolution
SOVA	Soft Output Viterbi Algorithm
SS	Spread Spectrum
VQ	Vector Quantizer
XOR	Exclusive-OR

**Symbols**

$\sigma$	Noise variance
$Z$	Set of all integers
$\eta$	Threshold value for similarity measure
$\alpha$	Watermark scaling parameter
$C(s_i)$	Set of binary elements representing the coded watermark signal
$e$	Error vector
$E_b/N_0$	Signal-to-noise ratio
$G$	Code generator matrix
$g(n)$	Unit-sample response representing the wavelet function of the wavelet transform (high-pass filter)
$G(w)$	The Fourier transform of $g(n)$
$H$	Parity check matrix
$h(n)$	Unit-sample response representing the scaling function of the wavelet transform (low-pass filter)
$H(w)$	The Fourier transform of $h(n)$
$k$	Number of input bits of the error control encoder
$L$	Wavelet decomposition level
$n$	Number of output bits of the error control encoder
$P$	The channel transition probability
$P_b$	Bit error probability
$P_c(\varepsilon)$	The probability of code word error
$Q$	Q-function
$Q\%$	JPEG compression quality factor
$R$	Rate of error control encoder
$s$	The error control code syndrom
$S$	Similarity measure (correlation coefficient between the embedded watermark and the extracted watermark)
$S(m,n)$	The original signature (watermark signal)
$S^*(m,n)$	The recovered signature at the decoder
$T(X,Y,L)$	Transfer function of the convolutional encoder
$v_j$	Set of the DWT coefficients of the host signal (host signal vector)
$\tilde{v}_j$	Set of the watermarked DWT coefficients of the host signal

# **Chapter 1**

## **Introduction**

### **1.1 Terminology**

### **1.2 Research Objectives**

### **1.3 Organization of the Thesis**

## Introduction

---

Digital watermarking, information embedding, data hiding, and Steganography have received considerable attention recently. These terms, which can for the most part be used interchangeably, refer to the process of embedding one signal, called the "embedded signal" or "digital watermark", within another signal, called the "host signal". The host signal is typically a speech, audio, image, or video signal, and the watermarking must be done in such a way that the host signal is not perceptibly degraded unacceptably. At the same time, the digital watermark must be difficult to remove without causing significant damage to the host signal and must reliably survive common signal processing manipulation such as lossy compression, additive noise, and resampling.

The motivation behind embedding information into a signal is that if the embedded information can be reliably recovered, then this information can specify the affiliation between the signal and its rightful owner; thus the information must be embedded in a manner that prevent others from destroying it easily. Though data hiding or multimedia Steganography is often confused with the relatively well

known encryption (or cryptography), the two are but loosely related. Cryptography is about hiding the contents of a message. Steganography, on the other hand, is about concealing the very fact that a message is hidden. Moreover, encryption cannot protect a signal once it has been decrypted in preparation for human use. Indeed, programs are available today that allow a consumer to capture perfect copies of an encrypted music or video files as it plays on a computer. On the other hand, watermarking does protect a file permanently, in theory at least. The watermark is added to the audio or visual data like a tattoo and cannot be removed without damaging the data. This makes it very attractive tool for copy protection, file tracking, and usage monitoring and for turning music tracks and video clips into intelligent tracks and clips with added entertainment features [Maes, et. al., 2000].

Data hiding may be considered as communication through a watermarking channel, or secret communication. Data hiding in multimedia could help in providing proof of origin and distribution of content. Multimedia content providers would be able to communicate with the compliant multimedia players through the watermarking channel provided by data hiding. This communication could control access, provide customized delivery, and provide solutions for pay-per-view implementations. This thesis explores the application of digital communication theories into multimedia data embedding.

Depending on the desired properties of the data hiding scheme, data hiding applications can be classified into two categories: watermarking for protecting intellectual property rights and data hiding for multimedia delivery.

Digital watermarking for copyright protection typically requires very few bits, on the order of 1% of the host data size. These watermarks could be alphanumeric characters, or could be multimedia data as well. The primary objective of watermarking is to be able to identify the rightful owners by authenticating the watermarks. As such, it is desirable that the methods for embedding and extracting digital watermarks be resistant to typical signal processing operations on the host, such as compression, and malicious attacks to remove the watermarks. Signal compression is of specific interest as it is perhaps the most frequently performed operation on multimedia data. In particular, lossy compression affects the internal representation of the hidden data. There is a great need for techniques that are robust to lossy compression, and development of such techniques is one of the focuses of this dissertation. However, in this research we concentrate on the so-called data hiding problem, that refers to embodiment and retrieval of hidden information in a given image or video sequence with the highest possible reliability. The creation of this hidden channel proves to be extremely important in many commercial applications where identifies for the copyright owner, recipient, transactions dates, etc, need to be hidden. We will show how channel-coding techniques known for digital communications can be effectively used for significantly improving the performance of data embedding systems. The capability to hide large amounts of data will also enable robust hiding of digital watermarks by introducing redundancies in the data.

Our main objective is to develop techniques for embedding (hiding) large amounts of multimedia data, such as text, images, audio, and video, in images and video.

Hence, the requirements are different from typical digital watermarking for data authentication. Enabling applications for such large scale data hiding include embedded control to track the use of a video clip in pay-per-view applications, hidden communications of text (e-mail), voice and visual data, to mention a few.

## 1.1 Terminology

In the following we introduce the terminology that is used in this thesis. The *signature* or *message data* refers to the secure data that we would like to embed or conceal. This is also referred to as a *digital watermark* in applications related to authentication. The source data is used to hide the signature data; we also refer to the source as the *host*. The procedure of inserting or hiding the signature data in the host is referred to as *embedding*, *hiding*, or *watermarking*. Embedding a signature into a host yields the *watermarked* or *embedded data*. The extraction procedure operates on the embedded data and possibly the original host, to yield the recovered data, also referred to as the *reconstructed data*.

## 1.2 Research Objectives

The main requirements for data embedding that are addressed in this dissertation are summarized as follows:

1. The embedded host should be perceptually indistinguishable from the original host.
2. The embedding techniques should allow for significantly larger amounts of data than that required by traditional digital watermarking problem.

3. The algorithms developed should exhibit demonstrable robustness against lossy compression.
4. Authorized retrieval will be considered both with and without the knowledge of the original host.
5. The methodology should have sufficient flexibility to achieve wide range of trade-off between the amount of data to be hidden, the level of robustness required, and the amount of perturbation to be tolerated by the host in the process of embedding.

The hosts considered here in our experimental results are images and video. Generalized algorithms are presented for hiding text or an image inside a host image, as well as image/video sequences. In general, however, the proposed algorithms may be extended to other multimedia sources.

In order to accomplish the above objectives, we have developed several new techniques for robust data hiding in images and video. The main contribution of this thesis is towards enhancing the functionality of data embedding, and to the development of robust methods for hiding. The emphasis of these techniques is to improve both the quality of the embedded data and the quantity and quality of the recovered signature data. The proposed methods are based on embedding the data in a transform domain, such as the discrete wavelet transform, and the use of error-control coding for robust recovery. The capability to hide large amounts of data enables hidden communication. Compared to the techniques in digital watermarking, which can embed data at about 1%, the proposed method can embed images up to 25% of the host data size. Since we can embed a significantly larger

number of bits for a given host, it is possible to make the embedding resistant to typical signal/image processing operations such as compression.

### 1.3 Organization of the Thesis

This thesis is organized into eight chapters. Chapter 2 discusses the general requirements on data hiding and digital watermarking. An overview of related research in watermarking schemes which includes topics in spatial watermarking, spread spectrum watermarking, spectral watermarking, and a brief survey of currently available commercial and public domain software are provided.

Chapter 3 describes a general approach to embedding signature data using a discrete wavelet transform. The compressed indices of the signature are distributed in the subbands of the host. The proposed scheme enables signature images to be as much as 25% of the host image data, and hence could be used both in digital watermarking as well as in image/data embedding.

In Chapter 4, we propose a robust data embedding scheme, which use error-correcting channel codes such as BCH and Reed-Solomon codes. A trade-off between the quantity of the hidden data and the quality of the watermarked image is achieved by varying the number of quantization levels for the signature and a scale factor for data embedding. Experimental results show that the watermarked image is transparent to embedding large amounts of hidden data, and the quality of the extracted signature is high even when the image is subjected to high compression ratio of JPEG lossy compression and noise addition. Moreover, lossless recovery was obtained for embedding text messages even at high

compression ratio. This research has been published in [Abdulaziz and Pang, 2000a], [Abdulaziz and Pang, 2000b], and [Abdulaziz and Pang, 2000c].

Chapter 5 presents a technique that hides data using convolutional codes with different rates and constraint length. In addition, concatenated codes designed from block and convolutional codes have been implemented. Further, turbo codes derived from parallel concatenation of convolutional codes has also been employed. The decoding is utilized using Viterbi decoding and maximum a posteriori probability techniques. Some of This work has been published in [Abdulaziz and Pang, 2001].

In Chapter 6, a new technique for embedding image data that can be recovered in the absence of the original host image is discussed.

Chapter 7 demonstrates signature recovery using the methods described in the previous chapters but for hiding images in video sequence. The watermark data is extracted from video sequence that is subjected to MPEG compression. Chapter 8 concludes with some discussions and future research directions.

## **C h a p t e r 2**

### **A Review of Data Embedding (Digital Watermarking)**

#### **2.1 Watermarking**

#### **2.2 Application Areas**

#### **2.3 General Requirements**

#### **2.4 Related Work on Watermarking Schemes**

#### **2.5 Perceptual Watermarking Based on Image-Adaptability**

#### **2.6 Marking Text Documents**

#### **2.7 Commercial Software**

#### **2.8 Summary**

## A Review of Data Embedding (Digital Watermarking)

---

Motivated by the overwhelming need for Internet data security, digital watermarking has emerged, as an important area of research in multimedia data processing. *Digital watermarking* is a technology being developed to ensure security and protection of multimedia data. The purpose of digital watermarking is not to restrict use of multimedia resources, but rather to facilitate data authentication and copyright protection. On the other hand, *data hiding* can be considered as a generalization of watermarking wherein large amounts of data are embedded into a host medium. In this chapter we provide an overview of the main issues in data hiding and watermarking and give detailed review of related work.

### 2.1 Watermarking

The use of digitally formatted audio, video, and printed information is increasing rapidly with the expansion of multimedia broadcasting, networked databases, electronic publishing, etc. This progressive switch to digital representation of multimedia information (text, video, and audio) provides many advantages, such as

easy and inexpensive duplication and distribution of the product. However, it also increases the potential for misuse and theft of such information, and significantly increases the problems associated with enforcing copyrights on multimedia information [Wolfgang and Delp, 1997a], [Cox and Miller, 1997], [Swanson et. al., 1998], [Anderson and Petitcolas, 1998], [Wu and Liu, 1998]. The protection and enforcement of intellectual property rights has become an important issue in the digital world. Well-established organizations are actively pursuing research into digital watermarking and are calling for proposals to incorporate these methods in current multimedia standards. There are various watermarking schemes applied to images [Mintzer et. al., 1997] and several methods applied to audio and video streams [Qiao and Nahrstedt, 1998], [Craver et. al., 1998]. Among them, a large class of the watermarking schemes addresses invisible watermarks.

We are currently in an evaluation phase of the technology in which researcher are developing general guidelines for effective watermarking algorithm design, improving reliability within the constraints of computational complexity and tailoring to the constantly changing needs of multimedia industries. Although there are commercially available digital watermarking systems [Zhao, et. al., 1998] the area is still at its active research phase. Traditionally, it takes a cryptographic system ten to twenty years to be adopted for general use. What makes watermarking a formidable problem is the urgency to incorporate it within the divers objectives of multimedia applications.

## 2.2 Application Areas

Digital watermarking aids owners in asserting their intellectual property rights (IPR) on the works of art they create. These rights are particularly difficult to enforce with digital images, since it is easy to copy and distribute perfect copies of an original image. Figure 2.1 shows that the basic components of any watermarking technique consist of a marking algorithm that inserts information, the watermark, into an image. The watermark is inserted into the image in the spatial domain or spatial frequency domain. As part of the watermarking technique, a testing algorithm must be defined that tests an image to see if a particular watermark is contained in the image. It is also desirable for the testing procedure to determine if the image has been altered and to supply localization information as to where the image was altered. Also, to assert ownership that is consistent with current intellectual property right law, the watermarking technique must support the use of third-party cryptographic-based digital time stamping that is embedded in the image through the watermarking process.

Some authors, for example in [Bender et. al., 1996], refer to watermarking technique only when the application embeds a few bits (as few as one bit) of data for copyright protection applications. Other applications are considered to fall into the category of data embedding. In this thesis we will be using both terms as our algorithm is a general algorithm and can be used both in digital watermarking and data embedding.

Depending on the desired properties of the data hiding scheme, we can classify data hiding applications into the following three categories:

- Watermarking for protecting IPR
- Watermarking for tamper detection
- Data hiding for multimedia delivery
  - Captioning
  - Customized media delivery
  - E-commerce
  - Access control
  - Access monitoring
  - Intelligent agents (executable codes for interactive communication)

It is also true that these applications vary greatly depending on the application, for instance, in the use of watermarking for protection of intellectual property, the watermark is used to supply digital objects on the Internet with an identification of origin. On the other hand, *Fingerprinting* attempts to identify individual copies of an object by means of embedding a unique marker in every copy that is distributed. If later an illegal copy is found, the copyright owner can identify the buyer by decoding the hidden information (*traitor tracing*) [Cox and Linnartz, 1998], [Kundur and Hatzinakos, 1999]. A number of applications and their properties are given in [Cox, et. al., 2000], [Barnett, 1999], and [Piva, et. al., 1999b].

Further, the popularity of the technology for multimedia applications is reflected in the calls for watermarking algorithms proposals for such mainstream digital media standards as digital versatile disk (DVD), MP3, MPEG\_4, JPEG2000, and the International Federation of the Phonographic Industry (IFPI). In fact, the Galaxy Group, comprised of Hitachi Ltd., IBM Corp., NEC Corp., Pioneer Electronic

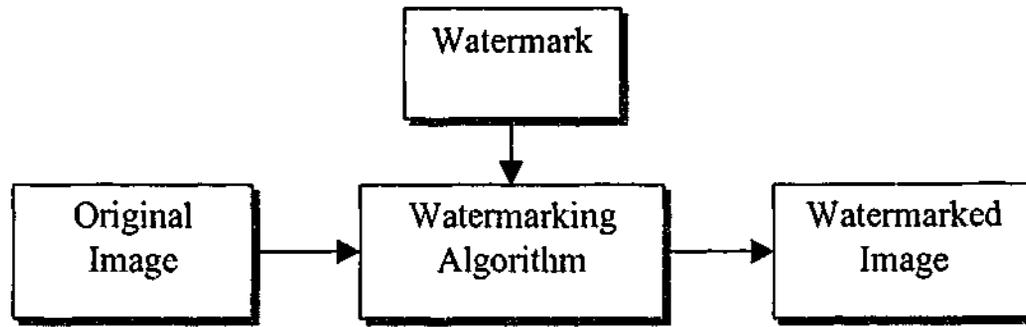


Figure 2.1 Block diagram of a watermarking algorithm.

Corp. and Sony Corp., has recently agreed upon a video standard for DVD copy protection [Yoshida, 1999]. Similarly, at the time of writing this thesis, reports in the press indicate that the selection made by Secure Digital Music Initiative (SDMI) in its phase I call for proposal, still suffer from audibility problems and may be easily removed by simple malicious attacks [Tewfik 2000], [Petitcolas, 2001]. However, recent studies of watermarking as a noisy communication channel may lead to a new wave of advances.

### 2.3 General Requirements

Each watermarking application has its own specific requirements. Therefore there is no universal set of requirements as such that must be met by all watermarking techniques. Nevertheless, some general directories can be given for most of the applications. These include that it be difficult to notice, robust to common distortions of the signal, resistance to malicious attempts to remove the watermark,

allow multiple watermarks to be added and that the decoder be scalable [Piva et al., 1998b]. Although only some factors are appropriate for a given application, we present all the most popular metrics below to highlight the character of a good watermarking scheme. Without loss of generality, we assume the host and watermarked signals are images.

### **1. Imperceptibility**

The watermark must not be obtrusive in the host signal. Specifically, a user should not be able to distinguish the watermarked signal from the host signal.

### **2. Robustness**

The watermark should resist both intentional and non-intentional tampering. Examples of non-intentional tampering are some common signal processing operations like lossy compression, histogram equalization, edge enhancement, low-pass filtering, gamma correction, scaling, rotation, D/A and A/D conversions, color adjustment etc. Intentional tampering is done with the sole purpose of removing the watermark while simultaneously trying to protect the quality of the image.

### **3. Security**

The watermark should be non-removable even if the embedding algorithm is known.

### **4. Computational Complexity**

Depending on the particular application and media being watermarked, computational complexity can be a significant factor in the assessment of the feasibility of a watermarking algorithm. For example, in DVD players watermark extraction must be performed real-time, but in image watermarking for high

security intellectual property applications, it may not be of as much concern. This thesis is concerned, in part, with assessing the potential performance of proposed watermarking strategies, so complexity is a secondary issue in the algorithm development process.

### **5. Modification and Multiple Watermarks**

In some circumstances, it is desirable to alter the watermark after insertion. For example, in the case of digital videodiscs, a disc may be watermarked to allow only a single copy. Once this copy has been made, it is then necessary to alter the watermark on the original disc to prohibit further copies. Changing a watermark can be accomplished by either (i) removing the first watermark and then adding a new one or (ii) inserting a second watermark such that both are readable, but one overrides the other. The first alternative does not allow a watermark to be tamper resistant since it implies that a watermark is easily removable. Allowing multiple watermarks to co-exist is preferable and also facilitates the tracking of content from manufacturing to distribution to eventual sales, since each point in the distribution chain can insert their own unique watermark.

### **6. Scalability**

In commercial applications, the computational costs of the encoder and the decoder are important. In some applications, the insertion is only done once and can be performed off-line. Consequently, the cost of encoding may be less important than the cost of decoding, which may have to occur at real-time video rates, for example, computational requirements constrain a watermark to be simple, but this simplicity may significantly reduce the resistance to tampering. Further, it is well

known that computer speeds are doubling approximately every eighteen months, so that what looks computationally unreasonable today may very quickly become a reality. It is therefore very desirable to design a watermark whose decoder is scalable with each generation of computers. Thus, for example, the first generation of decoder might be computationally inexpensive but might not be as reliable as next generation decoders that can afford to expand more computation to deal with issues such as geometric distortions. More specific performance measures will be discussed for each of the embedding techniques in Chapters 3-7.

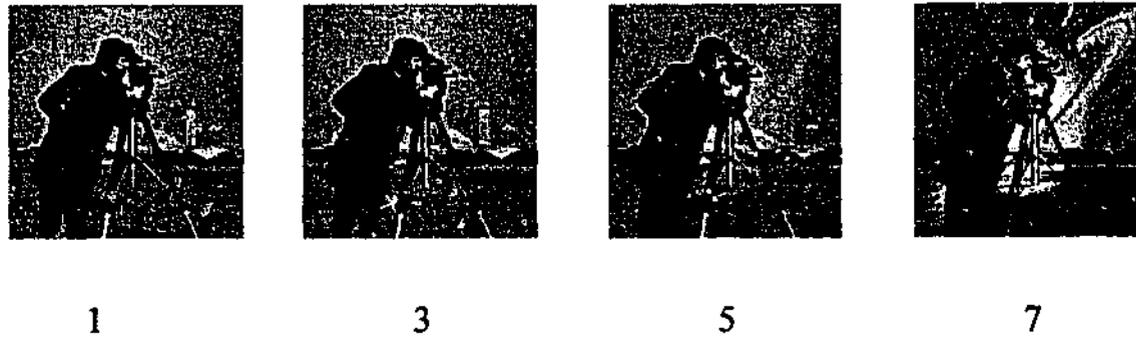
## 2.4 Related Work on Watermarking Schemes

During the past few years, a number of digital watermarking methods have been proposed. The two basic modalities for image watermark encoding are: spatial-domain techniques (spatial watermarks) and spatial frequency-domain techniques (spectral watermarks). The following section describes several spatial watermarking algorithms that rely on some type of perceptual knowledge in the encoder. Many of the spatial watermarking techniques provide simple and effective schemes for embedding an invisible watermark into the original image but are not robust to common image alterations. Another way to mark an image is to transform it into the frequency domain- Fourier, DCT, wavelet, etc. – before marking it. The mark is incorporated directly into the transform coefficients of an image. The inverse-transform coefficients form the marked image. These types of algorithms are often called spectral watermarks, and commonly use frequency sensitivity of the human visual system to ensure that the watermark is invisible. Many of these techniques are *private watermarks*, which require the original image to verify the

mark. Algorithms that do not require the original image for testing are called *public watermarks* [Petitcolas, et. al., 1999].

#### 2.4.1 Spatial Watermarking: Simple Systems

Among the earliest works in image watermarking, [Tirkel et. al 1993], [Schnydel et. al., 1994], proposed a digital watermarking method which substitutes the least significant bits of randomly selected pixels with bits from M-sequence generator, to represent the watermark. The original 8 bit gray scale image data is compressed to 7 bits by adaptive histogram manipulation, and then the LSB from the watermark sequence is combined to form the encoded image. The watermark can be decoded by comparing the LSB bit pattern with a stored counterpart, Figure 2.2 shows the image of *Camera-man* as the host image and *Saturn* image as the watermark for different values of the number of bits replaced by the watermark. However, because of the use of the least significant bits, the identification code can be easily destroyed. For example, almost any trivial filtering process will change the value of many of the least significant bits. One possible countermeasure is to use redundancy: either apply an error correcting code, or simply embed the mark a large number of times. For example, the "Patchwork" algorithm of [Bender et. al., 1995], where he hides a bit of data in an image by increasing the variance in luminosity of a large number of pseudo-randomly chosen pixel pairs. In another way, Patchwork is a statistical approach. It randomly chooses a pair of image points  $(a_i, b_i)$ , then increases the brightness of  $a_i$  by one and decreases the brightness of  $b_i$  by one. These two steps repeated  $n$  times. The expected value of the sum of



**Figure 2.2** Watermarking using the LSB replacement for number of bits replaced showing under each image.

the difference of such  $n$  pairs is then  $2n$ . This method is resistant to compression, filtering, and cropping. However, its assumption that all brightness levels are equally likely is not practical. A second method presented in the same paper is called texture block coding. A watermark consisting of a textured patch is embedded into an area of the image with a similar texture. This is a good example of using some common sense rules to determine where signal alterations will be least noticeable, which results in good visual quality for the marked image. The auto-correlation of the image results in the recovery of the shape of the region. This method is limited to those images containing relatively large areas of texture; the technique is also vulnerable to low-pass filtering. A similar system was proposed by [Pitas, 1996]. Much the same techniques can be used to mark digital audio as well.

One way to attack such systems is to break up the synchronization needed to locate the samples in which the mark is hidden, for example, one can crop the image.

### 2.4.2 Basic Spread Spectrum Approach

Most of the work on robust watermarking is based on spread spectrum (SS) principles [Dixon, 1984]. In SS watermarking, the embedded signal is generally a low energy pseudo-randomly generated white noise sequence. It is detected by correlating the known watermark sequence with either the extracted watermark or the watermarked signal itself (if the host is not available for extraction). If the correlation is above a given threshold then the watermark is detected. The anti-jamming properties of SS signaling makes it attractive for application in watermarking since a low energy, and hence imperceptible, watermark, robust to narrow band interference, can be embedded [Dixon, 1984]. In most SS techniques the pseudo-random white noise watermark sequence is added to the host signal and is detected by correlating the known watermark with the watermarked signal. That is, the watermark is embedded in some domain of the signal using linear addition. As an example, if the watermark is added in the spatial domain, then the watermarked image is given by

$$J = I + W \quad (2.1)$$

where,  $I$  and  $J$  are the host and watermarked signals, respectively, and  $W$  is the pseudo-randomly generated watermark.

The watermark in [Smith and Comiskey, 1996] is based on SS communications. The authors develop a theory of information hiding in images that sets out to quantify channel capacity and jamming margin. They describe test implementations of data hiding schemes inspired by both direct sequence and frequency hopping spread spectrum concepts. Also, they discuss the characteristics of the data hiding

schemes, such as the amount of information that can be hidden, the perceptibility, the robustness which is modeled using the quantities of channel capacity, the signal-to-noise ratio, and the jamming margin. They introduce new data hiding schemes whose parameters can be adjusted with the capacity, imperceptibility, and robustness. So robustness may be increased either by increasing signal-noise ratio<sup>1</sup> (at the cost of perceptibility), or by decreasing the size of the embedded data (the capacity), which increases the processing gain. They found that the frequency hopping spread spectrum techniques is superior perceptually and has better resistance to accidental removal by compression techniques, while the direct-sequence technique is more robust against deliberate removal attempts.

The shortcoming of such method is in the channel capacity estimate, where they used the capacity formula for a Gaussian channel, which is not the best model of the noise in a single image. It assumes that the Gaussian channel has the same power at each frequency. But the host images do not have a flat frequency characteristics, especially after compression.

### 2.4.3 Other Spatial Domain Watermarks

The watermark proposed in [Wolfgang and Delp, 1997a], and [Wolfgang and Delp 1997b]) is known as the Constant and Variable two-dimensional Watermark respectively. The authors reshape an m-sequence into two-dimensional watermark blocks, which are added and detected on a block-by-block basis.

---

<sup>1</sup> Here the signal-noise-ratio is the ratio of the watermark signal level compared to the host signal level.

Both schemes do not require the original for watermark detection. However, it can detect local alterations in a received image on a block-wise basis.

#### **2.4.4 Fractal Operations**

Darven and Scott [Darvin and Scott, 1996] presented an approach to image Steganography utilizing fractal image compression operations. An information bit is embedded into the Stego-image by transforming one similar block into an approximation for an another. The data are decoded using a visual key that specifies the position of the range and domain regions containing the message. Unfortunately, the amount of data that can be hidden using this method is small and susceptible to bit errors. Moreover, the search for similar blocks in the encoder, and the decoder comparison process, are both computationally expensive operations.

Similar approach was adopted by Bas et al [Bas et. al., 1998]. The watermarking scheme uses similarities to embed a mark. Two different algorithms were developed: the first adds the similarities in the spatial domain, the second in the DCT domain. Their results indicate that the DCT based scheme is more robust to compression than the spatial one.

#### **2.4.5 Pattern Overlaying**

By interpreting watermarking as a key-dependent pattern overlaying, a new watermarking scheme was proposed by Fridrich [Fridrich, 1997]. The method is based on overlaying a pattern with its power concentrated mostly in low frequencies in order to guarantee robustness. The method is described as follow:

The watermark is generated by choosing a string of bits (author's ID + image hash) which can be transformed into a smooth, almost transparent pattern to be overlaid

over the carrier image. The pattern should not exhibit traces of any regular building blocks. Also, patterns generated by two different watermarks should be uncorrelated. To achieve these goals, the author proposed to seed a random number generator with the watermark to create an initial black and white two-dimensional random pattern. The pattern will then further processed to eliminate high frequencies, which is done by applying low pass filters to the initial pattern. The initial pattern was initialized with 0's and 1's with the same probability 50%. The random black and white pattern was then processed by cellular automation with the voting rule. Since the overlaid has most of its power concentrated in low frequencies, excellent robustness properties is expected similar to the method proposed by [Cox et. al., 1996], and since the overlaid pattern depends on the key in a complicated way, even if the watermark shows in regions of nearly constant luminosity, it does not reveal any information about the key if a cryptographically strong number generator was used. Another advantage of this method is that it avoids transformations, which results in a faster and easy implementation.

To extract the watermark, the watermarked image is subtracted from the original and the correlation between the difference and the smoothed pattern is calculated. Based on the value of the correlation, decision is made about the presence of the watermark. The correlation was performed in the Fourier space rather than in the spatial domain.

The robustness of the method with respect to filtering, JPEG compression, cropping, noise adding, and collusion was studied using computer experiments. The results show that the watermark was resistant to all the above attacks. Also the

watermarking method require the original, unwatermarked image in order to recover the watermark.

#### **2.4.6 Spectral Watermarking: A Spread Spectrum Watermark**

##### **Embedded in the DCT Domain**

Perhaps Cox et. al. [Cox et. al., 1996] is the first work utilizing DCT decomposition for data embedding. They argue that a watermark must be placed in perceptually significant components of a signal if it is to be robust to common signal distortions and attacks. To avoid the perceptual degradation because of watermarking those components, they propose to insert a watermark into the spectral components of the data using techniques analogous to spread spectrum communications, hiding a narrow-band signal in a wide-band. The watermark consists of 1000 randomly generated numbers. The length of the watermark is variable and can be adjusted to suit the characteristics of the data. The published results show that the technique is very effective both in terms of transparency, robustness to signal processing, and attempt to remove the watermark. The types of image distortions to which the technique is robust include cropping, very low data rate JPEG compression, Printing and scanning, as well as collusion with several independently watermarked images. One of the significant contributions in this work is the realization that the watermark should be inserted in the perceptually significant portion of the image in order to be robust.

A Gaussian sequence is used as the signature. Detection of the signature is accomplished by correlating the Gaussian sequence with the 1000 (modified) DCT

coefficients after subtraction of the corresponding DCT coefficients of the host image.

This algorithm is one of the earliest attempts at providing some image adaptability in the watermark embedding scheme. This is due to the fact the watermark strength depends on the intensity value of the DCT coefficients of the original image. In this way, the watermark signal can be quite strong in the DCT values with large intensity values, and is attenuated in the areas with small DCT values. This provides a watermark signal that is quite robust and for most images, transparent. However, because the DCT transform in this scheme is based on the whole image rather than the usual block-based approach commonly found in image and video compression schemes, the transform does not allow for any local spatial control of the watermark insertion process. In other words, the addition of a watermark value to one DCT coefficient affects the entire image: there is no mechanism for local spatial control in this particular framework. The limitation of this scheme is its dependence on the original image for detection of the watermark, which makes it susceptible to multiple claims of ownership [Wolfgang and Delp, 1997a], [Craver et al., 1998], and [Zeng and Liu, 1997].

#### **2.4.7 A Linear Combination of Marked and Unmarked Images**

The method by Piva, Barni, Bartolini, and Cappellini [Piva et al., 1997], [Piva et al., 1998a], [Barni et al., 1998a], and [Barni et al., 1998b] is similar to Cox's method [Cox et al., 1996]. The DCT of the entire image  $I$  is computed, and the coefficients are ordered in the zigzag fashion of JPEG to form the coefficient vector  $I_D$ . To decrease the chance of the watermark being perceptible, the first  $L$

coefficients are not marked.  $W$  is of length  $M$  pseudo-random sequence of numbers, which is added to DCT coefficients. Testing is similar to Cox's method, but the detection does not require the original image.

#### 2.4.8 Sub-band Watermarking

The image watermarking technique in [Swanson et. al., 1996a], and [Swanson et. al., 1996b] first computes the DCT of an original image  $X$  on an  $8 \times 8$  block-wise basis. Making thresholds,  $m$ , are defined and computed for each block based on the DCT coefficients; these thresholds are similar in theory to the JND values used in IA-DCT. The watermark for an individual block, is a reshaped  $m$ -sequence and different watermarks are used for each block. To ensure that the addition of the watermark is imperceptible, spatial domain correction is employed on the marked blocks. The watermark verification is similar to the IA-W method and Cox's testing. A hypothesis test is performed on the normalized cross-correlation coefficient computed between the extracted watermark, and original watermark. If the result is above a certain threshold, the image is authentic. As in previous techniques, the threshold is determined according to the desired probability of detection and probability of false alarm.

A wavelet-based algorithm that also scales the watermark on a block-by-block basis is presented in [Kundur and Hatzinkos, 1997], [Kunder and Hatzinkos, 1998]. The watermark ( $W$ ) for this technique is much smaller than the original image, copies of  $W$  are tiled throughout the subbands of a wavelet decomposition of the host image. This protects against cropping, but may be susceptible to collusion attacks. The original image is needed to extract the watermark in [Kundur and

Hatzinkos, 1997], while in [Kundur and Hatzinkos, 1998], the authors describe a detection without the need for original image. Also, in [Kundur and Hatzinkos, 1998], analysis to estimate the probability of a false positive and the probability of a false negative was provided. The expressions suggest that increasing the length of the watermark can reduce the probability of detection error. Similar technique of fusing host image and watermark image in the wavelet domain was presented in [Chae and Manjunath, 1998].

#### 2.4.9 Other Transform-Based Approaches

Another global method also modulates DCT coefficients, but uses a one-dimensional bipolar binary sequence for the watermark [O'Ruanaidh et. al., 1995] and [O'Ruanaidh et. al., 1996a]. The DCT of the original image is first obtained. The marking procedure consists of sorting the DCT coefficients according to their absolute magnitude. The owner then defines a percentage of total energy,  $P$ , and identifies the largest  $n$  coefficients that make up  $P$  percent of the total energy. The watermark sequence is then added to all the AC coefficients

A larger  $P$  increases the number of elements of  $W$  that can be embedded in  $I$ , but increases the chance that  $W$  will be perceptible.  $W$  and the list of selected coefficients must be kept secret. The verification procedure first extracts the watermark from the marked coefficients in the received image and a procedure similar to Cox [Cox et. al., 1996] can then be used to verify the watermark, but both this method and Cox's method require the original host image to extract the watermark.

An early DCT-based technique was presented in [Koch and Zhao, 1995] and used in the product SysCop, is similar to direct sequence and frequency hopping spread spectrum communications. The proposed approach, called Randomly Sequenced Pulse Position Modulated Code (RSPPMC) copyright labeling, is rooted in the fact that typical digital images of people, buildings and natural settings can be considered as non-stationary statistical processes, which are highly redundant and tolerant of noise. Hence, changes in the image data caused by moderate levels of wide-band noise or controlled loss of information are hardly visibly noticeable.

The RSPPMC method consists of two components. The first component produces the actual copyright code and a random sequence of locations for embedding the code in the image. The second component embeds the code at the specified locations, using a simple pulsing method.

This seems to be a reasonable approach for adding some sort of perceptual criterion. As watermarks inserted into the high frequencies are most vulnerable to attack whereas the low frequency components are perceptually significant and very sensitive to alterations; such alterations may make the watermark visible.

The image is segmented into  $8 \times 8$  non-overlapping blocks; each block is transformed into the frequency domain using the DCT. This is the same building block that is used in JPEG image compression. A pseudo-random subset of the blocks is chosen (to minimize detection) and a triplet of midrange frequencies (which is equivalent to eight coefficients in the block) is slightly altered to encode a binary sequence. Cropping of images may lead to difficulties in extracting messages that was pseudo-randomly embedded.

Langelaar et al. report that image degradation is visible in their implementation studies to assess Koch and Zhao's method, and results are presented in [Langelaar et al., 1996]. In [Langelaar et al., 1997], they propose two image-watermarking methods. The first one extends the existing spatial labeling technique, which adds a positive integer constant  $k$  to the brightness of 50% of the pixels in an image. This constant  $k$  is called the label embedding level. By dividing the image into blocks and searching an optimal label-embedding level  $k$  for each block instead of using a fixed embedding level, a large and more robust label can be embedded in an image. The second method removes high frequency DCT-coefficients in some areas to embed a label. However, this method may remove too many DCT-coefficients therefore cause distortions. A review of the above-mentioned methods is presented in [Langelaar et al., 2000].

The basic idea introduced in [Koch and Zhao, 1995] are further extended in [Bors and Pitas, 1996] by introducing the watermark encoding in the actual quantization process of the mid-frequency coefficients. The result is two schemes that do not require the original image for watermark decoding. The first scheme embeds a linear constraint among selected DCT coefficients; the second defines a circular detection region in the DCT domain similar in concept to vector quantization (VQ) [Gersho and Gray, 1992]. A different improvement to [Koch and Zhao, 1995] is presented in [Tao and Dickinson, 1997], which classifies blocks of the image according to their energy content; the amount of energy in the block determines in part the amplitude of the mark to be embedded in that block. The original image is segmented into  $8 \times 8$  blocks. Image blocks that contain either sharp edges, or have

little texture are not marked. In these blocks the watermark would be more easily perceived.

A variation on Koch & Zhao's method for image authentication is presented by Schnieder and Chang [Schnieder and Chang, 1996]. The technique alters transform coefficients to enforce a relationship between the coefficients.

Another method [O'Ruanaidh et. al., 1996b] embeds the watermark in the phase information of the discrete Fourier Transform (DFT) of an image. In [O'Ruanaidh and Pun, 1998], the author presented a new method for embedding information in an invariant domain by combining a Fourier transform with a log polar map. The approach used in this paper withstands rotation and scaling by being invariant to these transformations. This watermarking scheme is difficult to implement in practice. The first difficulty is that both the log-polar mapping and the inverse log-polar mapping can cause a loss of image quality. The second difficulty is numerical, where the computation of the Fourier-Mellin transform somewhat problematic. In general, this method embeds watermarks, which resist rotation and scale transformations, however, with some loss in the robustness against JPEG compression. Also, the original image is needed for watermark extraction.

A wavelet-based version of Cox's method [Cox et. al., 1996] is described in [Xia et. al., 1997].

#### **2.4.10 Watermarking using the Lapped Orthogonal Transform (LOT)**

Periera et. al. [Perirra et. al., 1999] proposed a new approach based on Lapped Orthogonal Transforms (LOT) in which the watermark is inserted adaptively into the LOT domain. The motivation for using the LOT as the basis for embedding a

watermark is that the DCT may produce blocking artifacts if the strength of the watermark is increased sufficiently. The drawback of the LOT is that it is not robust in itself to cropping, rotation or scaling. Consequently, the authors suggested adding to the LOT domain watermark a template in the discrete Fourier transform (DFT) domain. The template is used to increase robustness of the watermark. The original image is not needed for detecting the watermark.

## 2.5 Perceptual Watermarking Based on Image-Adaptability

The use of either formal visual models or common sense rules based on some knowledge of human visual system would be beneficial in developing watermark encoders that provide transparent quality. In theory, a good visual model should provide the maximum strength, maximum length watermark sequence that can be inserted without introducing visual distortions. The techniques described below use formal visual models.

The two techniques described here, the image-adaptive DCT (IA-DCT) approach [Podilchuck and Zeng, 1997a] as well the image-adaptive wavelet (IA-W) approach [Podilchuck and Zeng, 1998]) have been motivated by the results presented in the spread spectrum technique of Cox [Cox et. al., 1996]. The authors introduced the use of formal visual models, into two watermarking frameworks.

The frequency decomposition for the image-adaptive DCT algorithm is based on an 8 x 8 DCT framework. Unlike the decomposition in the spread spectrum approach [Cox et. al., 1996], the block-based approach provides local control that allows for incorporating local visual masking effects. The local information is stored in what

is called a just-noticeable difference matrix (JND). The values in the JND matrix are based on the frequency domain representation of the image; they are the thresholds beyond which any changes to the respective coefficient will most likely be visible [Watson, 1992]. In the applications addressed here, the original image is available at the decoder and the JND threshold values can be obtained directly from this image. Actually, the JND thresholds can be estimated from the received watermarked image fairly accurately; this means that this technique can be applied to applications where the original image is not available for watermark detection.

The JND thresholds derived from the visual model consist of an image independent part based on frequency sensitivity, and an image dependent part based on luminance sensitivity and contrast masking. These three components of the visual model have been derived in the context of image compression to determine the maximum amount of quantization noise that can be tolerated at every image location without affecting the visual quality of the image [Watson, 1992]. In the context of image watermarking, the JND thresholds can be used to determine the maximum amount of watermark signal that can be tolerated at every image location without affecting the visual quality of the image.  $W$  consists of a sequence of real numbers generated from a Gaussian distribution with zero mean and unit variance as proposed in the spread spectrum technique of [Cox et. al., 1996].

For the IA-W scheme [Podilchuck and Zeng, 1998], frequency sensitivity thresholds are determined for a hierarchical decomposition using the 9-7 biorthogonal filters in [Antonini et. al., 1992]. Due to the hierarchical decomposition, this approach has the advantage of consisting of watermark

components that have varying spatial support. This provides the benefits of both a spatially local watermark and a spatially global watermark. The watermark component with the local spatial support is suited for local visual masking effects and is robust to signal processing such as cropping. The watermark component with global spatial support is robust to operations such as lowpass filtering. Due to the hierarchical nature of such approach, this scheme is more robust to certain types of distortions than the DCT-based framework.

Watermark detection for the spread spectrum approach as well as the IA-DCT and IA-W schemes is based on classical detection theory. The received image subtracted from the original image and the correlation between the signal difference and a specific watermark sequence is determined. The correlation value is compared to a threshold to determine whether the received image contains the watermark in question. Comparing the correlation coefficient to a threshold value performs the watermark detection. This threshold can be modified according to the tradeoff between the desired probability of detection, and the probability of false identification (false alarm).

Similarly, the watermark decoder for the wavelet scheme is also based on a correlation receiver. What is different in the IA-W decoder is that the correlation is computed separately for each resolution level as well as each frequency bin. Evaluating the correlations separately at each resolution can be used to our advantage in the detection process. For instance, cropping the image will impact the watermark values in the lower layers more than in the higher layers. This is due

to the fact that bands in higher layers (and the corresponding watermark sequence) correspond to a smaller spatial support.

A watermarking technique that is based on utilizing human visual systems (HVS) characteristics is presented in [Kim et. al., 1998]. In this scheme, Watson [Watson, 1992] visual model was used to determine image dependent upper bound values on watermark insertion. The Watson model is based on the same image independent component utilizing frequency sensitivity as determined by measurements of specific viewing conditions.

For watermark generation, the authors used bounded-normal (BN) distribution, which do not yield the value outside  $[-0.1, 1.0]$ . The reason was, watermarks which is generated from a normal distribution  $N(0,1)$ , sometimes results in values that exceeds the JND which in turn makes image impairment.

Watermark detection is performed by subtracting the original image from the received one, and the correlation between the signal difference and a specific watermark sequence was determined. The correlation value is compared to a threshold to determine whether the received image contains the watermark or not.

Results were slightly higher than Podilchuk's scheme [Podilchuck and Zeng, 1997a].

A revision to IA-DCT as applied to JPEG images [Podilchuk and Zeng, 1997b] is proposed in [Zeng and Liu, 1997], and avoids the use of the original unmarked image in the verification procedure. In this technique, it is assumed that the original image has already been JPEG compressed. The marking procedure is similar to IA-DCT, except a different subset of DCT coefficients are marked.

## 2.6 Marking Text Documents

The applications and problems associated with text marking are unique. A text document consists of objects of different sizes, such as paragraphs, lines, words, characters, figures, and captions. The basic idea is to encode information by moving these objects by small amounts. For instance, a text line can be moved up to encode a "1" or down to encode a "0". The movements may be as small as a pixel, or  $1/300^{\text{th}}$  inch at 300 dot-per-inch (dpi) resolution. The motivation for encoding data in this manner is that moving an entire object is less perceptible than distorting the object. Encoding techniques that distort the object include dithering and modifying the transform components [Fu and Au, 2000]. [Maxemchuck and Low, 1997] described several invisible techniques for encoding information in text documents, the encoding they used for text marking was moving a paragraph vertically (or horizontally), move a text line vertically, move a block of words or a single word horizontally, or move a character horizontally. The movement can be nested or combined to encode more information. For decoding the information that has been placed in text, several methods were implemented. Some of these techniques only require general knowledge about the structure of the document while others also use the characteristics of the specific document. In general, a decoder can more accurately extract a signal in the presence of noise when it has more information about the signal.

## 2.7 Commercial Software

Recently, there have been many commercial software packages for copyright authentication, (e.g. [PictureMarc] and [SureSign]), some of which could be used for multimedia data hiding. Johnson and Jajodia [Johnson and Jajodia, 1998], [Duric, et. al, 1999] provide a comparative evaluation of several different commercial software. Most of these methods employ variations of least-significant bit encoding for data embedding.

Another shareware program is StegoDos [Stego Dos]. This program uses the least significant bit method to hide messages.

Technical details of these commercially available software utilities are generally not available to the public. Furthermore, the embedding algorithms are generally not very robust. Even if the software is capable of hiding a large quantity of data, the embedded data can be easily removed with simple signal processing methods.

## 2.8 Summary

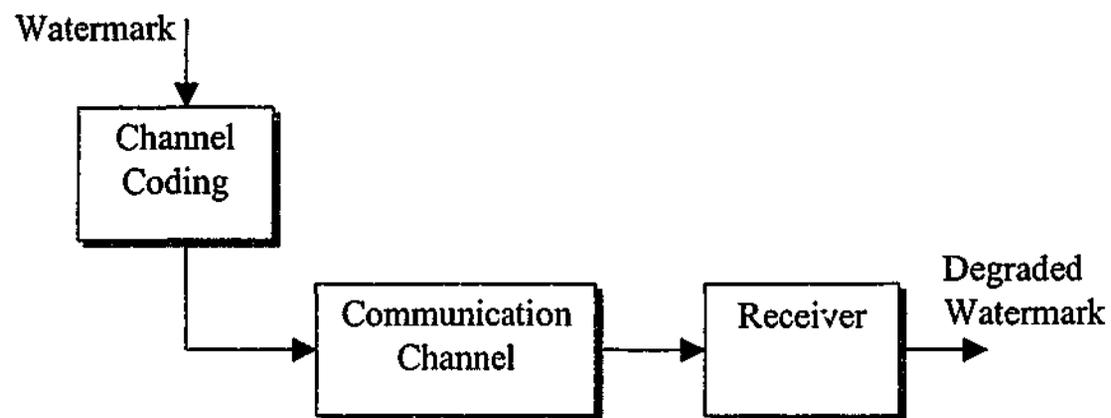
Digital watermarking and data hiding has a very active research area. A majority of the previous work is related to digital watermarking for copyright authentication. Methods for embedding data both in the spatial and in the frequency domain have been explored. However, most of these existing algorithms do not support large amounts of data hiding.

In this thesis, we attempt to develop data embedding algorithms, which are portable to a variety of applications such as those, presented in Section 2.2. We therefore,

require that the watermark be binary and be extractable not just detectable. For such design objectives SS approaches have the following limitations:

- Spread spectrum allows detection of a known watermark, but the fundamentally large bandwidth requirement does not facilitate the extraction of a long bit sequence or logo from an audio signal or image.
- Spread spectrum approaches are vulnerable to inter-symbol interference caused by multipath fading [Flikkema, 1997]. For watermarking this implies that if the energy of the watermark is reduced due to fading-like distortions on the watermark, any residual correlation between the host signal and watermark as discussed above can result in unreliable detection [Chen and Wornell, 1999].
- The correlator receiver structures used for watermark detection are not effective in the presence of fading. Although SS systems in general try to exploit spreading to average the fading, the techniques are not designed to maximize performance. SS is commonly used in wireless communications for its interference rejection capabilities of narrow band noise. It has no advantage in environments in which fading is prevalent. For such applications, path and antenna diversity are commonly used to overcome fading [Simon, et. al., 1989].

In this thesis, we consider a communication model to watermarking: communicating the watermark is analogous to transmission of the signal through an associated watermark channel as shown in Figure 2.3. We understand that common multimedia signal distortions including cropping, filtering, and perceptual coding are not accurately modeled as narrow band interference which is a common assumption in SS approaches. Instead, we believe that such signal modifications



**Figure 2.3** Watermarking as a Communication Problem. The watermark embedding and extraction process may be interpreted as communication transmission within a noisy fading channel.

have the effect of fading on the watermark. As a result, the watermark can be made more robust by employing effective channel coding techniques.

In the following chapters, we introduce several new techniques, which enable large quantities of data hiding in images and video and which implement channel codes such as block, convolutional and concatenated codes. The proposed algorithms are robust to image/video compression, and one can recover the hidden data without requiring the availability of the host signal.

## **C h a p t e r 3**

### **Image Embedding in the Wavelet Domain**

#### **3.1 Discrete Wavelet Transform: A Brief Review**

#### **3.2 Embedding Principle**

#### **3.3 Extraction principle**

#### **3.4 Similarity to Data Communication**

## Image Embedding in the Wavelet Domain

---

In this section we discuss some motivating factors in the design of our approach to watermarking. If embedding techniques can exploit the characteristics of the Human Visual system (HVS), it is possible to hide watermarks with more energy in an image, which makes watermarks more robust. From this point of view the discrete wavelet transform (DWT) is a very attractive transform, because it can be used as a computationally efficient version of the frequency models for the HVS [Barni, et. al., 1999b]. Research into human perception indicates that the retina of the eye splits an image into several frequency channels each spanning a bandwidth approximately one octave. The signals in these channels are processed independently. Similarly, in multiresolution decomposition of DWT, the image is separated into bands of approximately equal bandwidth on a logarithmic scale. It is therefore expected that use of the discrete wavelet transform will allow the independent processing of the resulting components without significant perceptible interaction between them, and hence makes the process of imperceptible marking more effective. Furthermore, our wavelet based watermarking framework is motivated by the fact that most network based images and video are in compressed

form, and that wavelets are playing an important role in upcoming compression standards such as JPEG2000 and MPEG-4. Furthermore, we can exploit the DWT decomposition to make real-time watermark applications.

In some of the recent work on using wavelets for digital watermarking, the signatures were encoded in high and middle frequency bands [Xia, et. al., 1998], [Kundur and Hatzinkos, 1997], [Kundur and Hatzinkos, 1998]. Such an embedding is sensitive to operations such as low pass filtering, JPEG lossy compression, and the Laplacian removal attack which has been found to be effective against several digital watermarking schemes that modify the mid to high frequency spectral components of the original image [Barnett and Pearson, 1998]. In contrast, the proposed scheme here focuses on hiding the signature mostly in the low frequency DWT bands, For these reasons, a digital signature should be placed in perceptually significant regions of the host data. Inserting signatures in the low frequency components creates problems if one is interested in invisible watermarks. This is particularly true in data hiding applications where the data to be hidden could be a significant percentage of the original data.

In the following, a brief review about the discrete wavelet transform and in particular, the Haar transform will be given. General embedding and extraction of the watermark is described in Sections 3.2 and 3.3. Finally, the similarity of the embedding algorithm with digital communication is given in Section 3.4.

### 3.1 Discrete Wavelet Transform: A Brief Review

In this section, we briefly consider and describe the notation used for the general 2-D discrete wavelet transform (DWT). The wavelet transform has been extensively studied in the last decade, and the reader is referred to [Daubechies, 1992], [Akansu and Smith, 1996], [Barlaud, 1994], [Leduc, 1994], and [Nievergelt, 1999] for a comprehensive description of the DWT. Many applications, such as compression, detection, and communications, of wavelet transform have been found. Here, we introduce the necessary concepts of the DWT for the purposes of this work.

The DWT refers to a discrete time framework for implementing the orthonormal wavelet transform. Since we are primarily concerned with images in this thesis we use the term “space” interchangeably for “time” when dealing with the 2-D wavelet transform. The transform decomposes a signal into basis functions that are dilations and translations of a signal function referred to as the *basic wavelet*. If the spectral overlap between basis functions is small, the wavelet coefficients provide an estimate of the frequency content in the signal localized to the corresponding frequency band and orientation. Similarly, given that the basic wavelet is localized in space, the coefficients provide a picture of the spatial development of the frequency contents.

#### 3.1.1 Wavelets

The basic idea in the DWT for a one-dimensional signal is the following. A signal is split into two parts, usually high frequencies and low frequencies. The edge components of the signal are largely confined to the high frequency part. The low

frequency part is split again into two parts of high and low frequencies. This process is continued an arbitrary number of times, which is usually determined by the application at hand. The significance of discrete wavelet analysis in comparison with discrete Fourier analysis lies in the decomposition and reconstruction process inherent in wavelet analysis. Discrete wavelet coefficient sets are analogous to images resulting from filtering the original image in the frequency domain using filters with bandwidths of one octave. To decompose an image using the standard discrete wavelet transform (DWT), two sets of two discrete, one-dimensional filters are used separately; a filter representing a *scaling function* and a filter representing the *wavelet function*. The filter representing the scaling function is usually denoted as  $h(n)$  and has a frequency response displaying low-pass characteristics. The filter representing the wavelet function is usually denoted as  $g(n)$  and has a frequency response displaying high-pass characteristics.

For example, the Daubechies wavelet is generated from the scaling function using

$$g(n) = (-1)^{1-n} \times h(1-n) \quad (3.1)$$

Hence  $h(n)$  can be used to generate  $g(n)$ . The functions used for decomposition are actually the above-mentioned functions reflected about zero ( $\tilde{h}(n)$  and  $\tilde{g}(n)$ ). The approximation and detail images are the result of different convolutional combinations of the scaling and wavelet functions. Reconstruction of the original image is the mirror operation of decomposition.

### 3.1.2 Why use Haar Wavelets?

The filtering technique based on the use of filter banks is well known in signal processing community. Under certain conditions, this technique can generate a highly useful orthogonal multiresolution analysis when the signal's characteristics are sought at different scales. However, it is impractical in image processing because the associated filters are nonlinear phase. Since images are, for the most part, smooth to the eye, it would seem appropriate to use exact reconstruction filters corresponding to an orthonormal wavelet basis with a mother wavelet exhibiting good regularity. In addition, in order to perform rapid convolutions, the FIR filters used must be short. On the other hand, these filters should be linear phase (and even zero phase). To avoid distortion in image processing, the filter  $H(\omega)$  associated with the scale function must be linear phase or ideally zero phase. Non-linear phase filters degrade edges and are more difficult to implement than linear phase filters. In addition, the number of elements making up the impulse response of  $h(n)$  must be small in order to limit the number of convolution operations to be performed in the analysis/reconstruction algorithm. It corresponds to wavelets whose support is compact (making the wavelet well localized). Unfortunately, not all of these conditions can be satisfied simultaneously since there are no orthonormal linear phase FIR filters enabling exact reconstruction, regardless of regularity. The only symmetric exact reconstruction filters are those which correspond to the Haar basis. Nonetheless, the highly important linear phase constraint corresponding to symmetrical wavelets can be maintained by relaxing the orthonormality constraint and by using biorthogonal bases. We can then

construct filters associated with wavelets exhibiting a high degree of regularity. However, when choosing filters for subband image decomposition, there are additional requirements that are specific to image coding. Analysis filters should have a short impulse response to preserve the localization of image features. Synthesis filters should also have a short impulse response in order to prevent spreading of artifacts resulting from quantization errors at edges and another local features. Long synthesis filters often have very good mean squared error performance but lead to annoying ringing effects around edges. In addition, linear phase filters are desirable for subband image coding. Filters with nonlinear phase introduce subjectively unpleasant waveform distortions, when the low pass channel is viewed by itself [Akansu and Smith, 1996]. There are many filter types available for general use but for this specific work the Haar filter and biorthogonal filters, biorthogonal (1.1), and (3.1) are found to be most suitable for the analysis. However, in the following chapters, the Haar filter are the mostly used wavelet filters.

### 3.1.3 Haar Wavelet Transform

The relationship between the filters  $h(n)$ ,  $g(n)$  and orthogonal wavelets was first established by Mallat [Mallat, 1989], who showed that the wavelet coefficients can be computed from a pyramidal transform implemented using digital filters.

The algorithm for one dimensional DWT and IDWT can be mathematically stated as follows,

Let  $s_0(n) \in Z$  be a sampled signal to be decomposed into several resolution levels corresponding to different space-frequency bands. This decomposition is achieved using the algorithm:

$$s_m(n) = \sum_k h(2n-k)s_{m-1}(k) \quad (3.2)$$

$$c_m(n) = \sum_k g(2n-k)s_{m-1}(k) \quad (3.3)$$

The signal  $s_m(n)$  is an approximation of signal  $s_{m-1}(n)$  at resolution  $2^{-m}$ , the coefficients  $s_m(n)$  and  $c_m(n)$  are called the DWT of signal  $s_0(n)$  at various bands of frequencies, and  $h(n)$  and  $g(n)$  are related by Equation 3.1.

The basic principle of the multiresolution analysis involves decomposing a signal  $s_0(n)$  into two subsignals  $s_1(n)$  and  $c_1(n)$ . This operation can then be repeated on signal  $s_1(n)$  and so on up to resolution  $2^{-j}$ . In this case, the signal set  $c_1 \cup c_2 \cup c_3 \cup \dots \cup c_j \cup s_j$  provides a lossless representation of  $s_0$ , and hence enables the exact reconstruction of this signal [Vetterli and Kovacevic, 1995]. Since the two filters  $h(n)$  and  $g(n)$  are associated with an orthonormal wavelet basis, they ensure the exact reconstruction of the signal  $s_{m-1}(n)$ . The reconstruction formula is as follows:

$$s_{m-1}(k) = \sum_k h(2n-k)s_m(n) + \sum_k g(2n-k)c_m(n) \quad (3.4)$$

The above reconstruction is called the inverse discrete wavelet transform (IDWT) of  $s_0(n)$ . To ensure the above IDWT and DWT relationship, certain conditions on the filters  $h(n)$  and  $g(n)$  has to be satisfied.  $H(w)$  and  $G(w)$  which are the Fourier transform of  $h(n)$  and  $g(n)$  respectively, defined as follows:

$$H(w) = \sum_n h_n e^{-jnw}, \text{ and } G(w) = \sum_n g_n e^{-jnw}. \quad (3.5)$$

are chosen such that  $H(w)$  generates multiresolution analysis and be of fixed number of coefficients to achieve compactly supported wavelets. Moreover, the relation between  $H(w)$  and  $G(w)$  must satisfy the following condition [Daubechies, 1992]:

$$|H(w)|^2 + |G(w)|^2 = 1. \quad (3.6)$$

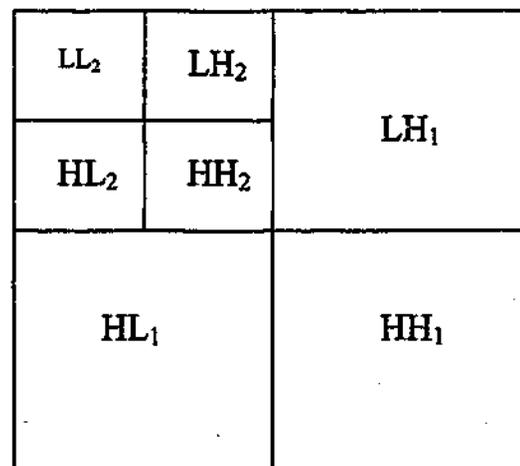
An example of such  $H(w)$  and  $G(w)$  is given by

$$H(w) = \frac{1}{2} + \frac{1}{2}e^{-jw}, \text{ and } G(w) = \frac{1}{2} - \frac{1}{2}e^{-jw}, \quad (3.7)$$

which are known as the Haar wavelet filters.

The above DWT and IDWT for a one dimensional signal  $s_0(n)$  can be also described in the form of two channel tree-structured filterbanks. The DWT and IDWT for two dimensional images  $s_0(n_x, n_y)$  can be similarly defined by implementing the one dimensional DWT and IDWT for each dimension  $n_x$  and  $n_y$  separately. An image can be decomposed into a pyramid structure, as shown in Figure 3.1, with various band information: such as low-low frequency band, low-high frequency band, high-high frequency band etc. To better illustrate this principle, Figure 3.2 shows an example of multiresolution decomposition of *Barbara* image with two levels.

In this thesis, we make use of Daubechies wavelets [Daubeches, 1992]. The discrete Haar wavelet transform discussed in this work is a particular case of this class. Our work does not involve designing or selecting particular basic wavelet for



**Figure 3.1** DWT pyramid decomposition of image.

optimal watermarking. Instead, we use well-established wavelets in our work to develop multiresolution embedding methods. We make use of wavelets for their spatial and frequency localization and concentrate on developing improved embedding methods through the use of communication theory and error-control coding.

### 3.2 Embedding Principle

In this section we explain the generic embedding principle by means of the diagram in Figure 3.3. The signature data is first source-coded either losslessly or lossily depending on the nature of the data, to generate a sequence of symbols. The signature data or watermark could be a binary sequence or an image and may be an encrypted version of the author identification which is used to establish sender credibility or a mixture of the above mentioned types of watermarks. There are three main stages to the data embedding procedure shown in Figure 3.3. The host



**Figure 3.2** One level of discrete wavelet decomposition of *Barbara* image.

signal is first transformed into the discrete Haar wavelet domain where its coefficients are grouped into vectors  $v_j$ . The embedding process inserts one signature symbol in each coefficient vector  $v_j$  of the DWT coefficients of the host signal. The signature data is inserted into the host with the use of noise-resilient channel code by scaling it by a parameter  $\alpha$ , which determines the transparency constraint. The perturbed coefficients are inverse transformed back to form the embedded or watermarked image.

The step-by-step procedure of the embedding process is described more precisely as follows:

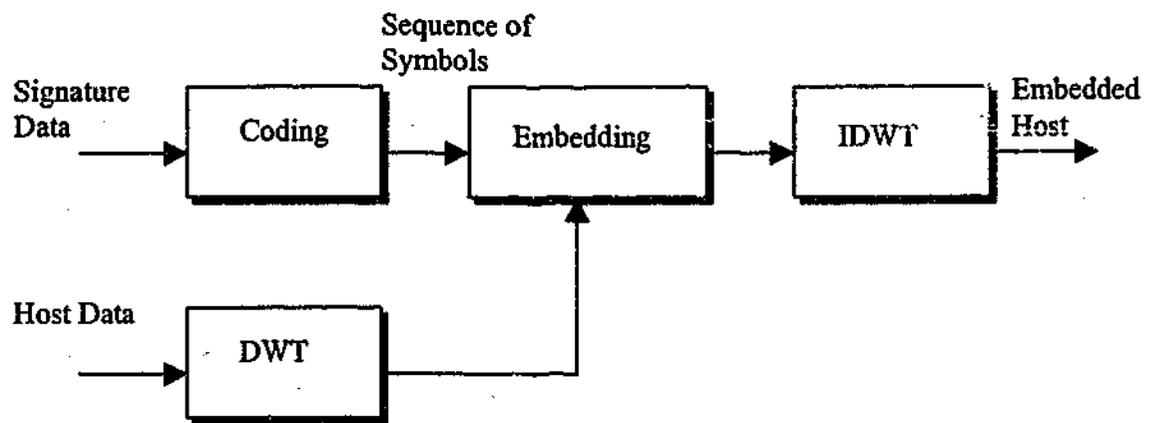


Figure 3.3 The Embedding Principle.

### Stage 1

We compute the  $L$ -level discrete Haar wavelet transform (DHWT) of the host image to produce a sequence of  $3L$  detail images, corresponding to the horizontal, diagonal and vertical details at each of the  $L$  resolution levels, and a gross approximation of the image at the coarsest resolution level. The value of  $L$  is user-defined. We choose the Haar transform in particular for its low complexity, and fixed-point arithmetic.

### Stage 2

In the second stage, we group selected wavelet coefficients into vectors  $v_j$ . These coefficients could be selected randomly by using a secret key so that the watermark is spread throughout the image. Alternatively, the selection of the coefficients could be confined to the fixed low frequency coefficients to ensure the robustness of the embedding against attacks. To actually inlay the signature bits within the selected coefficients, we incorporate the following function

$$\tilde{v}_j = v_j + \alpha C(s_i), \quad (3.8)$$

where the set of vectors  $C(s_i)$ , constitute a channel codebook for each signature symbol  $s_i$  and  $\alpha$  is a scale factor controls the transparency of the watermarked image. Equation (3.8) is similar to the embedding algorithm adopted in [Cox, et. al., 1996], however, since the watermark is of binary nature we used the additive insertion of the watermark instead of the multiplication insertion adopted in [Cox, et. al., 1996] for a watermark generated from a sequence of real numbers.

Each index of the signature image is hidden into N-coefficients in the LL band of the host; the remaining indices, if any, are hidden in the other subbands of the host (HL, LH, and HH). The scale factor  $\alpha$  for embedding is chosen in a way that assure an acceptable quality for the watermarked image. The mean square error between the original host image and the embedded image could be used in choosing suitable values for the parameter  $\alpha$ . However, it is well known that the mean square error or energy of the watermark does not correlate well with human perception. Better models of the human visual perception for watermark visibility could be involved in this procedure. For example, the spatial masking model of Girod [Girod, 1989] can be used to adjust the watermark strength (or the scaling factor  $\alpha$ ), so that the watermarked image is perceptually identical to the original image. The model is based on the physics of human visual perception and accurately describes the visibility of artifacts around edges and in flat areas in digital images. The model is a general model and can be applied to videos. In our work, only the spatial portion of the model is used.

Further, from Equation 3.8, the length of the coefficient vectors  $v_j$  and the channel codebook  $C(s_j)$  should be the same.

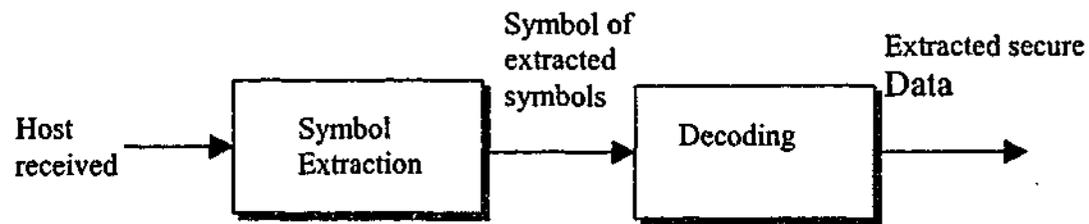
### Stage 3

The corresponding L-level inverse DHWT of the marked image components is computed to form the embedded or watermarked image before transmission or distribution.

The above scheme has two layers of security. The variability of the source and channel codebooks used makes unauthorized retrieval virtually impossible. The knowledge of the algorithm alone is not sufficient to extract the hidden information. The exact source and channel codebooks used for any application must be known. Further, if an encryption key is used in shuffling the transform coefficients before embedding, then additional layer of security is added. Moreover, if we suppose that the attacker knows the exact coefficients used for data embedding, he still cannot retrieve it without knowledge of either the source codebook or the channel codebook. Depending on the level of attack, an attacker may be able to destroy the hidden information, but in this process he cannot do it without significantly degrading the watermarked host.

### 3.3 Extraction principle

The extraction principle is outlined in Figure 3.4. The L-level DHWT is applied to the given image and the coefficients are grouped in the same way as in the encoder.



**Figure 3.4** The Extraction Principle.

The grouping can be done either using the random key or priory knowledge of the watermarked coefficients to determine the location of the embedded watermark bits. To extract the watermark bits from its associated wavelet coefficients, let us say that the  $j$ th perturbed vector  $\tilde{v}_j$ , corresponding to a hidden symbol  $s_i$ , has been received as  $w_j$ , as a result of additive noise  $n_j$  due to compression and other transformations:

$$w_j = \tilde{v}_j + n_j \quad (3.9)$$

The process of extraction is then formulated as a statistical estimation problem that estimates the transmitted symbol from the noisy version received. The extraction process uses its knowledge of the original host to decode, from each received vector, the symbol within whose decision boundaries the received perturbation lies. In other words, a nearest neighbor search with an appropriate distance measure is used. The sequence of the extracted symbols is then source-decoded to obtain the extracted watermark.

### 3.4 Similarity to Data Communication

Digital watermarking of multimedia can be viewed as a special communication problem. Many authors have drawn comparison between the principles of data embedding and that of communications, especially spread spectrum communications [O'Ruanaidh, et. al., 1996], [Ramkumar and Akansu, 1998], [Su, et al., 1999], [Cox, et. al., 1999]. In [Cox, et. al., 1999], suggestion has been made that watermarking most closely resembles communications with side information at the transmitter and or receiver, a configuration originally described by Shannon. A message (information to be embedded) is converted into a signal (the watermark), which is then sent through a channel to the receiver. The receiver must locate the watermark signal and attempt to recover the message from it. The channel is referred to as the watermark-channel to distinguish it from a conventional broadcast channel.

An attack is an operation, performed on the watermarked document, that may degrade a watermark and possibly make the detection of the watermark impossible. From a communications point of view, even coincidental manipulations such as lossy compression or cropping, are attacks. Attacks are assumed to occur only in the channel. While there are various noise models available for various kinds of channels, we will assume that the noise is of an Additive White Gaussian Noise (AWGN). This particular noise model approximates many real channels and also makes analysis simpler. The task of the receiver is then to estimate the symbols transmitted from the noisy waveform that is received. In a correlation receiver, the noisy signal is correlated with all the orthogonal basis signals, to obtain a set of

sufficient statistics that also represent a point in the same  $k$ -dimensional orthogonal signal space. In a noise free channel, the received signal point is exactly the same point as the one transmitted. However, as a result of noise, the received vector is different from the one transmitted. Using a maximum-likelihood decoder in the transform domain then yields a decoding rule, which for every vector received, chooses the symbol to whose channel code is closest in Euclidean distance.

In summary, the similarities between digital communication system and data hiding system can be summarized as in Table 3.1. Further, to illustrate the importance of maintaining the fidelity of the watermarked media content. One can look at the decoding process as if there is an added second *receiver* in the form of human sensory organs, which should receive a message that is essentially the same as the carrier media content. While this represents a deviation from the classical communications, in which the carrier signal's sole function is to carry the encoded message, the resulting fidelity constraint is analogous to a constraint on signal power in a communication channel, albeit with different metric and motivation.

From the discussion of data embedding techniques so far, it will be appreciated that the above dual problem of data embedding and watermarking, freely map to the source and channel coding problem in digital communications. As such, established concepts from digital communications could be used to solve this problem. This is explained in more detail in the next chapter.

**Table 3.1** Comparison of data hiding system to digital communication system.

<b>Digital Communication System</b>	<b>Data Hiding system</b>
In a classical communications system , the message to be transmitted is first encoded. This step typically takes a binary input stream and translates it into a binary output stream, usually for error correction and/or frequency spreading purposes.	Similar technique can be used for encoding the message bits before hiding them into the host signal.
The encoded message is then used to modulate a carrier signal in any of variety of ways, e.g. amplitude, frequency, phase, etc.	Here the modulation step is replaced by the embedding process of the encoded message. There is no passband modulation. The signal messages are all baseband signals.
The modulated carrier signal is transmitted via a transmission channel, where it encounters additive noise.	In watermarking, the noise in the transmission channel results from various types of processing that the watermarked media goes through before it received, e.g. compression and decompression, image or audio enhancements, etc. it might also result from malicious processing by pirates intent on removing the watermark.
The receiver demodulates the noisy signal to a (possibly corrupted) encoded message, and finally this message is decoded to produce the received message.	Here the demodulation step is replaced by the step of extracting the (possibly corrupted) encoded watermark signal from the received signal and then decoded to its original state.
Constraint on signal power in traditional communications (small signal to noise ratio).	The requirement that the fidelity of the media content must not be impaired implies that the magnitude of the watermark signal must be very small compared to the host signal.
Low bit rate communication.	The amount of information conveyed by the watermark, in number of bits, is small when compared to that of the original image.

## **Chapter 4**

### **Data Embedding using Source and Channel Coding**

#### **4.1 Data Embedding using Vector Embedding**

#### **4.2 Source Coding: Vector Quantization**

#### **4.3 Block Coding**

#### **4.4 Interleaver**

#### **4.5 Implementation and Experimental Results**

#### **4.6 Summary**

## Data Embedding using Source and Channel Coding

---

In general, developing techniques that are robust to image processing operations is at the core issue of digital watermarking and data hiding. We are primarily interested in techniques that result in invisible watermarks. Quantity of the data that can be embedded without much perceptual distortion to the host is also an important issue. In this chapter, we present data embedding scheme that is suitable for both watermarking and data embedding or hiding. While watermarking requires robustness under image manipulation, data embedding aims at hiding large amounts of data with little perceptual distortion to the host.

In this chapter we consider the problem of hiding text messages and images in images. We specifically address the robustness to data compression and noise addition. Lossy compression techniques, such as JPEG, typically affect the high frequency components. This is also true with most perceptual coding techniques based on the human visual system. For these reasons, a digital signature should be placed in perceptually significant regions of the host data. For techniques based on frequency domain modifications, this implies embedding the signature in mostly low frequency components. On the other hand, inserting signatures in the low

frequency components creates problems if one is interested in invisible watermarks. This is particularly true in data hiding applications where the data to be hidden could be a significant percentage of the original data. The embedding algorithm adopts a data compression technique to the signature using vector quantization before embedding in the host image.

In this work, we present a data hiding method that allows large-scale image to image embedding that is robust to various compression techniques and low-pass filtering. The data is embedded in the wavelet transform domain. In the following sections, data embedding using source and channel coding is explained in section 4.1, section 4.2 discuss the extracting procedure. Experimental results are presented in section 4.3, and we conclude with discussions in section 4.4.

## 4.1 Data Embedding using Vector Embedding

The embedding and extracting of the digital watermarking system are similar to the encoder and decoder of the digital communication system. The requirement that the fidelity of the host signal must not be impaired implies that the magnitude of the watermark signal be very small to the host signal. This is analogous to the strict power constraint in traditional communication system. This characteristic has led to the idea of using channel coding techniques in the embedding process.

In this section, we discuss our data hiding approach, where vectors of wavelet transform coefficients of the host are modified using channel codes to represent source coded symbols.

According to the embedding technique in Chapter 3, the host data is orthogonally transformed before embedding the hidden signature in it. Let us consider a host data source  $(X_1, X_2, \dots, X_N)$  transformed orthogonally to a set of  $N$  coefficients  $(Y_1, Y_2, \dots, Y_N)$ . The transform-domain embedding of watermark signal modify the coefficients into a new set of coefficients given by  $(\hat{Y}_1, \hat{Y}_2, \dots, \hat{Y}_N)$ . The inverse transformation then yields the embedded host  $(\hat{X}_1, \hat{X}_2, \dots, \hat{X}_N)$ . Since the transformation is orthogonal, the mean squared error introduced in the coefficients is exactly equal to the mean squared error introduced in the host data [Servetto, et. al., 1998]. That is,

$$MSE = \frac{1}{N} \sum_{i=1}^N |X_i - \hat{X}_i|^2 = \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i|^2 \quad (4.1)$$

With a transparency constraint imposed on the value of  $MSE$ . This specifies a maximum value  $P$  which upper bounds  $MSE$  for a given application:

$$\frac{1}{N} \sum_{i=1}^N |X_i - \hat{X}_i|^2 < P \Rightarrow \frac{1}{N} \sum_{i=1}^N |Y_i - \hat{Y}_i|^2 < P \quad (4.2)$$

The smaller the value of  $P$ , the more transparent the embedding is, and vice-versa. The value of  $P$  could be computed based on models of the human visual system that has been studied in the context of perceptual coding.

At this stage we can explain the general embedding principle by means of the diagram in Figure 4.1. The signature data is first source coded, either losslessly or lossy, to generate a sequence of symbols. These coefficients are channel coded before adding them to the host coefficients. The modified coefficients are inverse transformed to obtain the embedded host.

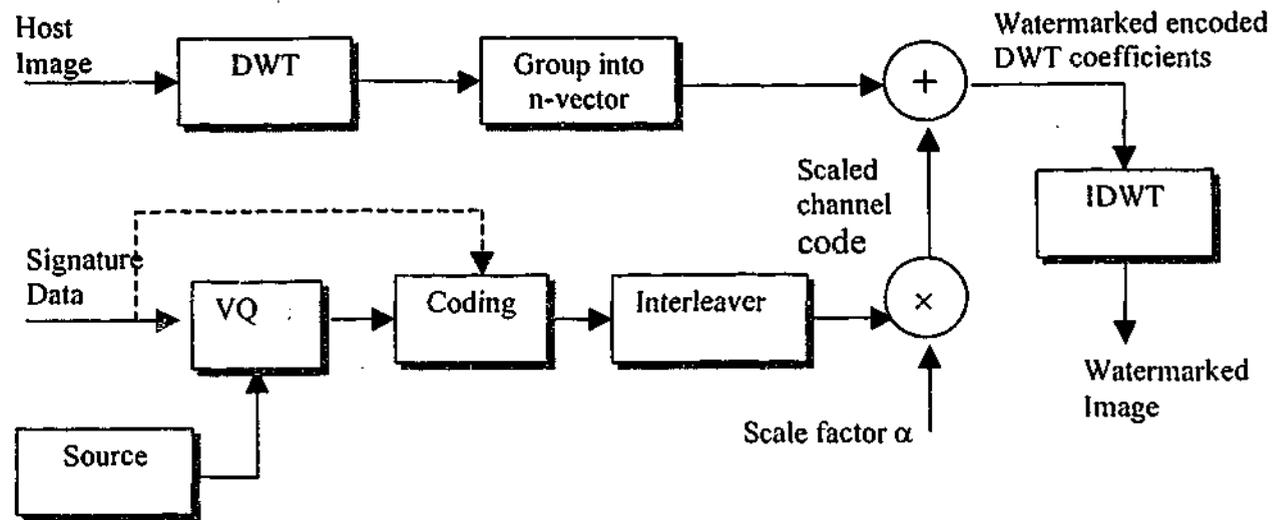
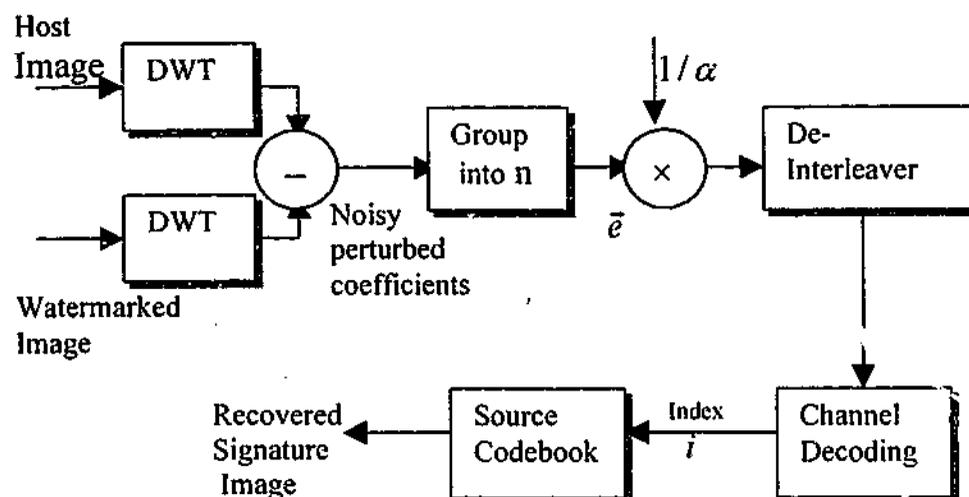


Figure 4.1 General block diagram of the encoder.

The extraction principle is outlined in Figure 4.2. The discrete wavelet transform is applied to the given image and the coefficients are grouped in the same way as in the encoder. The recovery process thus extracts from each vector the symbol within whose decision boundaries the received vector lies. In other words, a nearest neighbor search with an appropriate distance measure is used. The decision boundaries depend on the statistical model chosen for the additive noise. The sequences of extracted symbols are then decoded to obtain the extracted signature.

Finally, with increase in the amount of signature data, it makes sense to lossily source code the data if it is compressible. A method that works well for correlated sources is vector quantization. The indices obtained by vector quantization are embedded into the host transform coefficients by vector modifications derived from noise - resilient channel - codes. Note that it is also possible to design channel-



**Figure 4.2** General block diagram of the decoder.

optimized VQs, or power constrained VQ for better noise performance [Fossorier, et. al., 1998].

The vector quantizer using Linde-Buzo-Gray (LBG) algorithm is explained in more detail in the next section.

## 4.2 Source Coding: Vector Quantization

For most data hiding applications it is necessary to embed watermark data at a high rate. Therefore, it makes sense to compress the secure data losslessly or lossily before embedding. If the secure data is compressible, lossy schemes can be used for achieving lower rates. A scheme that works well for correlated sources is vector quantization (VQ) [Gersho and Gray, 1992]. Vector quantization (VQ) exploits the statistical redundancy between pixels to reduce the bit rate. The input data are divided into blocks and then tested against a set of code vectors to find the best

match. The index of the corresponding code vector is used in the embedding algorithm, and the decoder uses this index to extract the code vector from a local copy of the codebook [Bull et. al., 1999]. Moreover, VQ image coding can provide fixed length code words, which are useful for error-resilient coding applications since channel errors can not propagate between codewords [Doufexi et. al., 2000]. The codebook design in this study is performed by using Linde, Buzo, and Gray (LBG) algorithm. In this algorithm, the codebook is generated using a training set of images where the data to be compressed is used for the training set [Linde et. al., 1989], [Gersho and Gray, 1992]. In another word, the LBG algorithm is a generalization of the Lloyd-Max algorithm. The algorithm can be described as follows:

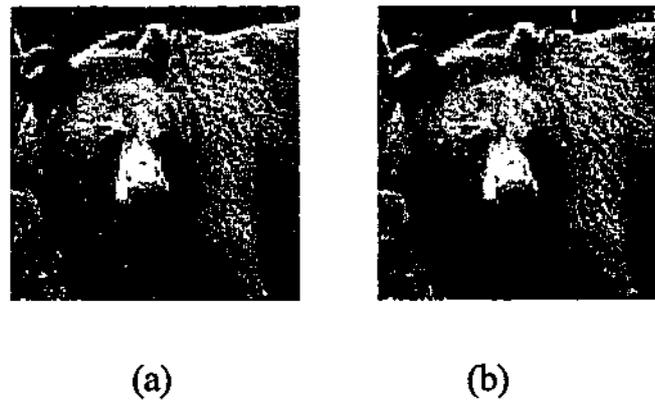
- The algorithm starts with a good initial codebook, often found using other methods such as Pairwise Nearest Neighbor (PNN).
- Encode the training set by mapping each vector in the training set to its nearest codevector.
- Compute the average distortion resulting from the encoding process.
- If the fractional change in the average distortion from the previous iteration is less than or equal to a certain threshold, then the convergence has been achieved and the algorithm terminates.
- Update the codebook by replacing each codevector within a decision region by a new codevector that minimizes the quantization error within that decision region.

- For the mean square error (MSE) measure, the minimum distortion vector is the average of the training vectors enclosed by that decision region.
- While each iteration results in a non increasing distortion, convergence may take many iterations if the threshold is set too low. Therefore, it's necessary to terminate after a maximum number of iterations.

To illustrate the concept of vector quantization. First, the 8-bit/pixel monochrome test image of *Bear* of size 128 x 128 is decomposed into 4-dimensional image vectors; in this case the signature image is divided into 2 x 2 blocks. Each vector is compared with a collection of representative codevectors taken from a previously generated codebook (the source codebook). Best match codevector is chosen using a minimum distortion rule. After the minimum distortion codevector has been found the index  $i$  is used to represent the signature vector. For 4-dimensional vector quantizer, the codebook contains 16 levels divided into 2x2 dimension blocks. The compression is 1/4., corresponding to a transmission rate of 2 bits/pixel. Figure 4.3 shows the VQ-coded image of *Bear*. As can be seen from the Figure, the VQ compressed image is still of high quality.

The peak signal to noise ratio (PSNR) of an image is a measure of the distortion of an image relative to a reference image. It can be used to measure the distortion of an image due to compression or transmission errors, compared with the original image. PSNR is defined as:

$$PSNR(dB) = 10 \log_{10} \frac{(2^n - 1)^2}{MSE} \quad (4.3)$$



**Figure 4.3** Vector quantization of *Bear* image. (a) Original *Bear* image, (b) Reconstructed image with PSNR = 34.64 dB.

where  $n$  is the number of bits required to represent each pixel in the original image ( $n = 8$ ) and MSE is the mean squared error between the distorted and the original image [Jayant and Noll, 1984]. The PSNR in our example was 34.64 dB.

Even with compression of the watermark data, the rate through the channel (measured in terms of bits/pixel) may be too high to support error-free communication. For example, the watermarked host may be compressed too severely. In this case, the recovered watermark image will be severely distorted. Hence, for a given transparency constraint for the host image, we need to have a form of error correction to allow reliable recovery of the hidden data. In such case, it is advantageous to combine source and channel coding by using VQ and error-control coding. Therefore, the indices obtained by vector quantization form the alphabet that is embedded into the host transform coefficients after it has been channel coded using error-correction codes.

### 4.3 Block Coding

The performance of a reliable information hiding scheme can be improved by means of coding. Coding is an effective way of reducing the probability of bit error by creating interdependencies between the transmitted symbols at the expense of an increased complexity.

In this chapter, we first deal with block codes and then in the next chapter will proceed to describe how convolutional codes can also be used for data hiding purposes.

Suppose that instead of transmitting raw information symbols through the hidden channel, we use a  $(n, k)$  block code that maps  $k$  information symbols into  $n$  binary antipodal channel symbols  $C[i] \in \{\pm 1\}$ ,  $i = 1, \dots, n$ . From the way it is constructed, it is clear that this code consists of a total of  $2^k$  codewords, each with  $n$  binary antipodal symbols. In order to use this code for data hiding, the set of  $N$  source information bits is divided into  $N/k$  blocks and each block of size  $k$  bits mapped into  $n$  symbols that are hidden using a procedure for watermark insertion similar to that summarized in Equation 3.8. Regarding watermark extraction, two strategies are possible: hard and soft decision decoding, each admitting different implementations and simplifications. For the case of block coding, only hard thresholding is used.

### 4.3.1 Why use Error-Control Coding?

Error correcting codes or block codes, provide coders with a tool to recover lost information such as errors, erasures and deletions.

The purpose of error-control codes is to improve the capacity of a channel by adding some carefully designed redundant information to the data being transmitted through the channel [Shannon, 1948]. The process of adding this redundant information is known as channel coding. Convolutional codes operate on serial data, one or few bits at a time. Block codes operate on relatively large message blocks. There are variety of useful convolutional and block codes, and a variety of algorithms for decoding the received coded information sequences to recover the original data. The properties of block codes are well known and we will make use of some of the knowledge about them to construct some of the following codes.

### 4.3.2 BCH Codes

BCH codes, named after the inventors, Bose, Ray-Chaudhuri, and Hocquenghem, are a large class of multiple-error correcting codes invented around 1960. For any positive integers  $m$  and  $t$ , there is a  $t$ -error-correcting binary BCH code with

$$n = 2^m - 1 \quad k \geq n - mt \quad (4.4)$$

in order to correct  $t$  errors, it is clear that the minimum Hamming distance is bounded by

$$d_{H,\min} \geq 2t + 1 \quad (4.5)$$

BCH codes are important primarily because practical and efficient decoding techniques have been found [Lin and Costello, 1983], and because of the flexibility in the choice of parameters  $(n, k)$ .

A code with codewords of length  $n$  and data words of length  $k$  is called  $(n, k)$  code. Its codeword consist of  $k$  information bits and  $n-k$  redundant, so-called parity bits. The codebook  $C$  of an  $(n, k)$  code contains  $2^k$  codewords. An  $(n, k)$  code can be characterized by its minimal distance

$$d_{\min} = \min_{v_1, v_2 \in C} d_H(v_1, v_2) \quad (4.6)$$

A code with  $d_{\min}$  can correct

$$t = \left\lfloor \frac{d_{\min} - 1}{2} \right\rfloor \quad (4.7)$$

errors.

The error detection and correction codes for the signature data are based on the principles, which follows. The discrete information to be embedded on a host image is segmented into words of  $k$  binary symbols. Each word can represent  $2^k$  different messages. To reduce the errors, the  $2^k$  possible messages out of  $k$  bits are transformed into  $2^n$  distinct messages contained in words of  $n$  bits. As a matter of fact,  $n$  is larger than  $k$  and only  $2^k$  words taken among the  $2^n$  correspond actually to useful messages, the other possibilities  $(2^n - 2^k)$  are a reserve for redundancy available for control operations. The result is a  $(n, k)$  code where  $k$  symbols carry information and  $(n - k)$  bits allow error-detection and correction controls. The code rate of a block is defined by the ratio  $k/n$ .

### 4.3.3 Reed-Solomon Codes

Reed-Solomon (RS) codes are burst error-correcting codes. All of the other coding methods like BCH, Cyclic, Hamming, and Block codes are random error-correction codes. A burst error occurs when noise corrupts several consecutive bits.

The theory of Reed-Solomon code is based on finite field theory. In particular, the fields used are of the form Galois Field ( $GF(q^M)$ ), where  $q$  is any prime number and  $M$  any positive integer. The elements of  $GF(2^M)$  are defined by a power series format. The message length  $k$  can be any positive integer smaller than  $n$ , where  $n$  is the codeword length.

The basic parameters of RS code are:

- Codeword length:  $n = 2^M - 1$
- Number of check symbols:  $n - k = 2t$
- Error-correction capability:  $t = ((n - k)/2)$

With the exception that the elements of RS code are in  $GF(2^M)$ . The data input for RS code/decode can be one of the three forms: binary, integers in the range from 0 to  $2^M - 1$ , or power with the elements in  $GF(2^M)$ .

### 4.3.4 Description of Codes by Generator Matrices

For an  $(n, k)$  code, the codebook  $C$  has  $2^k$  members, so that  $C$  can hardly be described by enumeration of its members. A more efficient, but nevertheless complete description is done by a generator matrix  $G$ .

$G$  has the size of  $n \times k$  and encoding work as

$$r = x G \quad (4.8)$$

all operations are done in the Galois field  $Gf(2)$ .

A code is called systematic, if its codewords  $r$  consist of the unmodified information vector  $x$  followed by a number of parity symbols, so that

$$r = (x_1, \dots, x_k, r_{k+1}, \dots, r_n) \quad (4.9)$$

For the generator matrix  $G$  this means that the first  $k$  columns contain a  $k \times k$  unity matrix  $I$  and matrix  $P$  such that  $G$  has the form  $G = [I, P]$ , where  $P$  is an  $k$ -by- $(n-k)$  matrix (note that some authors define the generator matrix as  $[P, I]$ ) [Reed and Chen, 1999].

$$G = \begin{pmatrix} 1 & 0 & \dots & 0 & G_{k+1,1} & \dots & G_{n,1} \\ 0 & 1 & \ddots & 0 & G_{k+1,2} & \dots & G_{n,2} \\ \vdots & \ddots & \ddots & \vdots & \vdots & & \vdots \\ 0 & \dots & 0 & 1 & G_{k+1,k} & \dots & G_{n,k} \end{pmatrix} \quad (4.10)$$

It has been shown that for any generator matrix  $G$  an equivalent systematic form with the same code properties can be found. This can be done by either permuting any rows of  $G$  or replacing any row of  $G$  by a linearly independent combination of rows [Lin and Costello, 1983]. Moreover, for each generator matrix  $G$ , it is possible to find a corresponding parity check matrix  $H$  that will be used for the decoding of the block codes.

The recovery of the length  $k$  message from the codeword involves the calculation of a matrix called the syndrome. The *syndrome* is a matrix that, when multiplied by the codeword vectors, produces the original message plus a length  $(n-k)$  set of error-correction bits. Within the limitations of the designed code, the syndrome detects the number of errors contained in the received message.

if we assume that the received vector  $r$  of  $n$  bits is

$$r = c + e \quad (4.11)$$

Where  $e$  is sometimes called the error vector. Then the decoding process can be described with the following four steps:

1. Compute the syndrome of  $r$ .
2. Locate the error  $e$  by using the syndrome.
3. Decode the received vector into the code vector  $c = r - e$ .
4. Recover the message vector from the reconstructed codeword vector  $c$ .

The syndrome is computed by  $s = r H^T$  where  $H$  is called a parity-check matrix.  $H$  is a null space matrix of the generator matrix  $G$ .  $H$  is a  $((n-k)$ -by- $n$ ) matrix, which has the property that  $G H^T = 0$ .

Since  $r = c + e$ ,  $c = x G$ , and  $G H^T = 0$ , we have

$$s = r H^T = (c + e) H^T = c H^T + e H^T = e H^T \quad (4.12)$$

which means that the syndrome is a linear function of the transmission error.

There are several decoding methods used depending on the type of the block codes. For short codes syndrome decoding is quite efficient, on the other hand, for long codes bounded distance decoding such as Berlekamp-Massey algorithm is usually used [Blahut, 1984]. Moreover, regarding watermark extraction, two strategies are possible: hard and soft decision decoding, each admitting different implementations and simplifications. In this chapter only hard-decision decoding will be implemented for decoding block codes, details on trellis decoding of block is found in [Lin, et. al., 1998]. Soft-decision decoding for trellis codes will be covered in the next chapter.

### 4.3.5 Hard-Decision Decoding

In this case, an independent threshold-based decision is taken for each symbol of the transmitted codeword, producing a received word. Then, the codeword with minimum Hamming distance to the received word is chosen. Note that this two-step decoding process is not optimal in the maximum-likelihood (ML) sense, but gives good results at a low computational cost, since efficient decoding algorithms are available for certain types of block codes (generally belonging to the class of linear codes) [Blahut, 1990]. It is worth noting here that soft-decision decoding using the maximum likelihood decoder could be implemented for block coding, however, in our work, only hard decision decoding for block codes is used.

When trying to assess the performance of hard decoding, one finds that the number of errors depends in a complicated manner with the information sequence and the type of partition used for embedding, so this hinders obtaining an exact expression for the probability of bit error. However, useful approximations can be given for many cases of interest, particularly for perfect linear codes [Proakis, 1995]. First, instead of the bit error probability, it is simpler to obtain the probability of block error or codeword error, that is, the probability of incorrectly decoding a certain transmitted codeword  $C(s)$ . This probability, denoted here by  $P_c(\varepsilon)$ , can be used to bound the bit error probability  $p$  and its derivation is shown below.

### 4.3.6 Probability of Codeword Error

In this section we wish to compute the probability of codeword error  $P_c(\varepsilon)$  that a bounded distance decoder will fail.

From previous discussion, we know that the decoder can correct up to, but not more than

$$t = \lfloor (d_{H,\min} - 1) / 2 \rfloor \text{ errors} \quad (4.13)$$

where  $d_{H,\min}$  is the minimum Hamming distance (minimum number of differing antipodal symbols) between any two codewords.

If we assume that the probability of an individual symbol error is  $p$ , and that symbol errors occur independently, then if we send  $n$  bits, the probability of receiving a specific pattern of  $i$  errors and  $n-i$  correct bits is:  $p^i(1-p)^{n-i}$

there are  $\binom{n}{i} = \frac{n!}{i!(n-i)!}$  distinct patterns of  $n$  bits with  $i$  errors and  $n-i$  correct bits,

so the total probability of receiving a pattern with  $i$  errors is:

$\binom{n}{i} p^i (1-p)^{n-i}$  and since we can correct any pattern of up to  $t$  errors, then the

overall probability of codeword error is :

$$P_c(\varepsilon) = 1 - \sum_{i=0}^t \binom{n}{i} p^i (1-p)^{n-i} = \sum_{i=t+1}^n \binom{n}{i} p^i (1-p)^{n-i} \quad (4.14)$$

using equation 4.17, for example, the probability of codeword error for BCH code of (4,7) will be  $2 \times 10^{-5}$ .

#### 4.4 Interleaver

Interleaving is used to convert burst errors into random errors in error-control coding. It can be used for both block error-control coding and convolutional error-control coding. The data input rate is the same as the data output rate. A de-

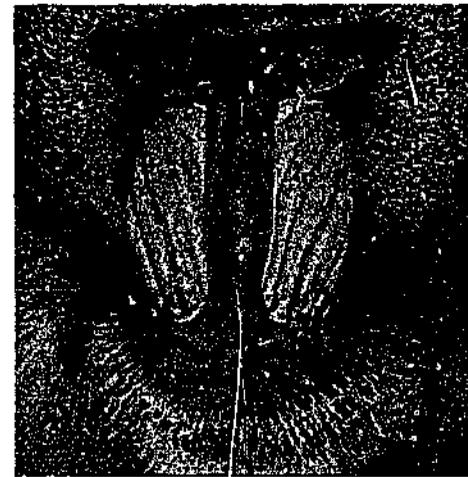
interleaving process reverses the interleaving operation. Typically, one selects the interleaver parameters so that the number of columns,  $n$ , is greater than the expected burst lengths. The choice of the number of rows depends on the type of error-control-coding scheme used. For block coding, the number of rows should be greater than the codeword length; thus, a burst of length  $n$  can cause at most a single error in any block codeword.

#### 4.5 Implementation and Experimental Results

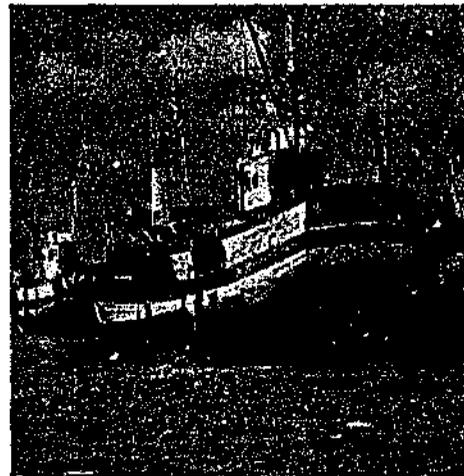
The test images used in our experiments are shown in Figure 4.4. They are: *Lena*, *Baboon*, and *Fishingboat* among other test images. All images were of scale 256 x 256 pixels. The images shown in Figure 4.4 are at 80% of its actual size. Different types of signatures were used, for example, image data (*Bear* image), of size 128 x 128 pixels gray scale and ASCII text file of length 890 bits, and random messages. All the experiments described below use the discrete Haar wavelet basis. Different values for the scale factor  $\alpha$  were used depending on the length of the watermark and the number of wavelet decomposition used. For example,  $\alpha = 10$  was used for image embedding in images and  $\alpha$  between 10 and 40 was used for hiding text messages.  $\alpha = 10$  corresponds to 2% of the DWT coefficients amplitudes for one level decomposition, and around 1% for two-level decomposition. To measure the robustness, we used the bit-error rate, the peak signal-noise ratio PSNR of the recovered watermark image, and the similarity measure. The attacks were JPEG compression and noise addition.



(a)



(b)



(c)



(d) Signature images (128 x 128)

**Figure 4.4** Test host and watermark images used, (a) Host *Lena* image (256 x 256), (b) *Baboon* (256x256), (c) *Fishingboat* (256x256), and (d) Signature images of *Bear* and *Peppers*.

The bit error versus attack strength graphs relate the watermark robustness to the attack, where the bit-error rate is plotted as a function of the attack strength for a given visual quality. This type of evaluation allows the direct comparison of the watermark robustness and shows the overall behavior of the method towards attacks. The bit error is measured by comparing bit by bit of the extracted signature to the original signature (watermark). The next sections describe the results of different embedding performance.

#### 4.5.1 Lossless Data Embedding

Much of the recent digital watermarking research is concerned with robustness to signal processing operations. Since the watermark is needed for authentication, lossless recovery is not a primary requirement. In general, in data hiding, lossless recovery may not be the main requirement if the embedded signal is an image, audio or video data. However, lossless recovery of embedded data would enable new application. While there exists some simple methods for lossless encoding and decoding, these methods are not robust to even small changes to the embedded signal.

Using the algorithm discussed in this chapter, we are able to embed and extract losslessly certain type of data such as text messages. The alphanumeric characters of the text are transformed to channel codes and are then embedded into the host wavelet coefficients. In this embedding, Reed-Solomon (RS) codes were utilized to encode the text message before embedding using RS(127,64). The text message "We are investigating data embedding using error-correcting codes", of length 890 bits, in ASCII file format, is embedded in the host *Lena* image. The original *Lena*

image and the embedded image with the hidden data is shown in Figure 4.5 with PSNR = 35.56 dB, and a scaling factor  $\alpha = 40$ , which produced watermarks with less than 3% of pixels with visible changes using Girods' model. The block coding used was BCH (15,7) and an interleaver of size (16 x 60).

In Figure 4.6 a plot of bit error (BER) for different levels of JPEG compression at different quality factors is shown for hiding text into the three test images. From this figure, it is clear that the quality factor at which the watermark is lost is around  $Q = 30\%$  which is much better than previous results published in reference [Swanson et. al., 1996] that was obtained using both spatial and frequency data hiding techniques. This Figure demonstrates the lossless recovery of the signature data even at high JPEG compression ratio.

The bit error rate for extracting text messages from three different host images is shown in Figures 4.7 as a function of JPEG compression. Figure 4.8 shows the bit error rate as a function of the PSNR of the watermarked image after noise addition. The noise addition attack is created by adding Gaussian noise with different variance to the watermarked image.

In checking for the presence of a signature, the quality of the signature is not an issue. A binary decision for the presence or absence of a signature needs to be made. We use a measure similar to the one defined in [Cox et. al., 1997] to compute the cross correlation between the recovered signature  $s^*(m,n)$  and the original signature  $s(m,n)$ . This similarity is defined as:

$$S = \frac{\sum_{m,n} s^*(m,n)s(m,n)}{\sum_{m,n} (s(m,n))^2} \quad (4.15)$$



(a)



(b)

Figure 4.5 Test images (a) Host *Lena* image, (b) Image with hidden text message.

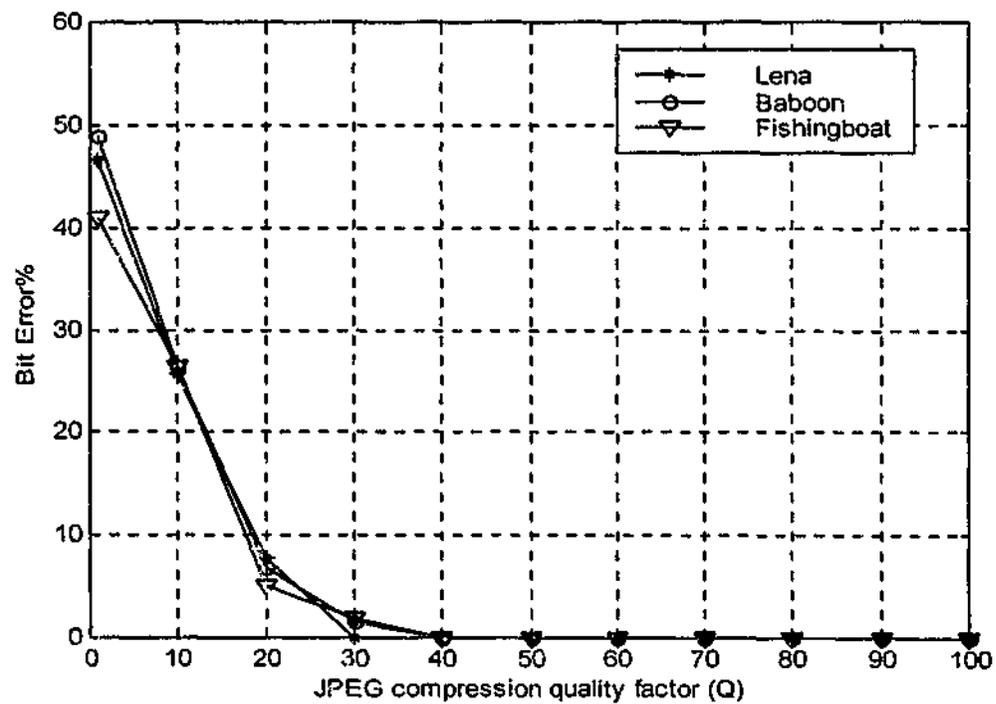


Figure 4.6 Bit error rate versus JPEG compression using text message as the signature.

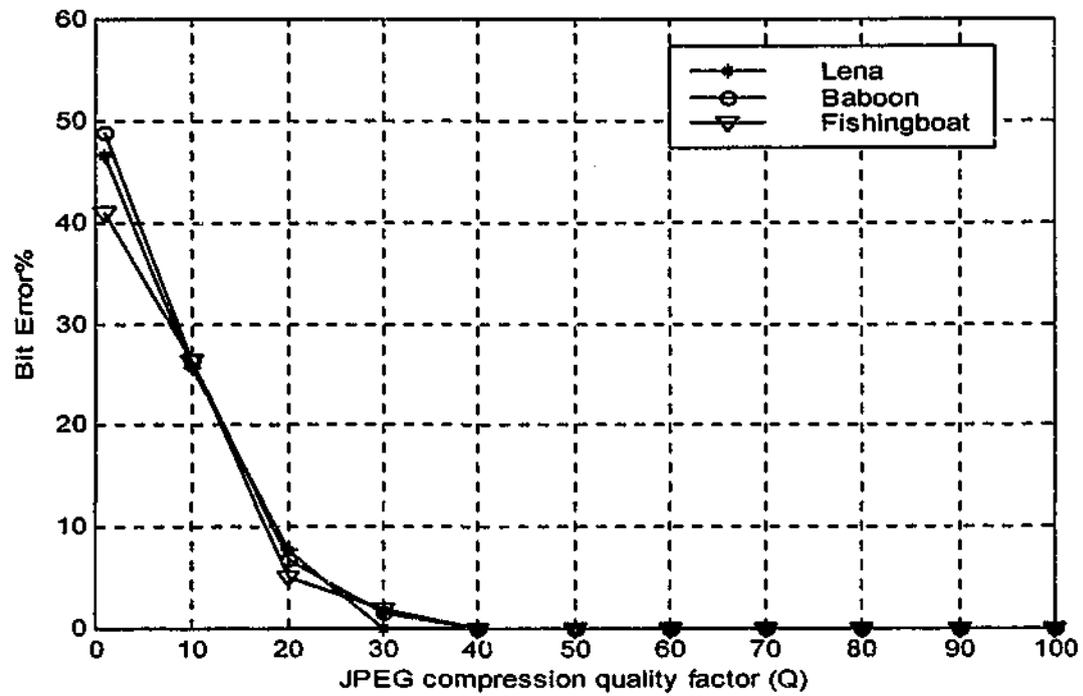


Figure 4.7 Bit error rate versus JPEG coding using text message as the signature.

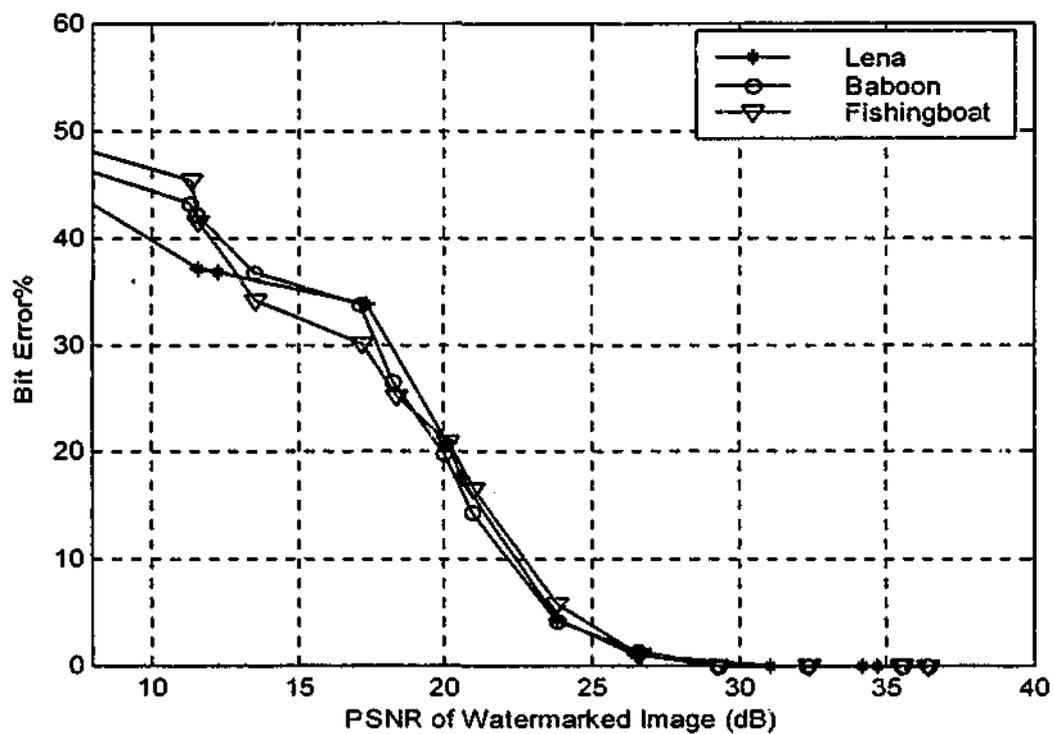
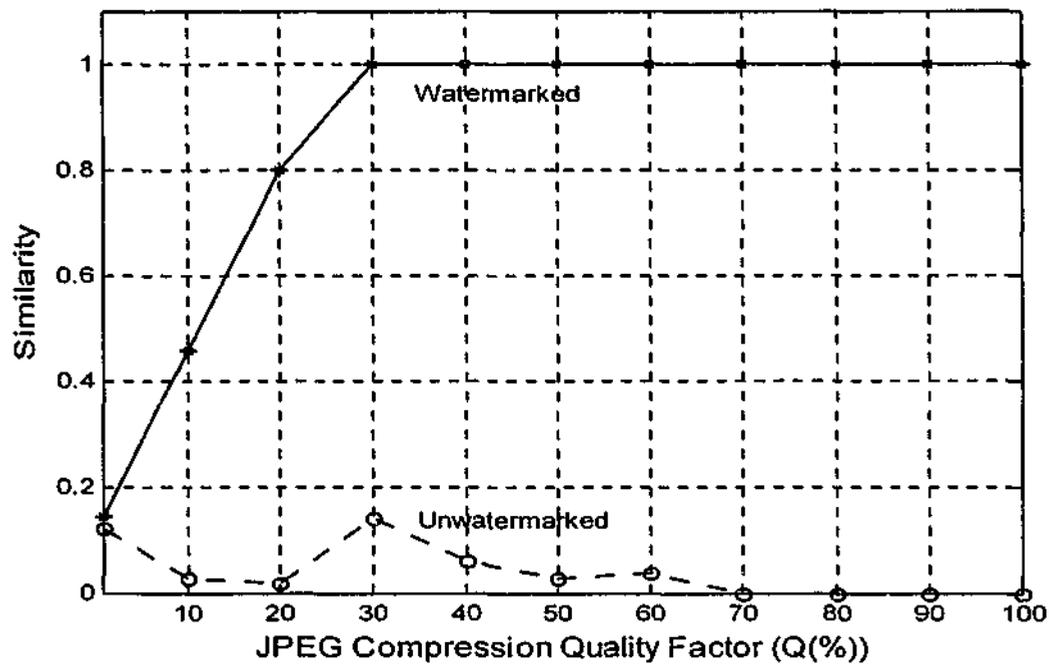


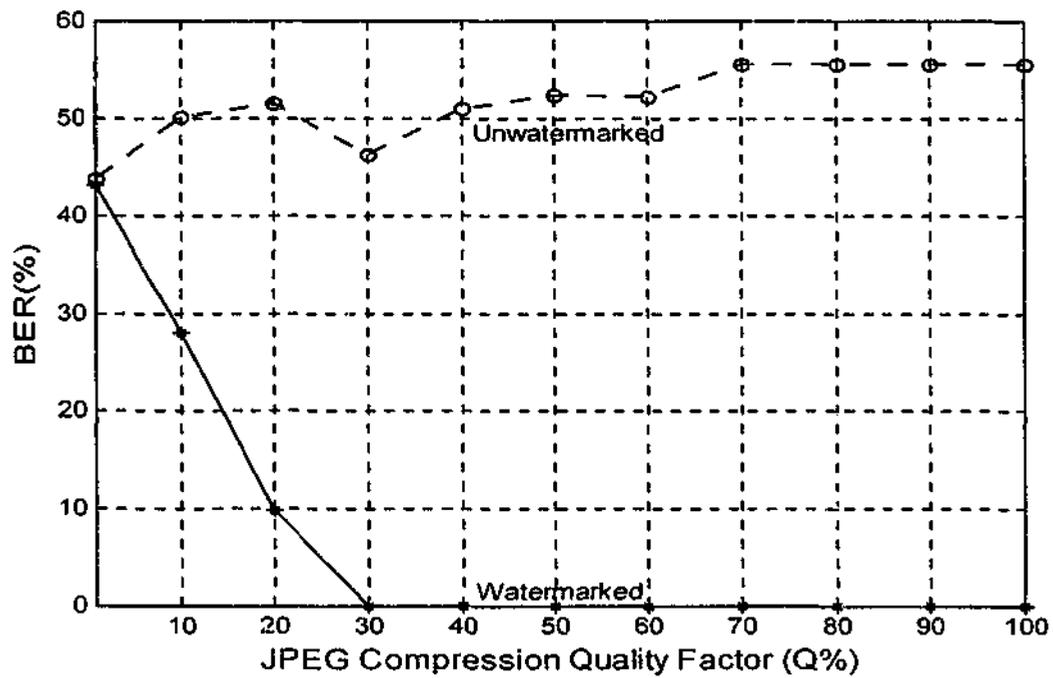
Figure 4.8 Bit error rate versus SNR using text message as the signature.

This similarity is simply the normalized correlation between  $s^*(m,n)$  and  $s(m,n)$ . The denominator of Equation (4.15) is 1 by definition. The correlation-based similarity measure gives a single number indicating the likelihood that the watermark or signature,  $s(m,n)$ , is present in the image. One can then set a threshold (based on a defined acceptable false positive probability) for comparison. If the similarity measure is higher than the threshold, then one can say that the watermark,  $s(m,n)$ , is present in the image, otherwise one declares that it is not. This procedure can be repeated for a number of different watermarks to determine which, if any, are present. It is worth noting here that the correlation-based similarity measure in Equation (4.15) only applies for zero-mean watermarks or signatures.

A graph of this similarity for varying JPEG compression is shown in Figure 4.9, as can be seen from this graph, it is easy to find a threshold for signature detection between unwatermarked and watermarked images. Moreover, Figure 4.10 shows the bit error rate for both watermarked and unwatermarked *Lena* image for the case of hiding text message. The solid curves in both figures are for the extraction of watermark from watermarked image while the dashed lines are for extraction from unwatermarked image. As a control experiment, the same extraction is performed on unwatermarked *Lena*, where there is no hidden signature. As expected, it yields a bit error of approximately 50%. The worst case of bit error rate is around 50% for



**Figure 4.9** Similarity measures for the extracted watermark from watermarked and unwatermarked images.



**Figure 4.10** Bit error rate for the extracted watermark from watermarked and unwatermarked images.

long sequences. However, for the case of limited number of bits as it is the case for text hiding, the bit error is slightly higher than 50% as shown in Figure 4.10.

#### 4.5.2 Embedding Images in Images

Using the algorithm of source and channel coding described in this Chapter, an image of  $\frac{1}{4}$  the size of the host image is used as the signature data. The embedding of *Bear* image of size 128 x 128 into the host image of *Lena* of size 256 x 256 is produced. The signature image is first compressed to  $\frac{1}{4}$  its original size using LBG vector quantizer with blocks of 2 x 2 and 16 levels. Then the indices obtained are channel coded employing BCH coding of (7,4). The coded indices are then multiplied by the scale factor  $\alpha$  before inserting them into the wavelet coefficients of *Lena* image. Different scale factors has been utilized to show the effect of this factor. Figure 4.11 shows *Lena* image watermarked with *Bear* image at various scale factor, without any compression. Note that the scale factor  $\alpha$  controls the relative weight of host and signature image contributions to the fused image. As the value of  $\alpha$  increases, the quality of the watermarked image degrades. For example, in Figure 4.11, one can see artifacts in the background for  $\alpha = 20$ .  $\alpha = 10$  appear to be a reasonable value in terms of the trade-off between quality of the watermarked image and robustness to signature recovery under image compression.

Figure 4.12 shows the PSNR of the recovered *Bear* image for different values of JPEG compression. At low compression values (i.e. higher JPEG quality factor), the quality of the recovered signature with a large scale factor  $\alpha$  is obviously much

better than those with a smaller  $\alpha$ . However, for high compression ratios, the effect of  $\alpha$  is minimal.

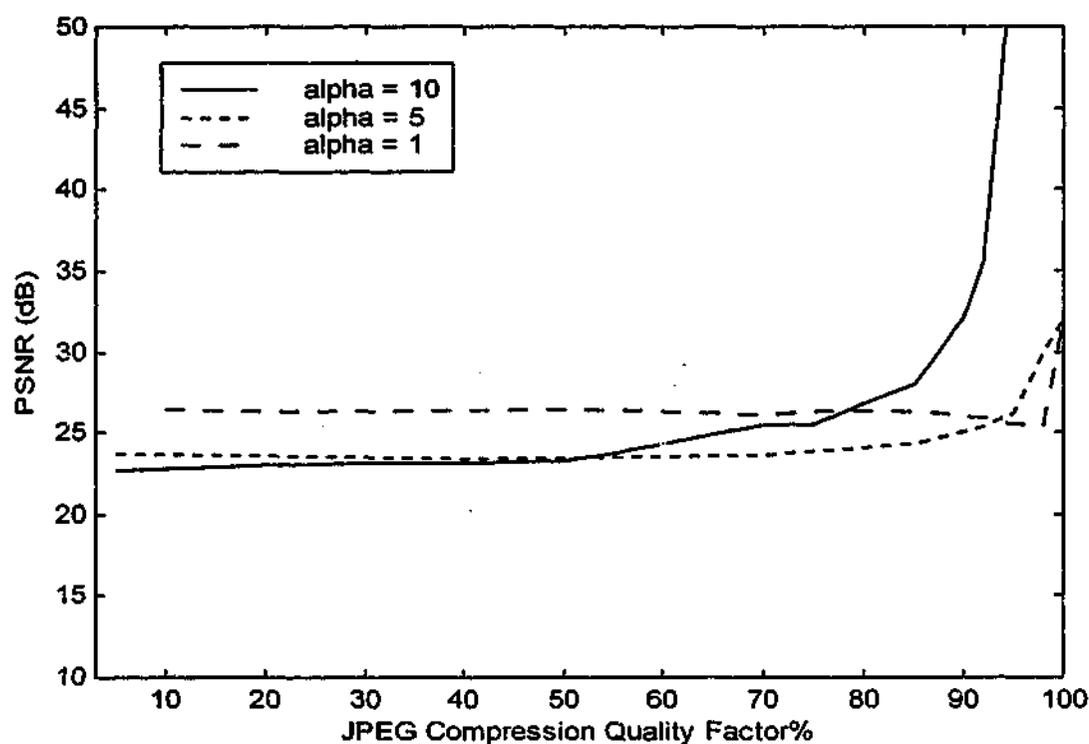
Figure 4.13 shows the compressed watermarked image of *Baboon* and the recovered *Peppers* image for different levels of JPEG compression. The recovered



Original

 $\alpha = 5$  $\alpha = 10$  $\alpha = 20$ 

Figure 4.11 Host *Lena* with embedded *Bear* image for various scale factors and BCH (4,7) code.



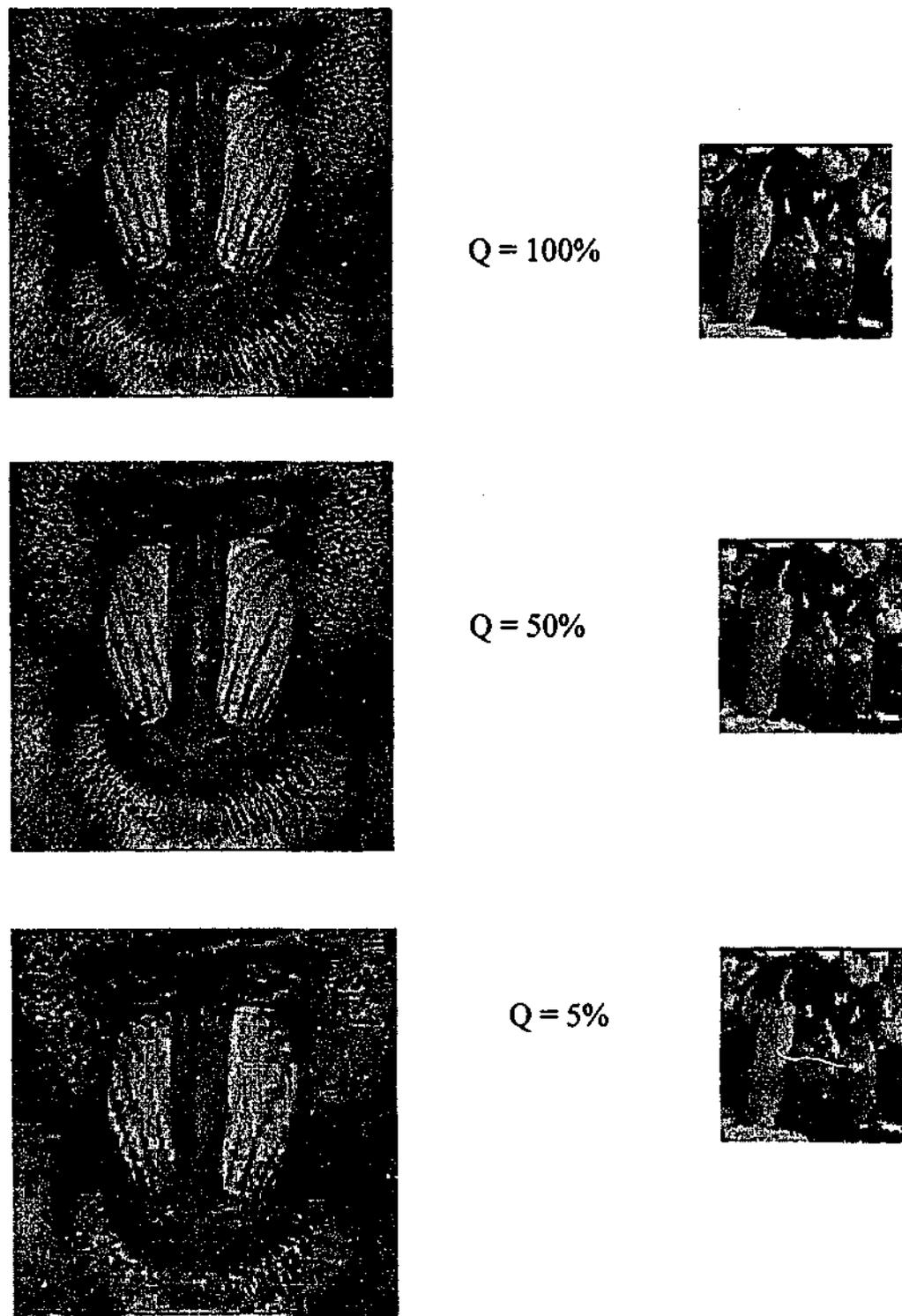
**Figure 4.12** The PSNR of the recovered *Bear* image for different scaling factor  $\alpha$  and different values of JPEG compression.

image at  $Q = 5\%$  is clear even though the watermarked image of *Baboon* has degraded to such an extent that it has no commercial value. The embedding was done using BCH(4,7) coding and  $2 \times 2$  vector quantizer for the signature data. The scale factor  $\alpha$  was equal to 10.

Figure 4.14 shows the signature images recovered from the watermarked *Lena* image after 100%, 80%, 50%, and 5% JPEG quality factor. In general, most of the recovered signature images are of high quality, the scale factor used was  $\alpha = 10$ .

In Figure 4.15 a plot of bit error (BER) for different levels of JPEG coding at different quality factors is shown for hiding *Bear* image into the three test images.

The algorithm works well under high quality coding conditions yet degrades more rapidly when the coding becomes too lossy.



**Figure 4.13** Embedded image using Baboon as the host image and the extracted Peppers image as the signature for different JPEG quality factors.

Similarly Figure 4.16 shows the plot of the PSNR of the recovered *Bear* image versus JPEG coding at different quality settings.

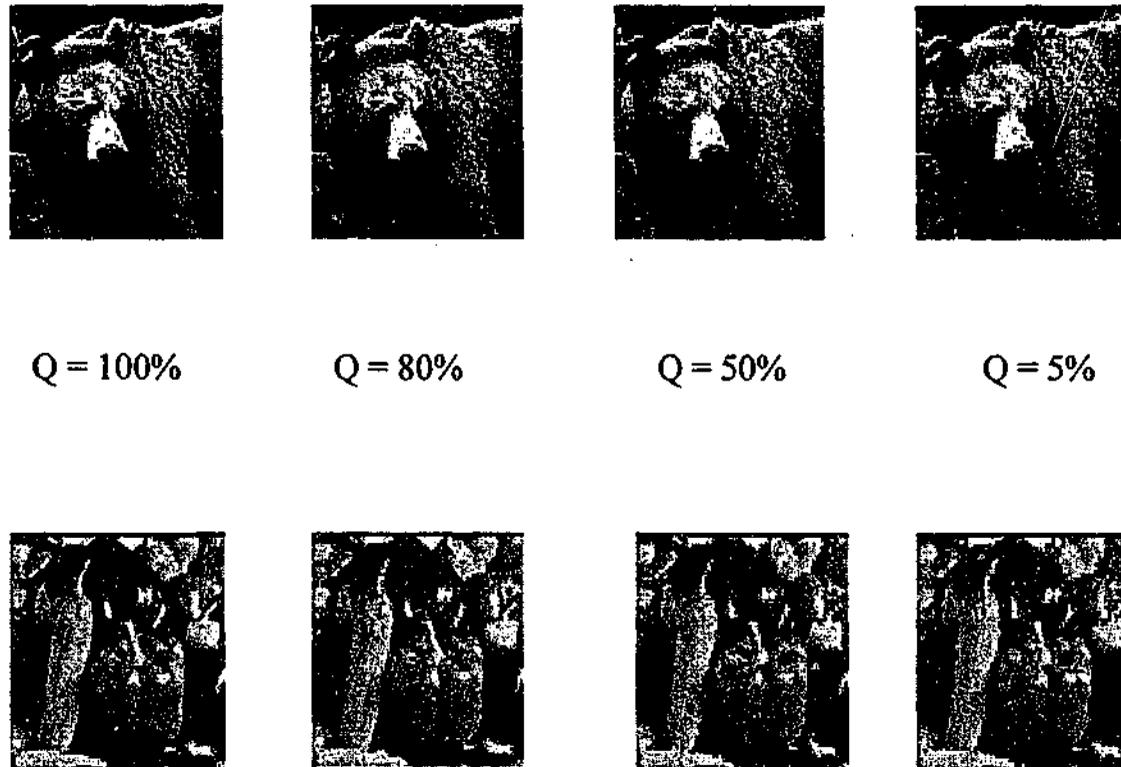
As one can see from these figures, the proposed method gives accurate results when the errors introduced in the image are small. The accuracy is reduced when the distortion grows larger. When there is no distortion, there is some errors introduced due to small errors which results from the inverse DWT and rounding to integers in the range from 0 to 255 for an 8 bit image.

### 4.5.3 Embedding Color Images

The embedded method described in this Chapter can be extended to embed data in color images. The color images are represented in the YIQ color space where the Y component is the luminance part of signal, and I and Q represent the chrominance components, see Figure 4.17 for YIQ color space representation of *Flower* image. Simulation studies had revealed that the color information is encoded more efficiently with the YIQ color space than for the RGB color space [Plataniotis and Venetsanopoulos, 2000]. Adopting the YIQ color space also facilitates a simple extension from images to digital video such as those in the MPEG format. Signature data is embedded only in the luminance component Y so as not to distort the color information. It is well known that the chrominance components occupy much less spectrum than the luminance components.

Figure 4.18 shows an example of color image embedding. Figure 4.18 (a) shows the original image of a 256 x 256 color image of *Lena* and Figure 4.18 (b) shows the watermarked image of *Lena* with a signature *Bear* image of size 128 x 128. The

entire signature data is inserted in the Y of the transform coefficients of the host *Lena* image. Figure 4.18 (c) shows the recovered watermark image of *Bear* from a



**Figure 4.14** Recovered signature images for different JPEG Quality factor (Q) using *Lena* image as the host and BCH(7,4) coding,  $\alpha = 10$ .

50% JPEG compressed watermarked image using BCH (4,7) and  $\alpha = 10$ . Note that there are no visible distortions in the watermarked image. Another example of color image embedding is shown in Figure 4.19 for *Flower* image.

The JPEG compression and the quality factor (Q%) used in this Chapter are obtained using Microsoft Photo Editor software version 3.0. The connection between the JPEG quality factor (Q%) and the compression ratio is shown in Table 4.1 together with the PSNR of the watermarked host image of *Lena* for *Bear* image

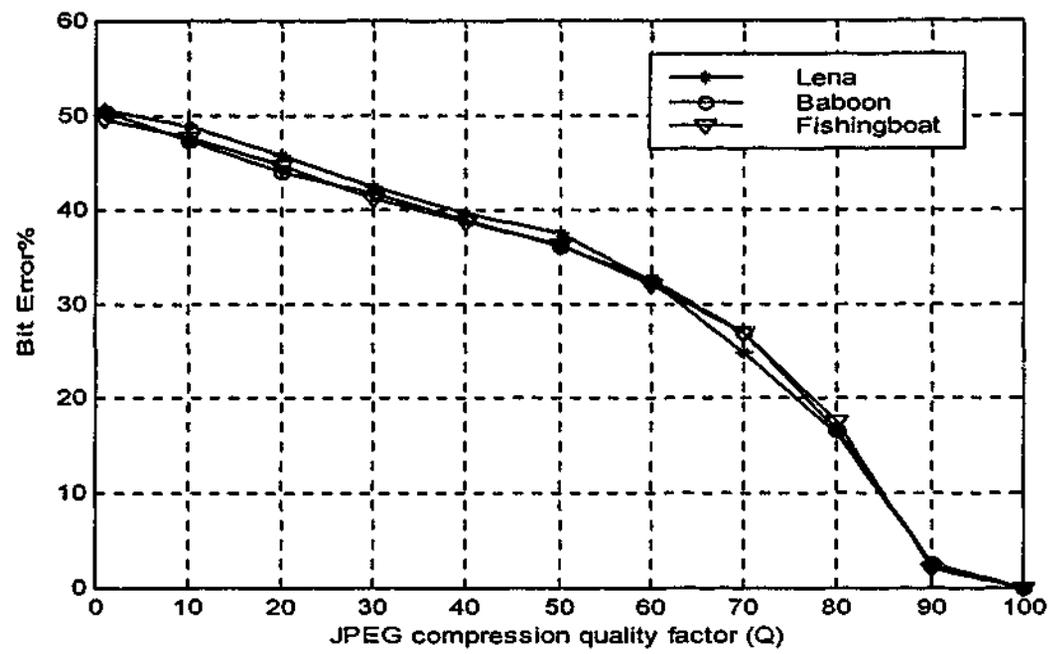


Figure 4.15 Bit error rate versus JPEG compression using *Bear* image as the signature.

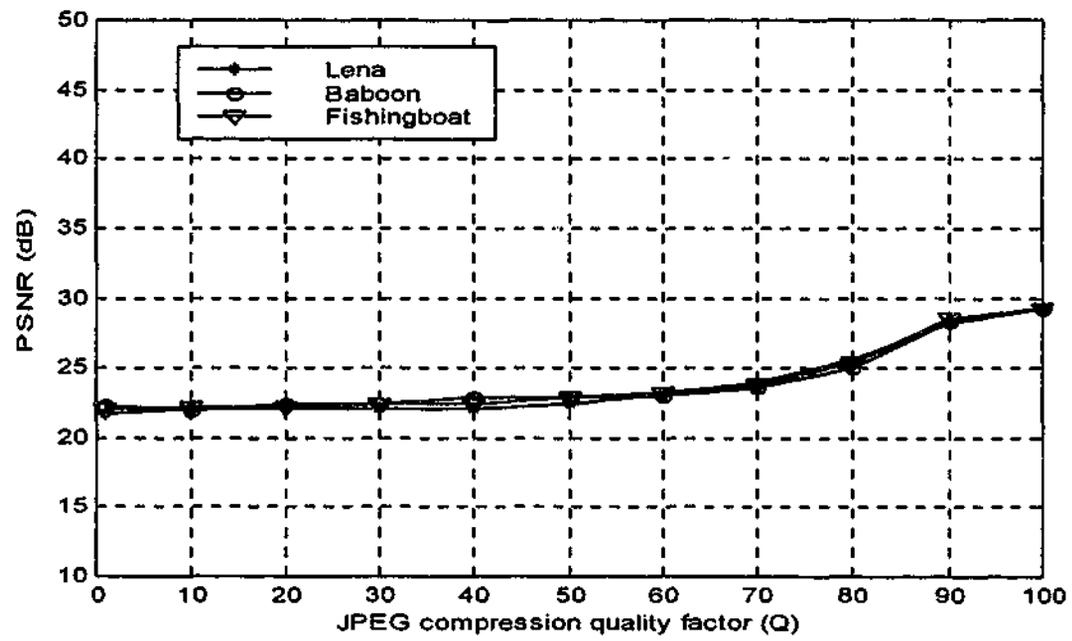


Figure 4.16 PSNR of recovered *Bear* image versus JPEG compression.

embedding and BCH coding. Table 4.2 shows similar parameters but for color host image embedding. For gray images the reasonable value for JPEG compression to obtain an accepted compressed image is at around 80% quality factor which gives a compression ratio of 5. For color images the affordable compression ratio is higher and the quality factor mostly used is around 50% which gives a compression ratio of 19. Matlab software [MATLAB] has been used in obtaining the results of this chapter and other chapters.

## 4.6 Summary

In this method, a scheme of embedding large amount of data in gray and colored images is presented. This approach could be used for both digital watermarking related applications as well as for data hiding purposes. The scale factor controls the relative amount of host and signature data in the embedded image. Experimental results demonstrate that good quality signature recovery and authentication is possible when the images are JPEG compressed and under noise addition.

The method performs very well against the JPEG compression attack and achieves 0% BER for quality factor as low as 30% as can be seen from Figure 4.10. This will be very important for the issue of lossless recovery that has been discussed earlier. There are relatively little differences in performance for different images indicating that the scheme is insensitive to image characteristics. Overall, the method performs better in JPEG compression attack than in noise addition.

The classic images used in the simulation (*Lena*, *Baboon*, and *Peppers*), were obtained from Image Database at the University of Southern California, Signal and Image processing Institute (USC-SIPI) [Image Database]. *Black Bear* Image and *Kid* images were obtained from Petitcolas web site [Petitcolas].

**Table 4.1** JPEG quality factor, compression ratio, and PSNR of watermarked gray *Lena* image.

JPEG Quality Factor (Q%)	Compression Ratio	PSNR of Watermarked <i>Lena</i> Image (dB)
100	1.50	30.54
90	3.50	30.19
80	5.00	29.89
70	6.00	29.23
60	7.33	28.86
50	8.25	29.10
40	9.45	29.18
30	11.00	29.07
20	13.20	28.43
10	16.50	26.94

**Table 4.2** JPEG quality factor, compression ratio, and PSNR of watermarked colored *Lena* image.

<b>JPEG Factor (Q%)</b>	<b>Quality</b>	<b>Compression Ratio</b>	<b>PSNR of Watermarked <i>Lena</i> Image (dB)</b>
100		2.92	29.50
90		7.72	28.91
80		11.35	28.57
70		13.78	28.00
60		16.08	27.66
50		19.30	27.80
40		21.44	27.88
30		27.57	27.48
20		32.16	26.76
10		48.25	25.05



Original



Luminance

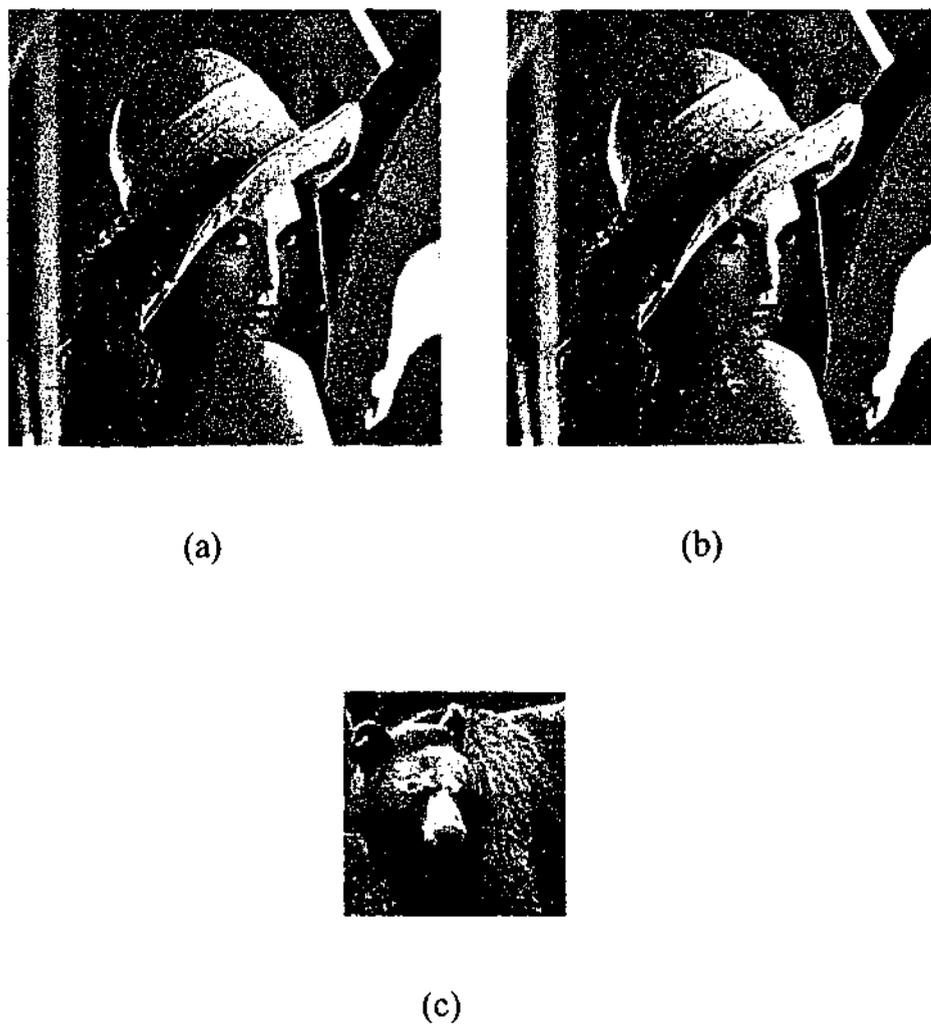


Hue



Saturation

Figure 4.17 YIQ color space representation of *Flower* image.



**Figure 4.18** (a) Original colored *Lena* image, (b) Watermarked *Lena* image with *Bear* image using BCH(4,7) and  $\alpha = 10$ , (c) recovered *Bear* image from 50% JPEG compressed watermarked image of *Lena*.



(a)



(b)



(c)

**Figure 4.19** (a) Original *Flower* image, (b) Watermarked *Flower* image with *Bear* image using BCH(4,7) and  $\alpha = 10$ , (c) recovered *Bear* image from 50% JPEG compressed watermarked image of *Flower*.

## **Chapter 5**

### **Data Embedding using Convolutional Coding**

#### **5.1 Need for Channel Coding in Data Embedding**

#### **5.2 General Embedding System**

#### **5.3 Convolutional Coding**

#### **5.4 Representation of Convolutional Codes**

#### **5.5 Distance Structure of a Convolutional Code**

#### **5.6 Decoding Techniques of Convolutional Codes**

#### **5.7 Evaluating Error probability using the Transfer Function**

#### **Bound**

#### **5.8 Concatenated Codes**

#### **5.9 Modulation Codes (Trellis Codes)**

#### **5.10 Turbo Codes**

#### **5.11 Implementation and Experimental Results**

#### **5.12 Summary**

## Data Embedding using Convolutional Coding

---

As has been pointed out earlier in Chapter 3, data hiding can be thought of as a special communication problem, where signature data are the information to be sent from sender to receiver through special channel. The channel is composed of the host signal (image or video) and the noise introduced by signal processing and/or attacks. Imperceptibility, robustness against moderate compression and processing, and the ability to hide many bits are the basic but rather contradictory requirements for many data hiding applications. The traditional way to handle this is to target at a specific capacity-robustness pair. Some approaches choose to robustly embed just one or a few bits [Barni, et. al., 1998a], [Cox et. al, 1996], [Langelaar, et. al., 1997], while others choose to embed a lot of bits but to tolerate little or no distortion [StegoDos]. More recently, it has been pointed out in the literature that similarities between the data hiding problem and digital communication can be utilized to improve the performance of the system [Tewfik, 2000].

In the previous chapter we investigated the use of block codes to improve the embedding algorithm. In this chapter we investigate the performance of the algorithm with the use of convolutional coding together with concatenated codes

and turbo coding. Maximum likelihood decoding using Viterbi algorithm and turbo decoding are employed in extracting the hidden information.

Some of the results of this Chapter were published in [Abdulaziz and Pang, 2001].

## 5.1 Need for Channel Coding in Data Embedding

Shannon has shown that it is possible to transmit digital information reliably over a channel of given capacity provided that the bit-rate does not exceed the channel capacity [Shannon, 1948]. Digital watermarking has been considered as an application of digital communication theory in [Ramkumar and Akansu, 1998], [Cox, et. al., 1999]. The host image constitutes the channel for transmission of the watermark data and is subject to various types of attacks. Note that attacks such as lossy compression, enhancements, or transformations can be treated as noise addition. Lossy compression algorithms such as JPEG might severely narrow the channel by totally disregarding large regions of the frequency spectrum of the image. However, using convolutional coding techniques can greatly improve the robustness of the embedded data against compression and standard digital image processing operations.

The use of channel coding in data hiding was proposed independently in [Hernandez et. al, 1998a] and [Marvel, et. al., 1998], where block codes have been implemented in the watermarking process for the spatial domain. Another scheme has been published in [Chae, et. al., 1998], employing lattice codes as the channel codes, such embedding suffers from the increased difficulties in implementation as the size of lattice codes increases [Conway and Sloane, 1993]. Our scheme is

different from the above mentioned, firstly, the embedding is implemented in the wavelet domain and not in the spatial domain, secondly, we propose to use new schemes adapted from the area of deep-space communications such as concatenated coding and turbo codes. In addition, it is well known that for a given probability of error, the use of channel coding results in a coding gain between 3-5 dB over the uncoded case for simple codes [Biglieri, et. al., 1991]. Moreover, there is a great similarity between the data hiding system and deep-space communications. In the latter, the received signal power is usually weak at the earth station, noise is additive white Gaussian, and the errors are random in nature. Therefore, a large error-correcting capability is needed. Because the bandwidth is not restricted, it is possible to build a complex channel decoder. Hence, low-rate, powerful, error-correcting codes are often used. To improve the system performance, concatenated coding is also implemented. Examples of error-correcting codes can be seen in the design of *Pioneer 9 solar* orbit mission, *Voyager* spacecraft, and *Galileo* spacecraft, where convolutional and concatenated codes were implemented for the missions. For *Pioneer 9* the convolution scheme with additive white Gaussian noise can achieve 5 dB coding gain over uncoded data at an error rate of  $10^{-5}$ , while the concatenated code designed for *Voyager* provided an extra 2dB of coding gain over *Pioneer 9*. However the development in the design of the concatenated code scheme for *Galileo* resulted in an improvement of 2 dB over voyager, resulting in a total of 9 dB coding gain over uncoded schemes [Charles-Lee, 1997].

In watermarking, the weak signal power in deep space communication is similar to the fidelity constraint of the watermarked media content. In other words, for the watermark to be invisible, the magnitude of the watermark signal must be very small compared to the host signal. However, in data hiding, the large bandwidth available for space communication is replaced by a fixed bandwidth that depends on the size of the host signal under application. While this represents a deviation from deep space communications, the resulting bandwidth constraint can be minimized by implementing some kind of compression to the watermark signal before embedding.

Throughout the chapter, we will use interchangeably the terms data extraction/decoding and data embedding/transmission.

In the next sections a detailed encounter of data embedding and extraction using convolutional coding will be provided.

## 5.2 General Embedding System

The model we are following for data embedding is represented in Figure 5.1. The original host image is first transformed to the wavelet domain. Suppose that we want to embed  $N$  bits of information. The  $N$  bits of information could represent a compressed version of the signature data where VQ has been implemented on the data. Now let the information sequence arranged in a vector of size  $n$  be hidden in the elements of the host. The embedding could be direct or it could depend on a secret key that is known only to copyright owner and to authorized recipient. The information symbols are added to the host coefficients after scaling by a factor  $\alpha$

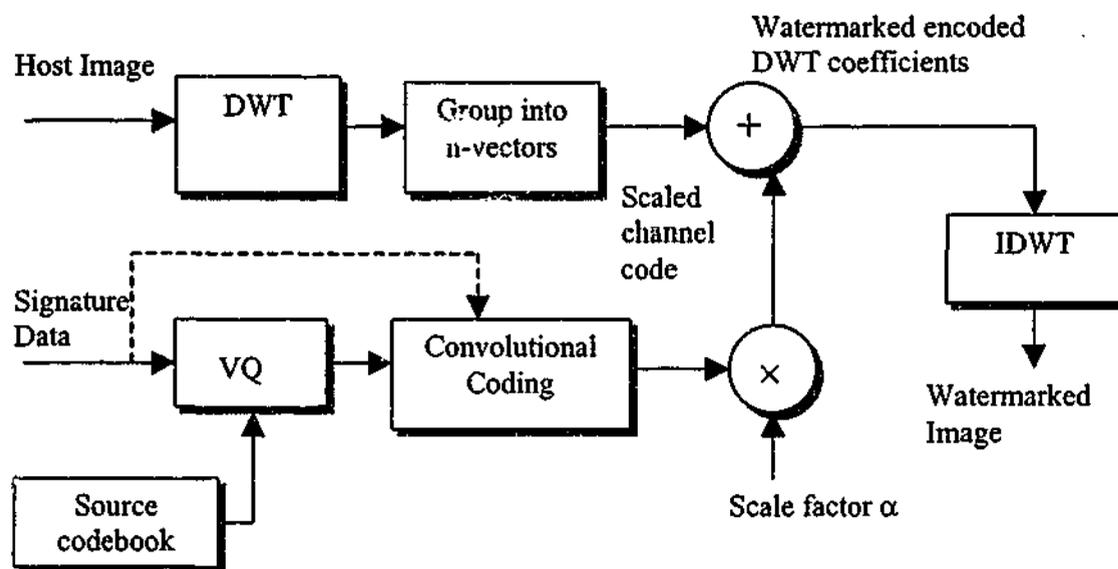


Figure 5.1 General block diagram of the encoder.

to produce a watermarked image that conceals the secret information. At the recipient side, the objective will be to extract this information with the highest possible fidelity. In order to guarantee invisibility, the watermark sequence bits could be multiplied by a perceptual mask and not a constant scale factor  $\alpha$ . The perceptual mask can be obtained after analyzing the original image with a psychovisual model, which takes into account the different sensitivity of the human visual system (HVS) to alterations in the different elements of the host. The perceptual mask for an image is different depending on the domain chosen for embedding and on the specific properties of the HVS that are being taken into account. The details on how  $\alpha$  is evaluated in different domains can be found in [Girod, 1988], and [Hernandez et. al., 1998b]. We simply mention that the use of the coding and the methodology of embedding proposed here can be readily

extended to other data embedding schemes. This is an important advantage of providing a general framework.

Once we have built a basic scheme for reliable information embedding, its performance can be improved by means of coding. Chapter 4 has dealt with block codes, here we will proceed to show how convolutional codes can be used for data embedding purposes.

Suppose that, instead of transmitting raw information symbols through the hidden channel, we use a  $(n, k)$  block code that maps  $k$  information symbols into  $n$  binary channel symbols  $s(i)$ , with  $i = 1, \dots, n$ . In order to use this code for data embedding, the set of  $N$  source information bits is divided into  $N/k$  blocks, and each block of size  $k$  bits mapped into  $n$  bits that are hidden using a procedure for watermark insertion similar to that summarized in section 4.1.

Regarding watermark extraction, the schematic block diagram of the decoder is shown in Figure 5.2. Soft and hard-decision decoding implementing the maximum likelihood (ML) decoding were used.

Below is a review of convolutional coding technique and how it is implemented in our algorithm.

### 5.3 Convolutional Coding

A convolutional coder is a finite memory system (rather than a memoryless system, as in the case of block coder). The name refers to the fact that the added redundant bits are generated by modulo-two convolutions.

Convolutional codes are more commonly used than block codes, primarily because they are conceptually and practically simpler, and their performance matches (and

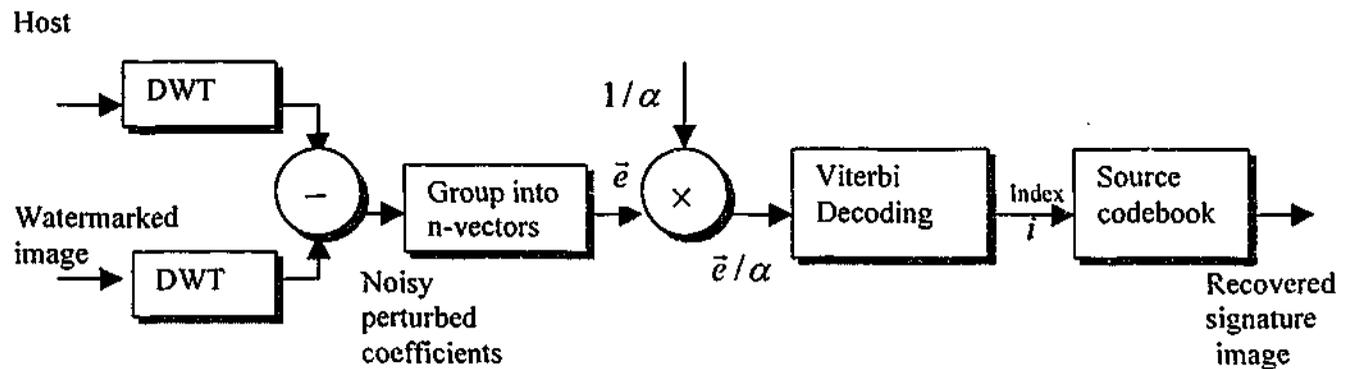


Figure 5.2 General block diagram of the decoder.

often exceeds) that of a good block code. Their good performance is attributed in part to the availability of practical soft decoding techniques [Bateman, 1999].

The main advantage of convolutional codes is their error correction power together with the availability of efficient algorithms that perform soft decoding. Convolutional codes are advantageous over block codes for similar rates. Detailed performance and properties of convolutional codes can be found in [Lin and Costello, 1984], [Biglieri et al. 1991], and [Charles-Lee, 1997].

Implementation of convolutional codes for data embedding follows the same lines for block codes but with soft-decision decoding in addition to hard-decision decoding. Given a rate  $R = k/n$  convolutional code, the  $N$  information bits are divided in groups of  $N/k$  blocks that are sequentially introduced in the convolutional encoder. The latter evolves through its state diagram and produces an output in groups of  $n$  bits, thus resulting a total of  $M = (N n)/k$  symbols, that are transmitted through the hidden channel exactly as was described in the previous chapter. The values for  $k$  and  $n$  are usually small (in the order of few bits). The

output bits depends not only on current set of  $k$  input bits, but also on past input. The number of bits which the output bits depend on is called "constraint length"  $K$ . Generally, the constraint length for linear convolutional codes is equal to  $m+1$ , where  $m$  is the number of shift registers or memory elements that affect the output bits. Example of a convolutional code with  $k = 1$ ,  $n = 2$ , and  $K = 3$ , is shown in Figure 5.3.

There are some differences between block codes and convolutional codes. Block codes such as BCH codes and Reed-Solomon codes breaks a message stream up into fixed size blocks and add redundancy symbols to offer error correction capability. They are usually decoded via algebraic methods. However, binary convolutional codes take a different approach to error control coding. The message stream is not broken into blocks with redundancy added to each block independently. Instead, redundancy is added continuously and is dependent on past bits of the message. This converts the entire message stream into one long codeword. One further difference between block codes and convolutional codes is that convolutional codes are decoded by an analytical approach rather than algebraically.

Just as for block codes, linear convolutional codes are constructed using modulo-two adders with the addition of delay elements. Convolutional coders can be described using a generator matrix where instead of the entries being zero or one, the entries are polynomials in  $D$  with coefficients that are either zero or one.  $D$  is used in place of  $z^{-1}$  and it is defined as the unit delay.

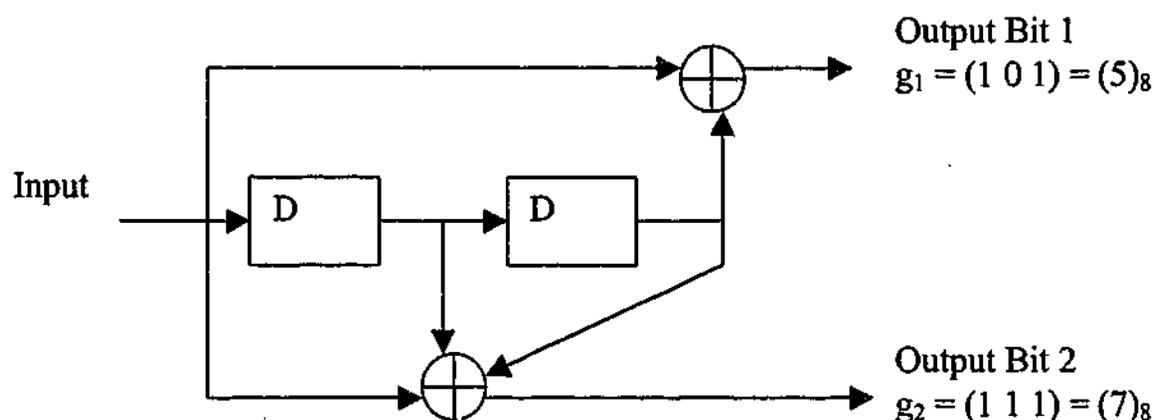


Figure 5.3 Block diagram representation of (2,1,2) convolutional encoder.

## 5.4 Representation of Convolutional Codes

An  $(n, k, m)$  binary convolutional encoder over the Galois field  $GF(2)$  is a  $k$ -input,  $n$ -output, time-invariant, causal, finite-state-machine of encoder memory order  $m$  [Viterbi, 1979]. The code rate is defined by  $k/n$ .

Convolutional codes can be described in many ways; e.g. block diagram representation, generator polynomials, state diagram, or a trellis diagram.

### 5.4.1 Encoder Block Diagram

In the convolutional encoder, message bits are fed into a shift register. The set of delay elements in the shift register can be thought of as a state machine. Codeword bits are generated as functions of the current state and input. These functions are sums of fixed patterns of taps into the shift register. What is referred to, as a codeword here is not the same as in block coding. In block coding, each coding is independent of all other blocks, this is not so for convolutional codes. Each

codeword is dependent on the current message word and the state of the register which stores information about the past values of the message bits. Therefore successive codewords are not independent of one another. An example of an encoder circuit is shown in Figure 5.3 for a rate  $\frac{1}{2}$  code. The codewords in convolutional codes are the collection of code bits that form the output of the encoder circuit for one period of the shift register. Likewise a message word is the collection of bits inputted to the encoder circuit during one period of the shift register. Message words can be any length (greater than 1) although they are usually just one bit long. The codeword length should be greater than the message length otherwise no redundancy is added and the code has no error correcting capability. The convolutional code analog of the minimum distance of a block code is the free distance, which is defined as the number of state changes that give a minimum weight codeword sequence.

The encoder block diagram is shown in Figure 5.3. Where the delay elements are represented using shift registers, and output of shift registers are connected in a certain pattern to multi-input modulo-adders. In  $GF(2)$ , the modulo-2 adders can be implemented as exclusive-or logic gates.

#### 5.4.2 Generator Representation

In this representation, one generator vector for each of the  $n$  output bits is given. The bits in the generator from left to right represent the connections in the encoder circuit. Where a "1" represents a link from the shift register and a "0" represents no link. Usually the encoder vectors are often given in octal representation. For

example, from Figure 5.3, the generator vector  $g_1 = (101)$ , and  $g_2 = (111)$ , or in octal form,  $g_1 = (5)_8$ , and  $g_2 = (7)_8$ .

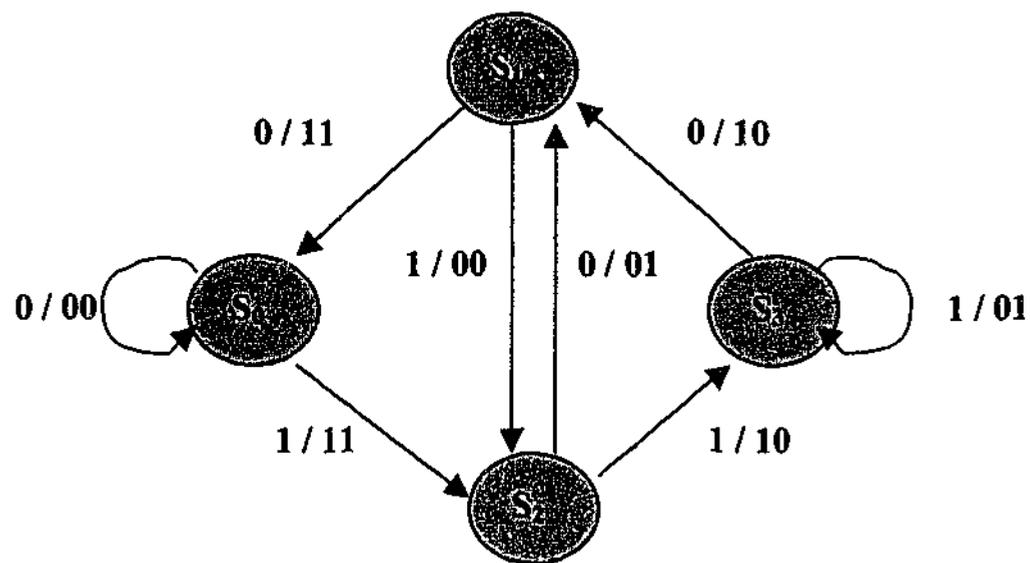
### 5.4.3 State Diagram Representation of Convolutional Codes

As previously mentioned, the encoder can be represented as a finite state machine with outputs as functions of the current state and input [Proakis, 1995].

The “state” of the code is represented by the shift register contents, where, most recent input is most significant bit of state and oldest input is least significant bit of state, although this convention sometimes reversed in some books. The arcs connecting states represent allowable transitions and are labeled with input/output bits transmitted during transition. Example of state diagram for the convolutional encoder of Figure 5.3 is shown in Figure 5.4, where  $S_0$  represents the state 00,  $S_1$  represents the state 01,  $S_2$  represents the state 10, and  $S_3$  represents the state 11.

### 5.4.4 Trellis Representation of Convolutional Codes

A code trellis is a graphical representation of a code, block or convolutional, in which every path represents a codeword (or a code sequence for a convolutional code). The trellis representation is an extension of the state machine that shows the passage of time. This representation makes it possible to implement maximum likelihood decoding (MLD) of a code with reduced decoding complexity. It is an important part of the Viterbi decoding algorithm. In this representation, the state diagram is “unfolded” as a function of time. The time is indicated by movement towards right. The contents of shift registers make up “state” of code, where as before, most recent input is most significant bit of state, and oldest input is least



**Figure 5.4** State diagram representation of the convolutional code (2,1,2) shown in Figure 5.3.

significant bit of state. Allowable transitions are denoted by connections between states. Example of trellis diagram is shown in Figure 5.5. Figure 5.6, shows an example of using trellis representation which will be explained in Section 5.6.1.

The trellis representation was first introduced and used for convolutional codes. This representation, together with the Viterbi decoding algorithm, has resulted in a wide range of applications of convolutional codes for error control in digital communications over the last two decades. [Charles-Lee, 1997].

## 5.5 Distance Structure of a Convolutional Code

The performance of a convolutional code depends not only on the decoding algorithm used but also on the distance properties of the code. In this context, the most important single measure of a convolutional code's ability to resist channel

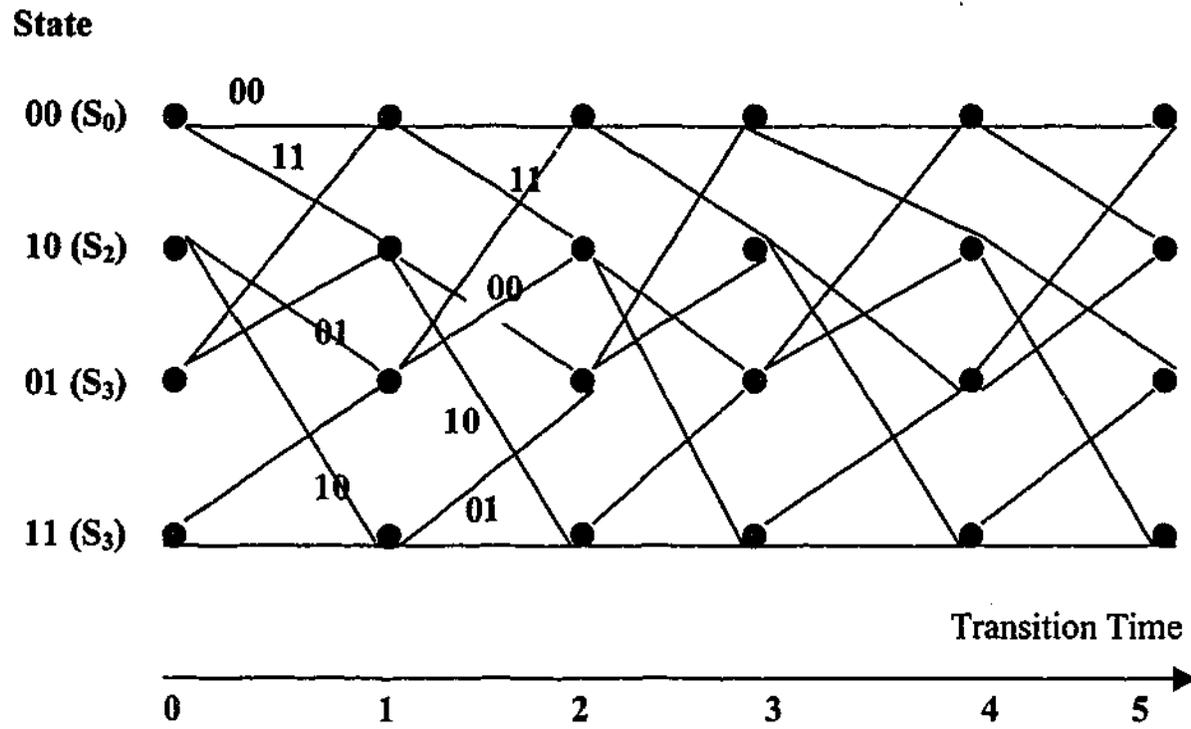


Figure 5.5 Trellis diagram for convolutional encoder with 4 states.

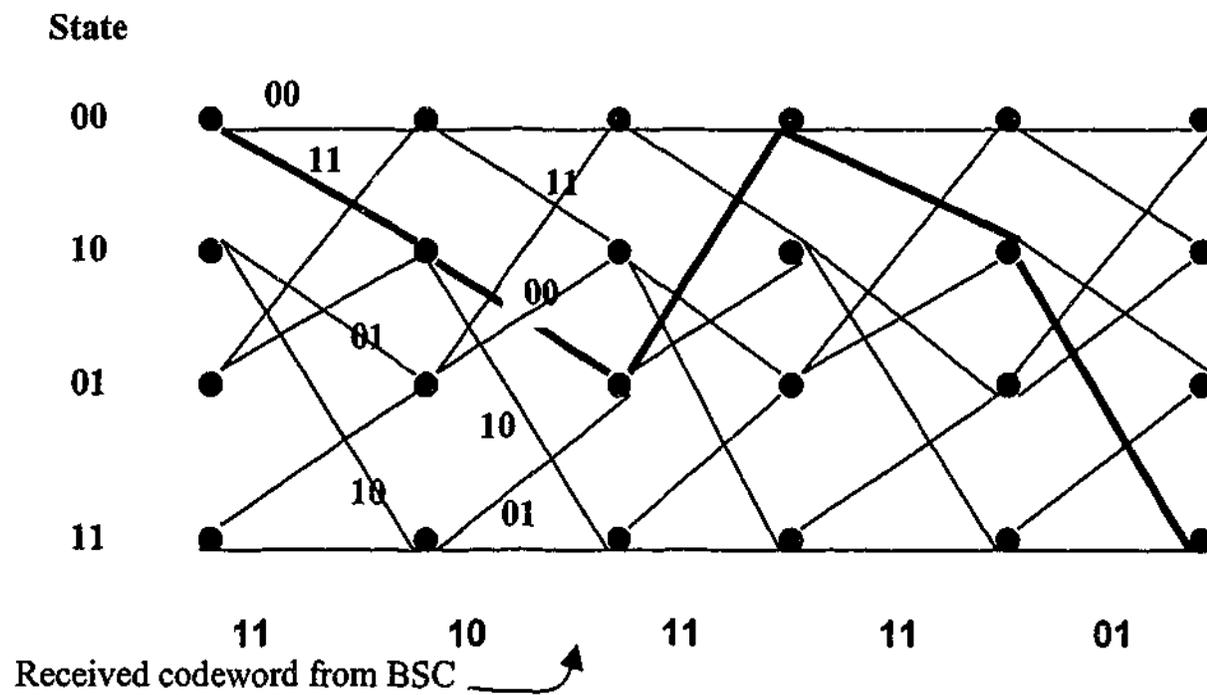


Figure 5.6 Encoding example using trellis representation.

noise is the free distance denoted by  $d_{free}$ . The free distance of a convolutional code is defined as the minimum Hamming distance between any two code words in the code. A convolutional code with free distance  $d_{free}$  can correct  $t$  errors if and only if  $d_{free}$  is greater than  $2t$ .

The state diagram of the convolutional code can be modified to provide a complete description of the weight distribution of the code. Consider for example, Figure 5.4 that shows the state diagram of the (2,1,2) convolutional encoder of Figure 5.3. The modified state diagram to this encoder can be obtained by splitting state  $S_0$  into an initial state,  $S_{in}$ , and a final state,  $S_{out}$ , with the self-loop around  $S_0$  being deleted. Let  $X$  be the indeterminate (variable) associated with the Hamming weight of the information vector  $\mathbf{X}$ ,  $Y$  is the indeterminate associated with the Hamming weight of the encoded output vector  $\mathbf{Y}$ , and  $L$  the indeterminate associated with every branch. Each path in the modified state diagram has a gain, labeled as  $X^i Y^j L^l$ , where  $i$  is the hamming weight of the encoder input row vector  $\mathbf{X}$  (the hamming weight of a binary word is defined as the number of 1s contained in the word),  $j$  is the Hamming weight of the encoder output row vector  $\mathbf{Y}$ , and  $l$  is the number of branches between two states. The modified state diagram expressed in terms of  $X$ ,  $Y$ , and  $L$  is called an augmented state diagram [Biglieri et. al., 1991]. Considering the (2,1,2) convolutional encoder in Figure 5.3 and its state diagram in Figure 5.4. The modified and the augmented state diagrams of the encoder are shown in Figures 5.7 and 5.8, respectively.

For the augmented state diagram, the transfer function that provides the weight distribution is given by

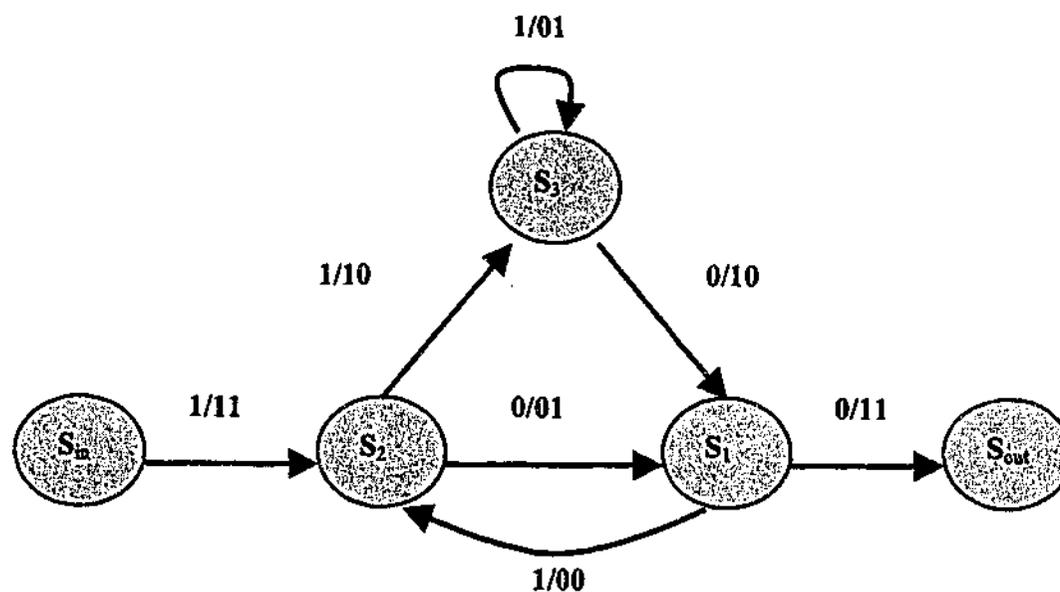


Figure 5.7 Modified state diagram of Figure 5.4.

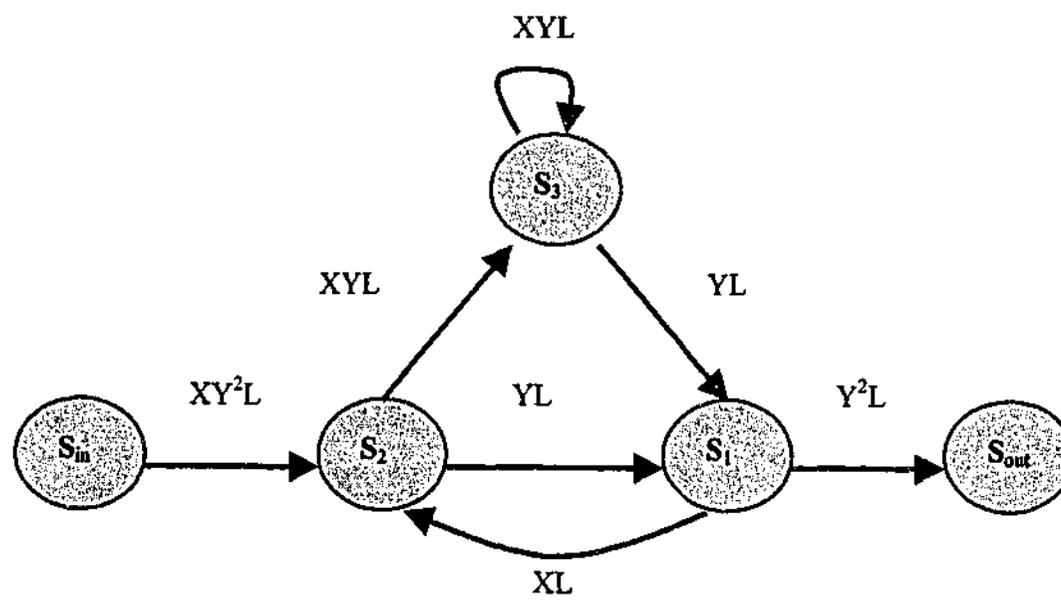


Figure 5.8 Augmented state diagram of Figure 5.4.

$$T(X, Y, L) = \sum A_{i,j,l} X^i Y^j L^l \quad (5.1)$$

The coefficient  $A_{i,j,l}$  denotes the number of  $n$  encoder output bits with Hamming weight  $j$ , whose associated  $k$  input information bits have weight  $i$  and branch length  $l$ . The modified state diagram may be viewed as a signal-flow graph with a single input and a single output and Mason's rule [Mason and Zimmermann, 1960] can be applied to get the transfer function  $T(X, Y, L)$ . Alternatively, the transition behavior in the augmented state diagram can be described by a set of equations and those equations are solved to obtain  $T(X, Y, L)$ . Considering again the augmented state diagram of the (2,1,2) convolutional encoder shown in Figure 5.8. The set of equations obtained are:

$$S_2 = XLS_1 + XY^2LS_m \quad (5.2)$$

$$S_1 = YLS_2 + YLS_3 \quad (5.3)$$

$$S_3 = XYLS_2 + XYLS_3 \quad (5.4)$$

$$S_{out} = Y^2LS_1 \quad (5.5)$$

From Equation (5.4),

$$S_3 = XYLS_2 / (1 - XYL) \quad (5.6)$$

Substituting Equation (5.6) into Equation (5.3), we get

$$S_1 = YLS_2 / (1 - XYL) \quad (5.7)$$

Substituting equation (5.7) into Equation (5.2), we get

$$S_2 = (1 - XYL)XY^2LS_m / (1 - XYL - XYL^2) \quad (5.8)$$

Substituting equation (5.7) into Equation (5.5), we get

$$S_{out} = Y^3 L^2 S_2 / (1 - XYL) \quad (5.9)$$

Substituting Equation (5.8) into Equation (5.9), we obtain the transfer function

$$S_{out} / S_{in} = XY^5 L^3 / \{1 - XYL(1 + L)\} \quad (5.10)$$

Using the binomial expansion, we may equivalently write

$$S_{out} / S_{in} = XY^5 L^3 + X^2 Y^6 L^4 + (X^2 Y^6 + X^3 Y^7) L^5 + (2X^3 Y^7 + X^4 Y^8) L^6 + \dots \quad (5.11)$$

$$S_{out} / S_{in} = T(X, Y, L) \quad (5.12)$$

An information sequence of weight 1 and the codeword of weight 5 have length-3 branches, one information sequence of weight 2 and codeword of weight 6 have length-4 branches, another information of weight 2 and codeword of weight 6 have length-5 branches, and so on. The length of each branch is one. For the example of Figure 5.8, we may use Equations (5.11) and (5.12) to obtain the following input-output relations that represent the transfer function of the graph

$$T(X, Y, L) \Big|_{L=1} = XY^5 + 2X^2 Y^6 + 4X^3 Y^7 + 8X^4 Y^8 + \dots \quad (5.13)$$

the total number of codewords paths of Hamming weight  $j$  can be found by setting  $X = 1$  in the transfer function in Equation (5.13), that is,

$$T(X, Y, L) \Big|_{X=1, L=1} = T(Y) = Y^5 + 2Y^6 + 4Y^7 + 8Y^8 + \dots \quad (5.14)$$

Furthermore, the total number of nonzero information bits on all paths of hamming weight  $j$  is given by the partial derivative of  $T(X, Y, L)$  with respect to  $X$ , where

$$\frac{\partial T(X, Y, L)}{\partial X} \Big|_{X=1, L=1} = Y^5 + 4Y^6 + 12 Y^7 + 32Y^8 + \dots \quad (5.15)$$

Since the free distance is the minimum Hamming distance between any two code words in the code and the distance transfer function  $T(Y)$  enumerates the number of

code words that are a given distance apart, it follows that the exponent of the first term in the expansion of  $T(Y)$  defines the free distance. Thus the convolutional code of Figure 5.3 has a free distance of  $d_{free} = 5$ .

## 5.6 Decoding Techniques of Convolutional Codes

Three main decoding techniques are available for convolutional codes: sequential decoding, Viterbi decoding, and majority logic decoding. Sequential decoding involves making trial hypotheses about the data and performs a tree search, abandoning a branch when its hypothesis disagrees too much from what is received. It is an effective technique for use with long constraint length codes. Viterbi decoding is a technique that has gained a broad application, it has the advantage that decoding complexity is deterministic. It also works with soft decisions, and is optimum in some sense for the code. Its disadvantage is that it is limited to short constraint length codes because memory goes up exponentially with constraint length. Majority logic decoding for convolutional codes is a direct extension of the technique for block codes. They are very simple, but are generally designed to work only after hard decision [Biglieri, 1994]. In this work, only Viterbi decoding has been implemented in the decoding of convolutional codes.

### 5.6.1 The Viterbi Algorithm

The Viterbi algorithm is a clever way of implementing maximum Likelihood decoding. It can be used for either hard or soft decision decoding. We consider hard decision decoding initially. In this case the algorithm chooses the code sequence

through the trellis which has the smallest Hamming distance to the received sequence.

Implementation of the Viterbi algorithm may be described in general terms as follow [Schlegel, 1997]:

- For the rate  $\frac{1}{2}$  convolutional encoder presented in Figure 5.3, the first step is to draw the code trellis as shown in Figure 5.5. Note that it is simply another way of drawing the state diagram, which is presented on Figure 5.4.
- The four possible states (00, 01, 10, 11) are labeled  $S_0, S_1, S_2, S_3$  as shown in Figure 5.5.
- Since the input bits  $k = 1$ , there are two branches emerging from each state, the upper branch represents an input of 0 and the lower branch an input of 1.
- The branch codeword is the output codeword associated with a branch. e.g. the lower branch emerging from state 00 and entering state 10 ( $S_2$ ) has the branch codeword 11. It is labeled 1/11 in Figure 5.4 which means that a binary digit 1 input to the encoder in state 00, will output the codeword 11 and move to the state 10.
- Using the code trellis, the *Viterbi Trellis* is drawn as a serial concatenation of many code trellis diagrams as in Figure 5.5.
- The *trellis depth* of a Viterbi trellis is the number of code trellis replications used. e.g. the trellis depth is 5 in Figure 5.5.
- The diagram in Figure 5.6 shows the received bits at the bottom of the Figure, which is used to calculate the nearest code to it from the trellis diagram. This is explained in more detail below.

- At time  $t$ , for a given state, the received binary codeword is compared with each branch codeword entering this state and the Hamming Distance (HD) is calculated for both the upper and lower branches.
- The branch with the smallest HD is identified as the surviving branch. Only one surviving branch per state (or node on the trellis) is obtained.
- Repeat the above step for all other states to mark the surviving branches for each state.
- Then these steps are repeated until we reach the end of the Viterbi trellis at time 5.
- From all final state metrics, choose the minimum metric, and trace back the path from this state.
- Output the information binary digits which corresponds to branches on this trace back path.

The exact same algorithm can be used for soft-decision decoding of convolutional codes, but in this case the minimum Euclidean distance between signals are calculated rather than Hamming distance. Although theoretically an infinite decoding delay is required to obtain optimal performance, the performance degradation is negligible provided a decoding delay of at least five times the memory of the encoder is used [Charles-Lee, 1997].

## 5.7 Evaluating Error probability using the Transfer Function Bound

In this section we consider the performance achieved by the Viterbi decoding algorithm. The Viterbi decoder performs either hard decision or soft decision decoding. A hard decision decoder assigns either a 0 or 1 to each received bit. Soft decision decoding involves using outputs that are proportional to the log likelihood of the received bit being either 0 or 1. The log likelihood of the received bit being, for example, 0 if the received bit was 1. Soft decision decoding generally produces better performance at the cost of increasing complexity. The next two sections discuss the probability of error for hard and soft decision decoding in turn.

### 5.7.1 Performance of Hard-Decision Decoding

A more useful measure of performance is the bit error probability. For hard-decision decoding, the performance of the convolutional code can be analyzed by transmitting the all-zero codeword. Also from the definition of the transfer function  $T(X, Y, L)$  in Section 5.5, we know that the exponents in the factor  $Y$  contained in the transfer function indicates the number of information bit errors (number of ones) in selecting an incorrect path that merges with the all-zero path at some node. If we multiply each error probability of selecting an incorrect path by the number of incorrectly decoded information bits, for each possible incorrect path that merges with the correct path, and summing over all  $d$ , we get

$$T(X, Y, L) \Big|_{X=1, L=1} = T(Y) = \sum_{d=d_{free}}^{\infty} a_d Y^d \quad (5.16)$$

and that the total number of nonzero information bits on all paths of Hamming weight  $d$  is given by

$$\left. \frac{\partial(X, Y, L)}{\partial X} \right|_{X=1, L=1} = \sum_{d=d_{free}}^{\infty} c_d Y^d \quad (5.17)$$

where  $a_d$  denotes the number of incorrect paths of Hamming weight  $d \geq d_{free}$  that diverge from the correct path and remerge to it at some later stage, and  $c_d$  is the total number of information bit errors produced by the incorrect paths of Hamming weight  $d \geq d_{free}$  that diverge from the correct path and remerge to it at some later stage. Considering an incorrect path of Hamming weight  $d$  that diverged from the all-zero state and remerged to the all-zero state for the first time at time  $l$ . A first error event is made at time  $l$  if the Viterbi decoder favors the incorrect path. That happens when the Hamming weight between the binary received vector and the incorrect path is closer than the Hamming weight between the binary received vector and the all-zero codeword [Bahl, 1991].

If we assume that the probability of selecting an incorrect path be  $P_d$ , for all incorrect paths of any length that diverged from the all zero codeword path and remerged to the all-zero codeword path, the error event probability is simply upper bounded by

$$P_e < \sum_{d=d_{free}}^{\infty} a_d P_d \quad (5.18)$$

Comparing Equation (5.16) to Equation (5.18), the error event probability becomes

$$P_e < T(Y) |_{Y=[p_d]}^{1/d} \quad (5.19)$$

for the binary symmetric channel, it can be shown that

$$P_d = [2\sqrt{p(1-p)}]^d \quad (5.20)$$

and

$$P_e < \sum_{d=d_{free}}^{\infty} a_d [2\sqrt{p(1-p)}]^d \quad (5.21)$$

$$P_e < T(Y) \Big|_{Y=2\sqrt{p(1-p)}} \quad (5.22)$$

where  $p$  is the channel transition probability.

Because each error event causes a number of nonzero information bit error, if we weighted each error event by the total number of nonzero information bits on all paths of Hamming weight  $d$ , the bit error probability per decoded information bit is bounded by

$$P_b < \frac{1}{k} \sum_{d=d_{free}}^{\infty} c_d P_d \quad (5.23)$$

and in terms of the partial derivative of the transfer function,

$$P_b < \frac{1}{k} \frac{\partial(X, Y, L)}{\partial X} \Big|_{X=1, Y=2\sqrt{p(1-p)}, L=1} \quad (5.24)$$

### 5.7.2 Performance of Soft-Decision Decoding

As in our treatment of hard-decision decoding, for soft-decision decoding we begin by determining the first event error probability. The all-zero path is assumed to be transmitted. For soft-decision decoding, the received codeword can be any real number. Because of channel fading, noise, interference, and other factors. A signal transmitted as a 0, for example, may be received as nonzero real value. From a priori knowledge of the channel, we can estimate the probabilities of the received

value falling in certain regions. A similar analysis holds for the transmitting of a signal 1. In general, for AWGN channels and unquantized received signals, it can be shown that the probability of choosing a path at a distance  $d$  from the correct path is [Haykin, 1994]

$$P_d = Q\left(\sqrt{\frac{2d(k/n)E_b}{N_o}}\right) \quad (5.25a)$$

and for uncoded case the probability of error is

$$P_d = Q\left(\sqrt{\frac{E_b}{N_o}}\right) \quad (5.25b)$$

where  $Q$  is the  $Q$ -function defined by:

$$Q(\alpha) \equiv \frac{1}{\sqrt{2\pi}} \int_{\alpha}^{\infty} e^{-\beta^2/2} d\beta \quad (5.26)$$

and  $E_b/N_o$  is the average ratio of bit energy to noise power spectral density.

Although the above expression gives the first-event error probability for a path of distance  $d$  from the all-zero path, there are many possible paths with different distances that merge with the all zero path at a given node. In fact, the transfer function  $T(Y)$  provides a complete description of all the possible paths that merge with the all zero path and their distances. Upon performing this summation, we obtain an upper bound on the first event-error-event error probability in a form similar to Equation (5.18)

$$P_e \leq \sum_{d_H=d_{free}}^{\infty} a_d P_d \quad (5.27)$$

and the bit error probability is

$$P_b < \frac{1}{k} \sum c_d Q(\sqrt{2d(k/n)E_b / N_o}) \quad (5.28)$$

It is much easier to compute the error-event and bit error probabilities from the channel transition probability and the transfer function of the code. For a binary input and Q-ary output discrete memoryless channel, the error-event probability is

$$P_e < T(X, Y, L) \Big|_{X=1, Y=\sum_{j=1}^Q \sqrt{P(j/0)P(j/1)}, L=1} \quad (5.29)$$

and the bit error probability per decoded information bit is

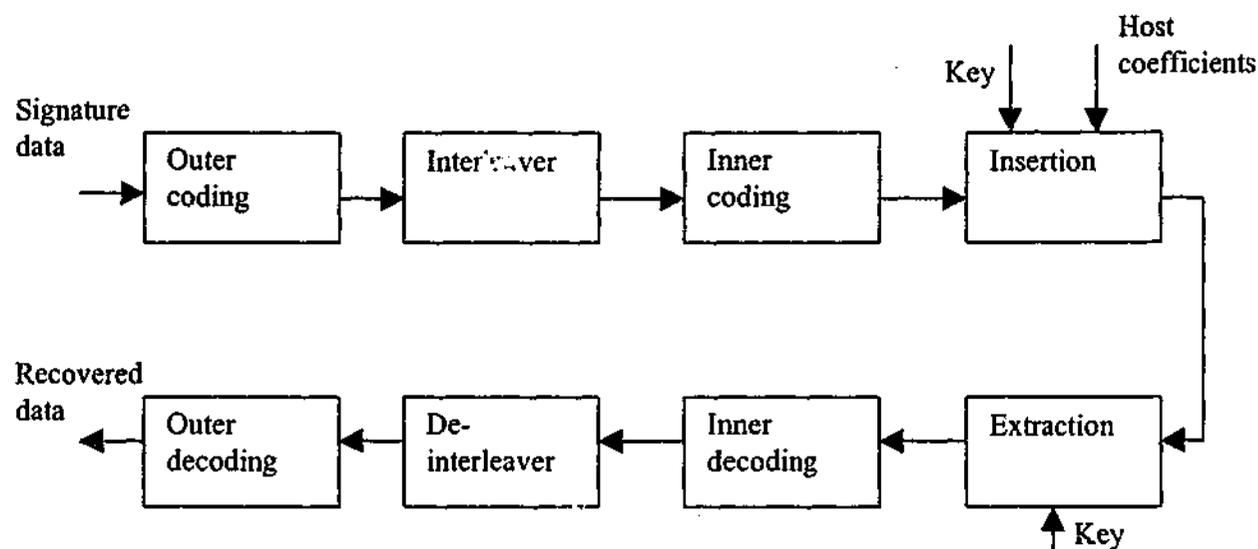
$$P_b < \frac{1}{k} \frac{\partial T(X, Y, L)}{\partial X} \Big|_{X=1, Y=\sum_{j=1}^Q \sqrt{P(j/0)P(j/1)}, L=1} \quad (5.30)$$

$P(j/i)$  is the channel transition probability between the  $i$ th input and the  $j$ th output of the discrete memoryless channel for  $i = 0$  or  $1$ . In other words, the notation of  $P(j/i)$  represent the probability of receiving  $j$  data while sending  $i$  data. Using this definition, we can describe a transmitting channel by a set of transfer probabilities. For instance, Table 5.1 shows the channel transition probability of signal 0 being received in range  $(-\text{Infinity}, 1/10]$  equals  $2/5$  or  $P((-\text{Infinity}, 1/10] | 0) = 2/5$ . The probability of signal 1 being received in range  $(-\text{Infinity}, 1/10]$  equals  $1/16$  or  $P((-\text{Infinity}, 1/10] | 1) = 1/16$ . In this table, the round brackets indicate an open range. Open ranges do not include the boundary points. Square brackets indicate a closed range. Closed brackets include the boundary points.

## 5.8 Concatenated Codes

We have shown that powerful codes can achieve a small probability of error. Unfortunately, even for moderately large values of  $d_{free}$  ( $d_{min}$ ) in the convolutional (block) code, decoding becomes too complicated. As mentioned earlier, the same problem occurred in deep-space communications where transmitted power is severely limited [Pattan, 2000].

A popular solution to this problem is that of concatenated codes, proposed by Forney [Forney, 1966] and summarized in Figure 5.9 for a typical data hiding application that only one level of concatenation is shown, but the idea is easily generalized to any level. In our context, the inner code would be a binary block ( $n$ ,  $k$ ) or convolutional ( $k/n$ ) code and the outer code would be a block code (typically, a Reed-Solomon). With concatenation it is possible to achieve a  $d_{min}$  which is the product of the minimum distance of the two concatenated codes; on the other hand, decoding complexity is simply that of individual code. An important element of the concatenated codes is the interleaver, which is a device that simply produces a deterministic permutation of the symbols at the output of the outer encoder. This permutation prevents the error bursts at the output of the inner decoder from appearing at the input of the output decoder, making it easier to correct them. These error burst are common at the output of convolutional decoders. Note that interleaver memory, a frequent concern when designing interleavers for communications applications is not so critical here due to the availability of the entire image.



**Figure 5.9** Model for concatenated coding in data embedding.

The performance of probability of bit error  $P_b$  vs. PSNR curves for concatenated codes are very steep, meaning that a small increase in the PSNR produces an enormous improvement in the overall  $P_b$ . In practice, complexity issues should be taken into account but concatenated codes used in digital communications will also perform well for data hiding purposes.

## 5.9 Modulation Codes (Trellis Codes)

In general, block and convolutional codes produce sequences with added redundancy bits. This new encoded binary stream is then added to the host coefficients in the watermarking process. The extra redundancy bits increase the required bandwidth for the same transmitted information rate. Coding and modulation are combined together to improve both bandwidth utilization and reliability to create what is known as modulation codes or trellis codes [Biglieri and

Luise, 1992]. Most schemes use the structure of trellis codes and make use of soft decision to achieve reliability at higher data rate.

Trellis coding is closely related to more traditional convolutional coding technique. In fact, every trellis code is based on a convolutional code. The key difference is the choice of modulation tightly coupled with the design of the convolutional encode. Every combination of encoded bits is mapped to a point in a signal constellation. The mapping is performed so as to maximize the squared Euclidean distance between distinct sequences of symbols, which can be transmitted. A special set of rules, known as "set partitioning", are used to devise this mapping. The mapping rules for common trellis codes are described in the papers by Ungerboeck [Ungerboeck, 1982], and [Ungerboeck, 1987]. The types of modulation most frequently used in trellis codes are M- array PSK and QAM signal constellations. The receiver for a trellis coded system uses a soft-decision Viterbi decoder to find allowable sequence of transmitted symbols which lies at the closest squared Euclidean distance to the received signal.

In the presence of AWGN, maximum likelihood decoding of trellis codes consists of finding that particular path through the trellis with minimum squared Euclidean distance to the received sequence. Thus, in the design of trellis codes, the emphasis is on maximizing the Euclidean distance between code vectors (or equivalently, code words) rather than maximizing the Hamming distance of an error-correcting code. Accordingly, the Euclidean distance is adopted as the distance measure of interest.

The approach used to design this type of trellis codes involves partitioning an  $M$ -array constellation of interest successively into 2, 4, 8, ... subsets with size  $M/2$ ,  $M/4$ ,  $M/8$ , ..., and having progressively larger increasing minimum Euclidean distance between their respective signal points. Such a design approach by set partitioning represents the "key idea" in the construction of efficient coded modulation techniques for band-limited channels.

Based on the subsets resulting from successive partitioning of a two-dimensional constellation, we may design relatively simple and yet highly effective coding scheme. Specifically, to send  $n$  bits/symbol with quadrature modulation (i.e., one that has in-phase and quadrature components), we start with a two-dimensional constellation of  $2^{n+1}$  signal points appropriate for the modulation format of interest. In any event, the constellation is partitioned into 4 or 8 subsets. One or two incoming bits per symbol enter a rate-1/2 or rate 2/3 binary convolutional encoder, respectively; the resulting two or three coded bits per symbol determine the selection of a particular subset. The remaining uncoded data bits determine which particular point from the selected subset is to be signaled. This class of trellis codes is known as Ungerboeck codes. Figure 5.10 shows an example of Ungerboeck encoder structure. A rate 2/3 trellis code is shown in Figure 5.11. In this figure, the most significant bit of the input bits is left uncoded.

Since the modulator has memory, we may use the Viterbi algorithm to perform maximum likelihood sequence detection at the receiver. Each branch in the trellis of the Ungerboeck code corresponds to a subset rather than an individual signal point. The first step in the detection is to determine the signal point within each

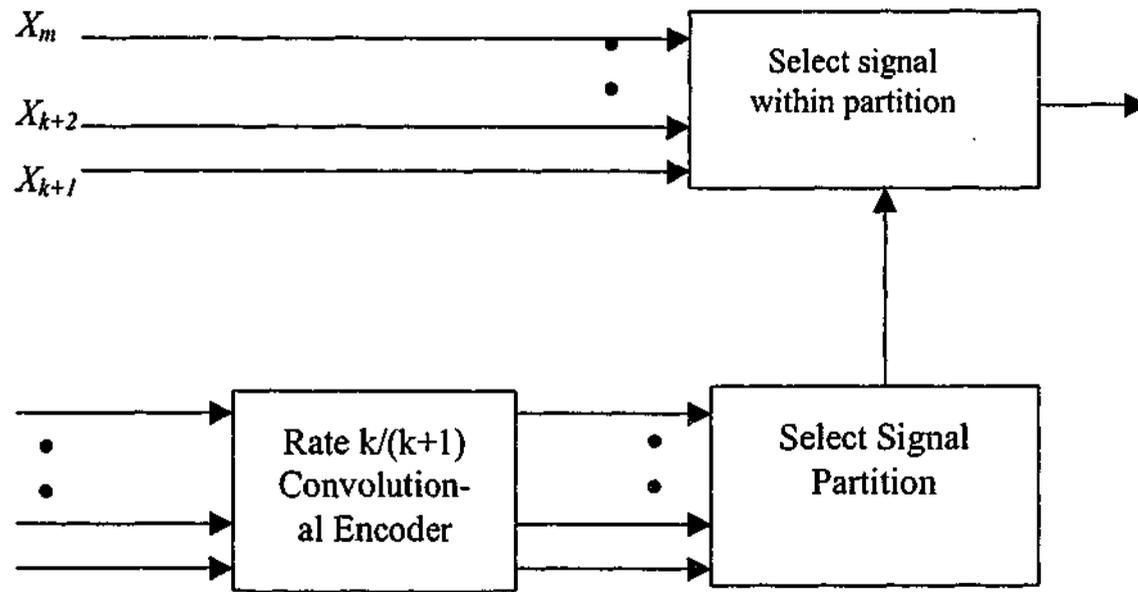


Figure 5.10 Ungerboeck TCM encoder structure.

subset that is closest to the received signal point in the Euclidean sense. The signal point so determined and its metric (i.e., the squared Euclidean distance between it and the received point) may be used thereafter for the branch in question, and the Viterbi algorithm may then proceed in the usual manner.

A trellis code defined on a finite set of points in the complex plane is called an Ungerboeck code, and the resulting waveform  $c(t)$  is called a trellis-coded modulation waveform. For a given fixed  $k$  and  $n$ , an  $(n, k)$  Ungerboeck code is designed to encode  $k$  bits in each frame (stage) into a complex signal constellation with  $2^n$  points, the encoding depending also on the state of the encoder. The code alphabet is the signal constellation, a discrete set of points chosen from the field of complex numbers. A codeword is a sequence of these complex numbers [Blahut, 1990].

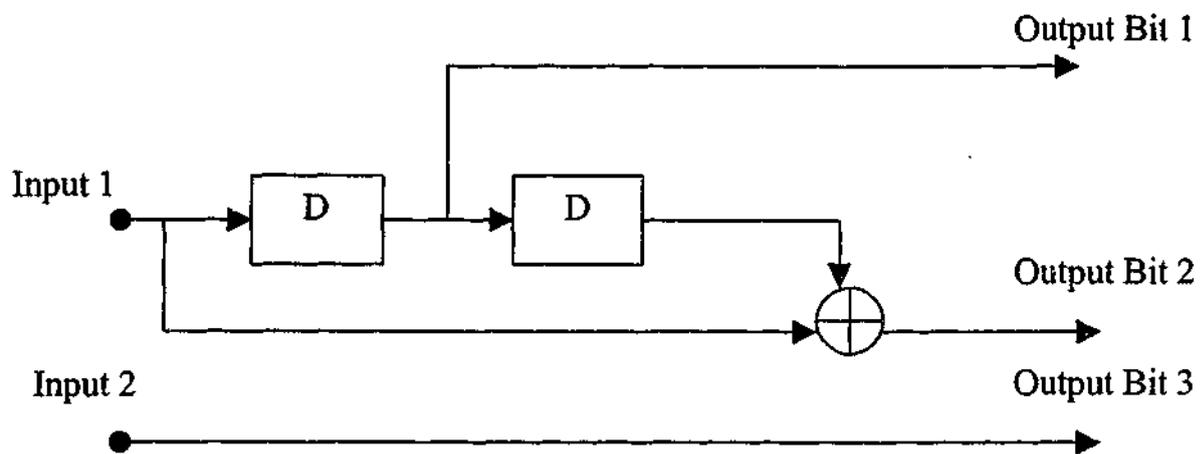


Figure 5.11 Transfer function for a convolutional code of rate 2/3.

The frequently used measure of the performance of a trellis code is the free Euclidean distance  $d_{free}$ , defined as the minimum Euclidean distance between any two distinct signals which could be transmitted by the encoder. The probability of an error event  $P_e$  may be related to  $d_{free}$  by the simple formula

$$P_e \approx N_{free} Q \left[ \sqrt{\frac{d_{free}}{\sigma}} \right] \quad (5.31)$$

where  $N_{free}$  is the number of distinct signals which lay at distance  $d_{free}$  from the desired signal,  $\sigma$  is the variance of the noise, and  $Q(\cdot)$  is the standard Q-function defined earlier in Equation (5.26). This approximation is valid for large values of  $E_b/N_o$ . Better approximations based on the codes transfer function also exists. Another common measure of performance of a trellis coded communications system is the asymptotic coding gain, defined as the reduction in  $E_b/N_o$  required to achieve a desired low Bit Error Rate (BER) as  $E_b/N_o$  becomes large. Trellis codes can have asymptotic coding gains of nearly 9 dB in Gaussian noise.

The performance of trellis codes will improve as the encoder memory increases, although the Viterbi decoder will become more complex.

## 5.10 Turbo Codes

Turbo codes are a new class of error correction codes that were introduced along with a practical decoding algorithm in [Berrou, et. al., 1996]. Turbo codes has been developed to have large block lengths, yet contain enough structure to make practical decoding possible. Parallel concatenation of convolutional codes is used to give the codes structure so they can be decoded easily. Pseudo-random interleaving is used to give the codes performance which approaches that for random coding.

As mentioned earlier in Section 5.9 that a good trade-off between coding gain and complexity can be achieved by serial concatenated codes. The primary reason for using a concatenated code is to achieve a low error rate with an overall decoding complexity lower than that required for a single code of the same rate. Turbo codes exploits a similar idea of connecting two codes and separating them by an interleaver. The difference between turbo and serial concatenated codes is that in turbo codes two identical systematic codes are connected in parallel. The information code for the second code are not transmitted thus increasing the code rate relative to a corresponding serial concatenated code. Although any systematic encoder can be used for the component encoders of the turbo encoder, the common practice is to use recursive systematic convolutional (RSC) encoders. The use of

convolutional encoder makes it possible for the decoder to utilize a modified version of the Viterbi algorithm [Schlegel, 1997].

The original turbo code is a parallel concatenation of two RSC codes, while the turbo decoder consists of two concatenated decoder of the component codes separated by the same interleaver. The component decoders are based on a maximum a posteriori (MAP) probability algorithm or a soft output Viterbi algorithm (SOVA) generating a weighted soft estimate of the input sequence. The iterative process performs information exchange between the two component decoders. The turbo encoder and decoder are discussed in more details in the next sections.

### 5.10.1 Turbo Encoder

A turbo encoder is formed by parallel concatenation of two RSC encoders separated by a random interleaver. The encoder structure is called parallel concatenation because the two encoders operate on the same set of input bits, rather than one encoding the output of the other. A block diagram of a basic rate 1/3 turbo encoder is shown in Figure 5.12. In this Figure, the two RSC encoders are of rate 1/2 and each data bit generates two separate sets of check bits (one for each code). The data is interleaved according to a pseudo-random pattern prior to generation of second set of output bits so that the two sets of check bits are independently generated. Therefore, the output of the encoder consists of the information sequence and two parity sequences, thus it has a code rate of 1/3. For many applications, it is common practice to *puncture* the output of the encoder in order to increase the code rate to 1/2. Alternately, puncturing (deleting) bits from the two

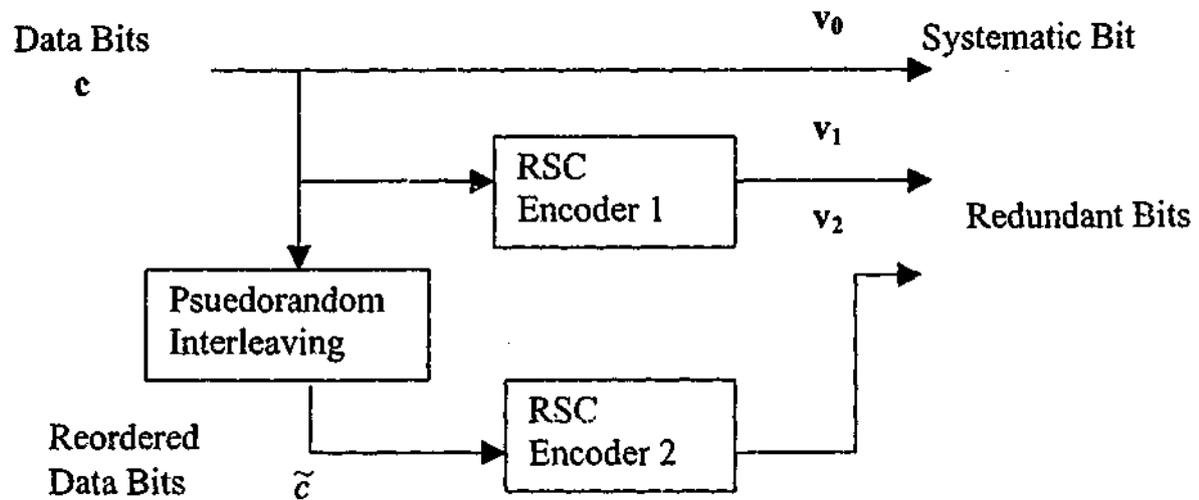


Figure 5.12 A turbo encoder for basic rate 1/3.

parity sequences produces a code rate of 1/2. Other code rates can be achieved by using additional parity generators and/or different puncturing patterns.

For a rate 1/2 punctured turbo code, the first output bit is the input bit itself, while the second output bit is generated by multiplexing the output of the RSC encoders.

From the encoder in Figure 5.12 the same information sequence is encoded twice but in a different order. The first encoder operates directly on the input sequence, denoted by  $c$ , of length  $N$ . The first RSC encoder has two outputs. The first output, denoted by  $v_0$ , is equal to the input sequence since the encoder is systematic (one of the outputs is the input). The other output is the parity check sequence, denoted by  $v_1$ . The interleaved information sequence at the input of the second encoder is denoted by  $\tilde{c}$  only, the parity check sequence of the second encoder, denoted by  $v_2$ , is transmitted. The information sequence  $v_0$  and the parity check sequences of the two encoders,  $v_1$ , and  $v_2$ , are multiplexed to generate the turbo code sequence. The overall code rate is 1/3.

The interleaver in turbo coding is a pseudo-random block scrambler defined by a permutation of  $N$  elements with no repetitions. The first role of the interleaver is to generate a long block code from small memory convolutional codes. Secondly, it decorrelates the inputs to the two decoders so that an iterative suboptimal decoding algorithm based on information exchange between the two component decoders can be applied. The decorrelation of the input sequences at the two component decoders means that there is a high probability that after correcting some of the errors in one decoder some of the remaining errors should be correctable in the second decoder [Reed and Chen, 1999]. In a pseudo-random interleaver a block of  $N$  input bits is read into the interleaver and read out pseudo-randomly. The pseudo-random interleaving pattern must be available at the decoder.

### 5.10.2 Decoding of Turbo Codes

The important feature of turbo codes is the iterative decoder, which uses a soft-in/soft-out maximum a posteriori probability (APP) decoding algorithm. This algorithm is more complex than the Viterbi algorithm by about a factor of 3, and for conventional convolutional codes it offers little performance advantage over Viterbi decoding. However, in turbo decoding, the fact that it gives the maximum MAP estimate of each individual information bit, is crucial in making it possible for the iterative decoding procedure to converge to the optimal decoding solution. The convergence properties of the iterative algorithm remain an interesting open research question and are discussed in the literature [Reed and Chen, 1999, Vucetic and Yuan, 2000].

In the original paper on turbo codes [Berrou, et. al., 1996], Berrou, et. al. Proposed an iterative decoding scheme based on an algorithm by Bahl et. al. [Bahl, et. al., 1974]. The Bahl decoding algorithm differs from the Viterbi algorithm in the sense that it produces soft outputs. While the Viterbi algorithm outputs either a 0 or 1 for each estimated bit, the Bahl algorithm outputs a continuous value that weights the confidence or Log-Likelihood of each bit estimate. While the goal of the Viterbi decoder is to minimize the codeword error by finding a maximum likelihood estimate of the transmitted code word, the Bahl algorithm attempts to minimize the bit error rate by estimating the a posteriori probabilities of the individual bits of the code word.

To decode the received bits, a decoder is designed for each elementary encoder. Each elementary decoder uses the Bahl algorithm to produce a soft decision for each received bit. In other word, each decoder estimates the a posteriori probability of each data bit. The APP's is used as a priori information by the other decoder. The decoding continues for a set of number of iterations. Generally performance improves from iteration to iteration, but follows a law of diminishing results. Figure 5.13 shows a block diagram of turbo decoder.

In this latter Figure, the received information stream is passed to both decoders and the parity stream is demultiplexed into the received parity-check bits. The demultiplexer sends the parity bits generated by the first encoder in Figure 5.12 to decoder 1, and the parity bits generated by second encoder to decoder 2. The first decoder produces a soft decision which is based on its received parity bits along with the received information bits. This soft decision is interleaved and passed as

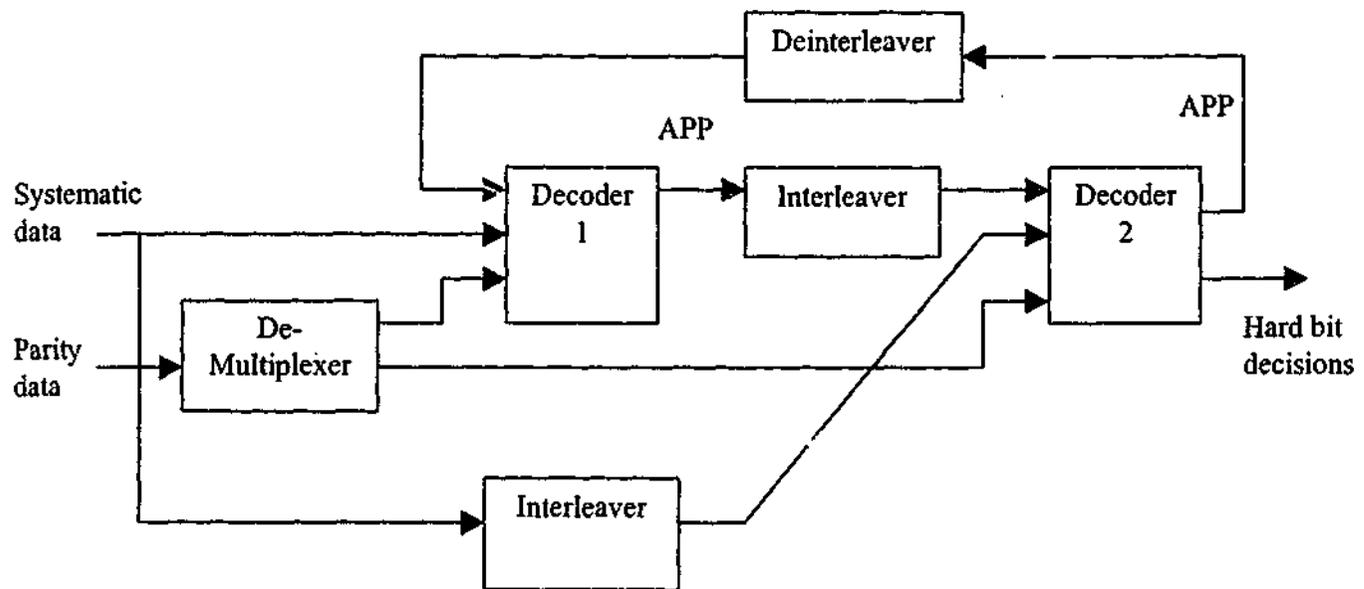


Figure 5.13 Turbo decoding block diagram.

the systematic input to decoder 2. The second decoder uses its received parity bits along with the interleaved soft decision from the first decoder to produce a soft decision of its own.

The soft decision at the output of the second decoder does not constitute a maximum a posteriori estimate of the codeword since the two decoders to this point are independent. In order to find an estimate that approaches the MAP estimate, the information produced by the two decoders are shared with one another in an iterative fashion. As the number of iterations approaches infinity, the estimates at the output of both decoders approaches the MAP solution. In practice, the number of iterations need to be only about 18, and in many cases as few as 5 iterations can provide satisfactory performance [Vucetic and Yuan, 2000].

Up to this point it has been assumed that the decoders of Figure 5.13 uses the Bahl algorithm. However, to reduce the complexity of the decoders a modification of

Viterbi algorithm is used to produce soft outputs. The algorithm is called the Soft-Output Viterbi Algorithm, or SOVA algorithm [Hagenauer, 1996].

Several turbo coding systems are already developed and we make use of such systems to encode the signature data before embedding them in the host image. Results of using turbo codes in the embedding process is shown in the next section and a comparison with other convolutional codes for the same rate is also produced.

### 5.11 Implementation and Experimental Results

In order to demonstrate the effectiveness of channel coding in the performance of data embedding we have watermarked several images with different types of signature data using the above mentioned codes. We specifically make use of the Haar wavelet transform for all simulations. The implementation is as specified in Section 5.2 with a scaling parameter  $\alpha = 10$  as suggested before. The method involves adding the watermark sequence to the low frequency DWT coefficients of the host image. The watermarked image is formed by taking the inverse DWT of the modified coefficients. The signature data comprises of images of size  $\frac{1}{4}$ th the host image and randomly generated binary watermarks (with a uniform distribution) are embedded within the host image.

We perform two classes of tests. We first demonstrate the performance of the proposed method in embedding and recovering watermarks when the watermarked image undergoes distortions. The resulting watermarked signal is distorted (corrupted) using two of the most common distortions (JPEG compression and noise addition separately). The watermark is then extracted and compared with the

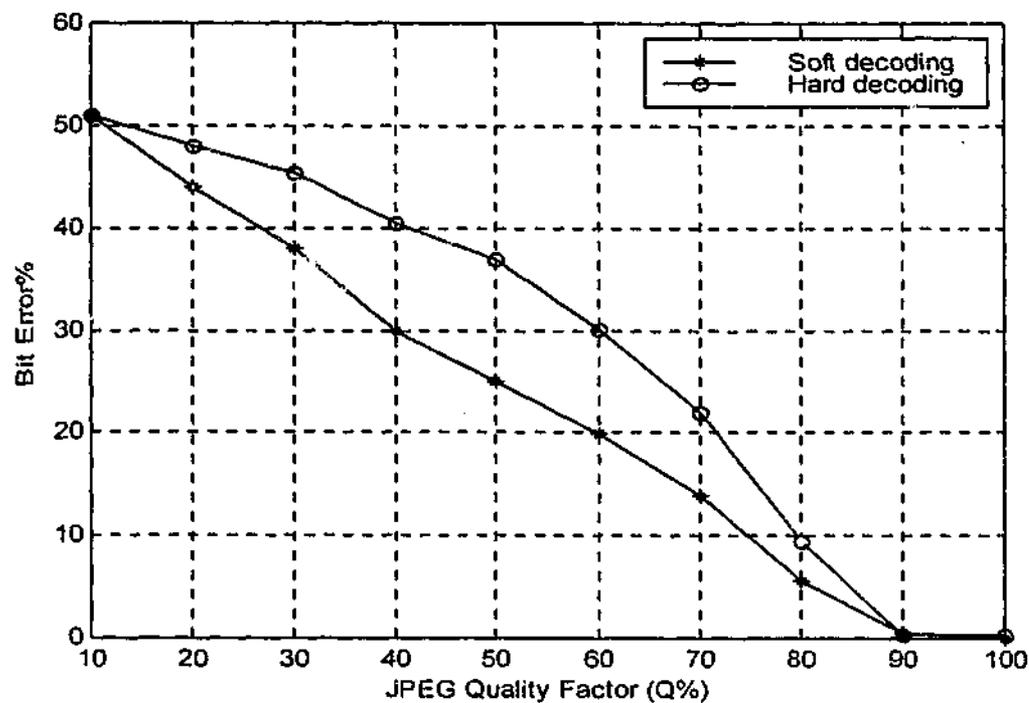
original watermark sequence to measure robustness and extraction capability of the technique.

In the next set of tests we demonstrate the improved performance of using channel coded watermarks over uncoded watermarks. In this test, the watermark sequence is left uncoded before embedding it in the host coefficients.

The following is a more detail counterpart of the simulation work.

### 5.11.1 Convolutional Codes

In this simulation the host image of *Lena* of size 256 x 256 was watermarked with *Bear* image of size 128 x 128 using the method described in this chapter. The *Bear* image was first compressed by a factor of 4 using vector quantization. The indices obtained were coded using convolutional code of rate  $\frac{1}{2}$  with generator function of (5,7). The transfer function of this encoder is shown in Figure 5.3. The constraint length in this case is 3 and the free distance  $d_{free}$  is 5. The watermarked image was subjected to JPEG compression and the watermark data was extracted from the compressed image. To decode the extracted data, Viterbi algorithm was implemented for both soft and hard decoding. The decoded bits are then source decoded to obtain the watermark image of *Bear*. The results of the BER versus JPEG compression are shown in Figure 5.14 for both hard and soft decoding. The improvement of the soft-threshold decoding over hard-threshold decoding is quite evident in this figure. Nonetheless, the bit error rate for hard-threshold decoding was still slightly better than of block codes for the case of BCH(4,7) that has been obtained in Chapter 4 in Figure 4.11.



**Figure 5.14** Soft and hard threshold decoding of signature image of *Bear* image extracted from the host image *Lena*.

The channel transition probability used for soft-threshold decoding is shown in table 5.1 on page 136, and the scale factor  $\alpha$  was 10.

Using the same procedure, the image of *Bird* of size 25% of the host image was used as the signature data. The indices of the vector quantizer were channel coded using convolutional code of rate  $2/3$ . The transfer function of the convolutional code with rate  $2/3$  is shown in Figure 5.15. The watermarked image of *Lena* was JPEG compressed before trying to retrieve the watermark image from it.

Figure 5.16 shows the recovered image of *Bird* at different quality factor for JPEG compression. The recovered *Bird* image is quite recognizable even after high compression ratio ( $Q < 20\%$ ). The PSNR of the recovered *Bird* image is shown in Figure 5.17 as a function of JPEG compression for different quality factor.

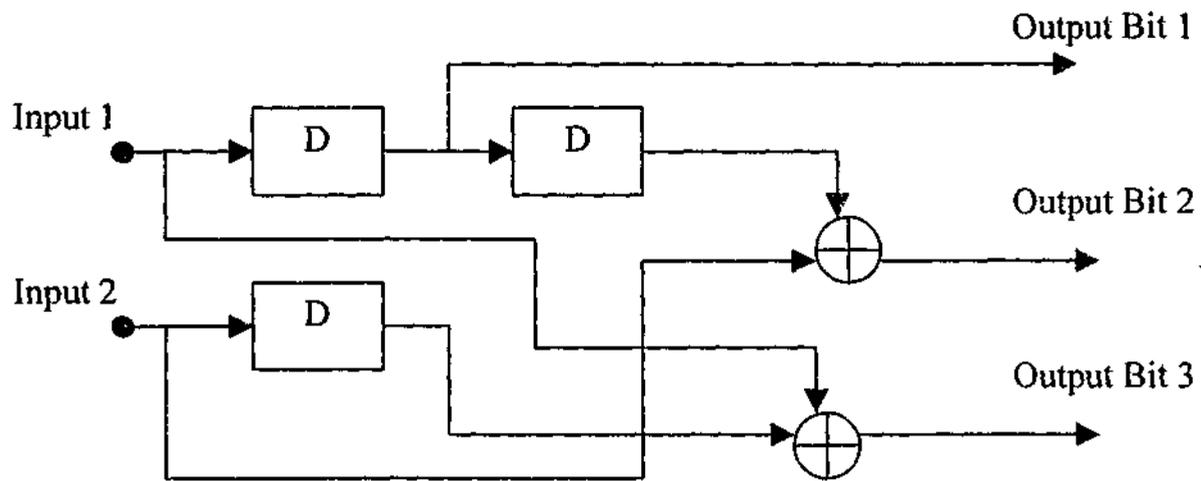


Figure 5.15 Transfer function of convolutional code of rate 2/3.

Table 5.1 The channel receiving probability.

Receiving probability When Sending 0	Receiving probability When Sending 1
$P((-\infty, 1/10]   0) = 2/5$	$P((-\infty, 1/10]   1) = 1/16$
$P((1/10, 1/2]   0) = 1/3$	$P((1/10, 1/2]   1) = 1/8$
$P((1/2, 9/10]   0) = 1/5$	$P((1/2, 9/10]   1) = 3/8$
$P((9/10, \infty)   0) = 1/15$	$P((9/10, \infty)   1) = 7/16$

Another signature data in the shape of a message of random data of length 2000 bits was also hidden in the host image of *Lena*. The signature data was first encoded using convolutional code of rate 2/3 to obtain a code of length 3009 bits which then embedded into the DWT coefficients of the host image *Lena*. The watermarked image of *Lena* is shown in Figure 5.18. Also shown in this Figure the original image of *Lena* for comparison. It is evident from these images that the fidelity of the host image has not been compromised visibly under the embedding process.

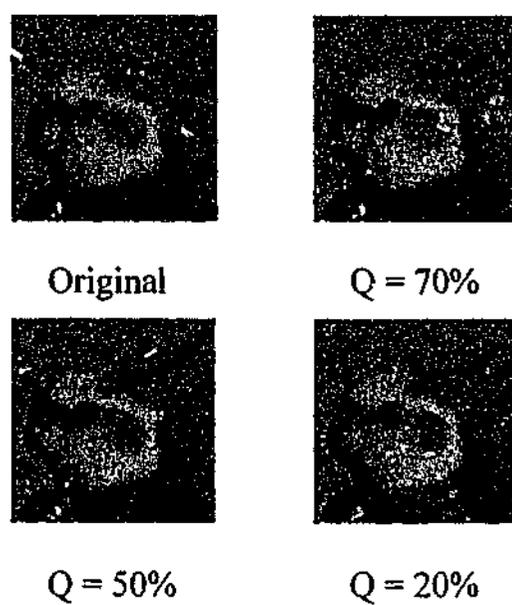


Figure 5.16 The original and recovered watermark image *Bird* for different JPEG compression.

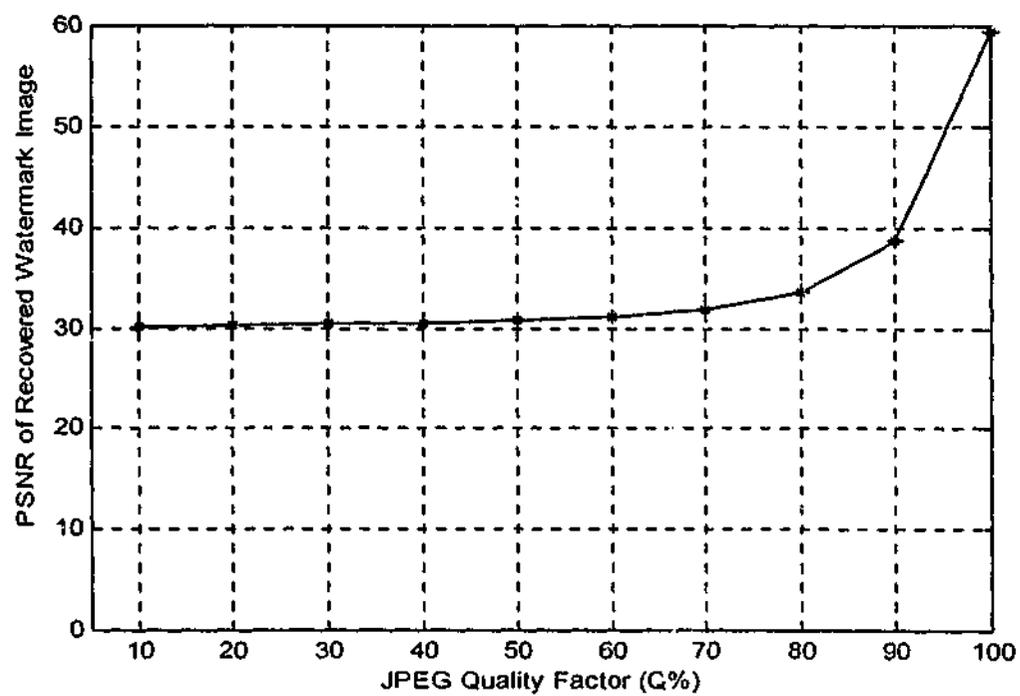
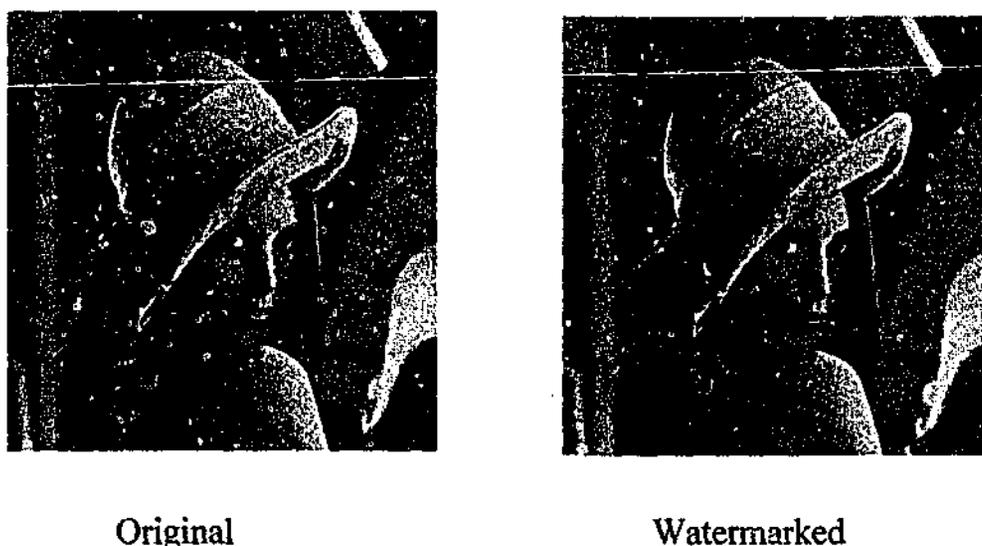


Figure 5.17 PSNR of recovered watermark image *Bird* vs. JPEG compression attack.



**Figure 5.18** Original and Watermarked *Lena* image with random 2000 bits using convolutional coding with rate 2/3.

The watermarked host image was degraded by applying JPEG compression and noise addition. To simulate the JPEG compression attack, the watermarked host image is subjected to JPEG compression with different quality factors (compression ratios). The signature data was then extracted using soft-threshold decoding. For comparison, uncoded signature was similarly hidden in the transform domain of the host image *Lena* and subjected to the same JPEG compression. The corresponding results on bit error rates were plotted in Figure 5.19. It shows a marked improvement when the signature data was channel coded. At a quality factor of  $Q = 75\%$ , say, the bit error rate could reduce from 8% to zero.

However, it could be argued that the comparison just presented may not be fair. In the coded case, the number of bits hidden are one and a half as many as that for uncoded case. The fidelity of the watermarked host images are therefore not identical. This is true to a certain extent. Nevertheless, we like to remark first that

coding of the signature data only resulted in a slight difference in the PSNR of the watermarked image for both cases. The PSNR of watermarked *Lena* in the case of the uncoded watermark was 37.35 dB, and 37.13 dB for coded the watermark. The same scale factor  $\alpha = 10$  was used in both cases.

To take into consideration the fidelity factor, another set of experiments was planned as follows. The same host image *Lena*, the same signature data, and the same convolutional code as before were used to embed the signature data into the host DWT coefficients. Instead of JPEG compression, the watermarked host image is now degraded by direct noise addition in the spatial domain. The noise is Gaussian noise with zero mean and varying power (variance). Then the signature data was extracted from the distorted image and decoded using soft-threshold decoding as before. The decoded signature data is compared to the original signature and bit error rate was calculated therefrom.

The result of bit error rate is plotted against PSNR of watermarked host image in Figure 5.20. PSNR is computed in standard manner as described in Equation (4.3) of Chapter 4, where the noise power is equal to the variance of the added Gaussian noise. The result is compared to the uncoded case and the improvement in the bit error value with channel coding is quite obvious. For the same fidelity of the host image (e.g. PSNR = 35 dB), the bit error rate of coded signature is much lower than uncoded message (e.g. zero versus 8%). The comparison In this case is fairer than that shown in Figure 5.19, because the extra code bits have already been taken into account in the PSNR measure.

The bit error rate versus fidelity curves, as shown in Figure 5.20, are not the customarily depicted in digital communications. It is more common to show bit error rate against signal-noise-ratio. As signal in this case is the signature data, not the host image, the SNR must be defined accordingly and precisely (strictly speaking the PSNR defined earlier is a misnomer). The signal strength in this case is calculated by the amplitude of the scale factor  $\alpha$  that was used in the embedding process. The noise power is equal to the additive noise variance as before. The simulation result are plotted in Figure 5.21 for both coded and uncoded signatures. From this figure, an average of 5 dB SNR is needed for the uncoded case to obtain the same bit error rate. There is a great similarity between these curves and the widely used theoretical curves representing the bit error rate versus  $E_b/N_o$  (signal-noise-ratio) in digital communication. The theoretical curves representing bit error rate versus SNR are plotted in the same figure for comparison. The theoretical curves were calculated using Equations (5.25a) and (5.25b) [Taub and Schilling, 1986]. The value of  $d$  in used Equation (5.25a) was 4 and the code rate was 2/3. The small discrepancy between the theoretical and the simulation ones is due to the distortion introduced by the wavelet transformation processes. The closeness of the theoretical and experimental results confirms in some way the correctness of the experimental procedure carried out above.

Finally, it must be added that the range of BER (0% to 50%) computed in Figures 5.19-5.21 are far too large for digital data transmission. It would be more appropriate to show BER in the range of  $10^{-3}$  to  $10^{-9}$  in log scale. Our main purpose in carrying out the experiments were to show the difference in performance of

coded and uncoded cases. By necessity, the signature length cannot be too long, and we have chosen 2000 bits for our experiment. To compute BER of  $10^{-9}$  would have required a signal length of  $10^{10}$  or equivalent. This is impractical. As it is, the BER is high below  $Q < 75\%$  in Figure 5.19 and PSNR below 30 dB in Figure 5.20. But at this level of attack, the host image would have been deteriorated to such an extent that it no longer has commercial value.

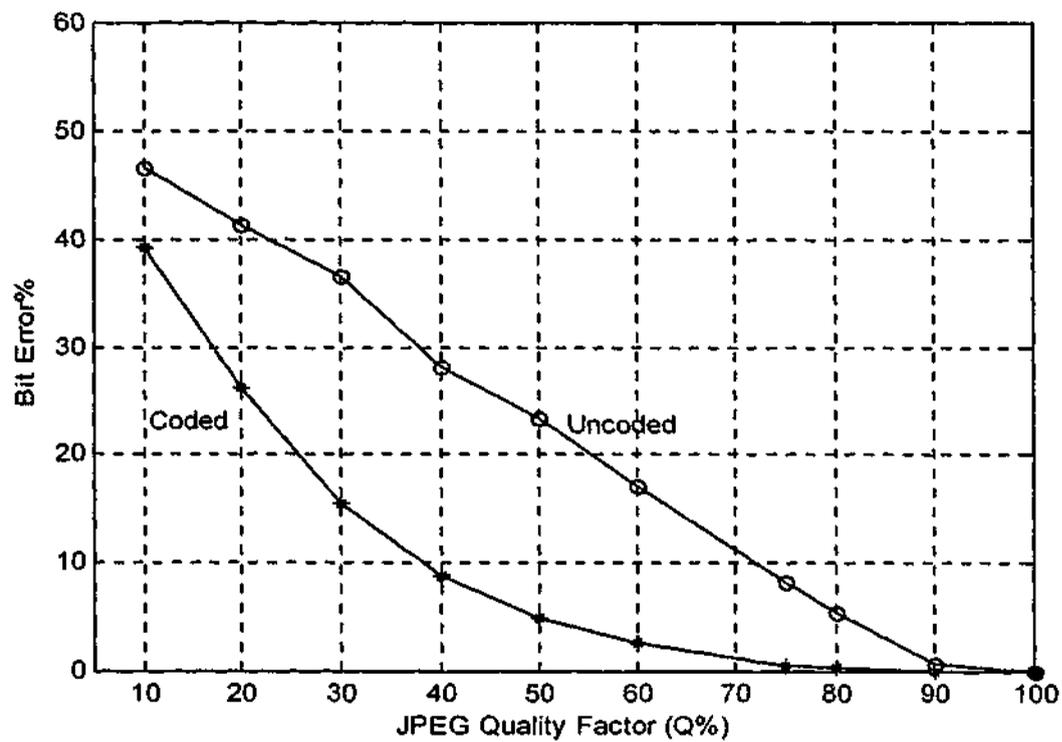


Figure 5.19 Bit error versus JPEG compression attack for different quality factors.

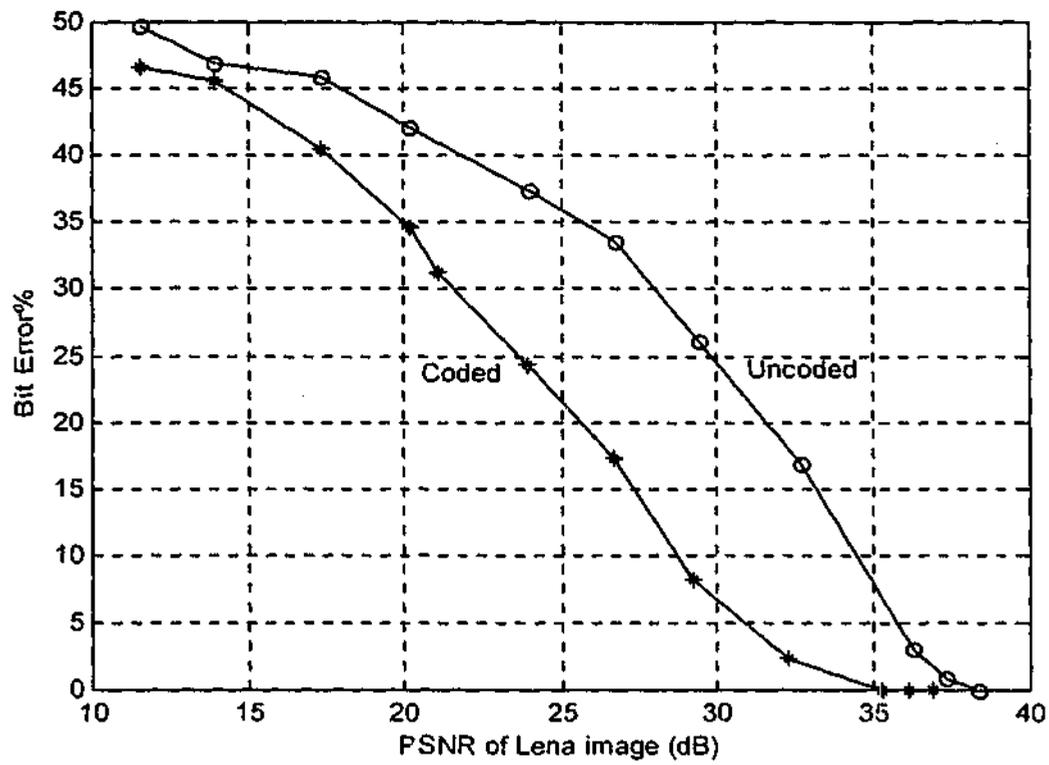


Figure 5.20 Bit error versus noise addition attack for different quality factors.

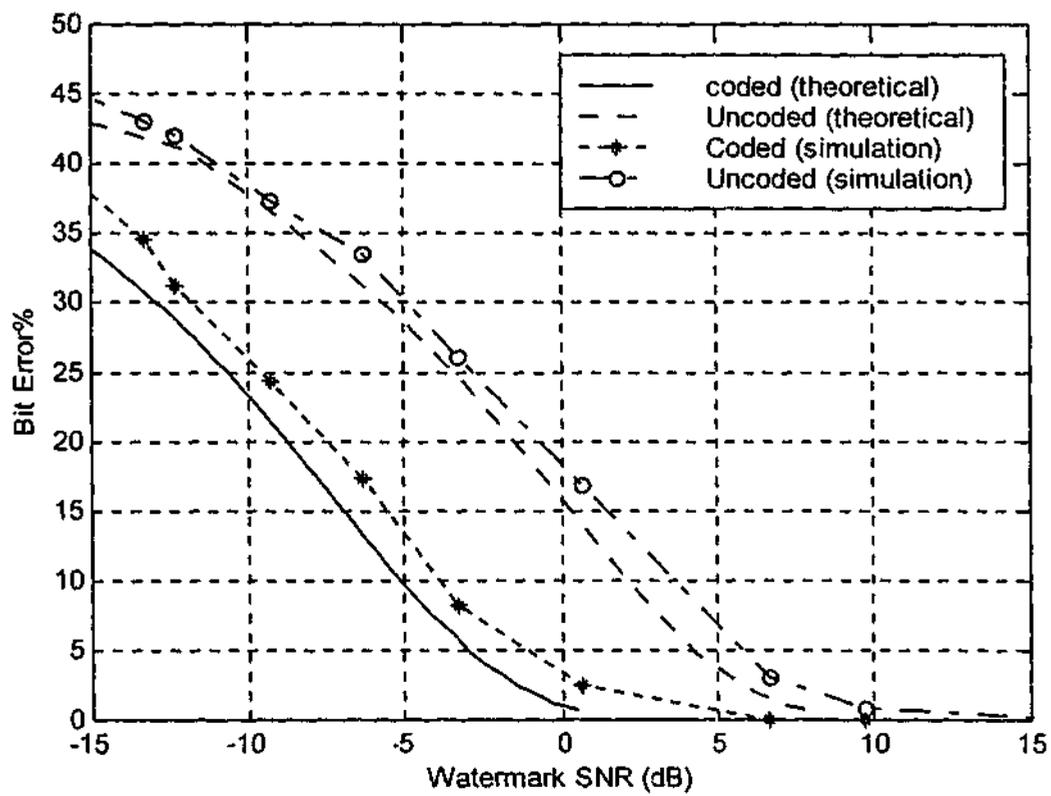


Figure 5.21 Bit error rate versus SNR of the watermark signal.

### 5.11.2 Concatenated Codes

In this simulation, the concatenated channel coding used was designed of two codes and an interleaver as in Figure 5.9. the outer code was RS (63,18) and the inner code was convolutional code of rate  $2/3$ . The convolutional code was designed using the transfer function shown in Figure 5.11 for trellis coding, which has a generator polynomials expressed in octal form as (4, 21, 10). The respective constraint length is 3 and the respective  $d_{free}$  parameter is 4. Results of embedding using trellis codes without modulation is similar to that of convolutional code with the same rate. The interleaver used was (6,630). The host image of *Lena* was watermarked with a random message of size 1080 bits and the resulting code was of length 5676 bits. The scale factor used was 10. The results for bit error of the extracted watermark as a function of JPEG compression is shown in Figure 5.22. The results of the concatenated code is compared to a single convolutional code of rate  $2/3$  and to uncoded watermark. From this graph, one can see the improvement of both concatenated and convolutional coding over uncoded case

Also from this figure, at high compression ratio, the two graphs for concatenated and convolutional coding are almost the same, however, the concatenated codes has larger margin for lossless recovery (i.e., zero bit error rate).

### 5.11.3 Turbo codes

Finally, for using turbo codes in encoding the signature data, the DWT coefficients of the host image of *Lena* is modified to enable hiding the encoded message data. A random message of length 1000 bits was turbo coded using two parallel convolutional encoders of rate  $1/2$  with transfer function = [07,05] in octal form.

Punctured encoding resulting in a rate of  $\frac{1}{2}$  for the turbo code while unpunctured code gives a rate of  $\frac{1}{3}$ . The scale factor  $\alpha = 10$  and the decoder uses logmap decoding with 5 iterations and the results were obtained over 80 frames. Results are shown in Figure 5.23 and Figure 5.24 for the bit error versus JPEG compression and noise addition respectively. The difference in bit error between the coded and uncoded messages is quite significant especially for lossless recovery (bit error = zero). However, little difference noticed between punctured and unpunctured codes. Though, the number of bits hidden in the case of punctured codes will be smaller than unpunctured code.

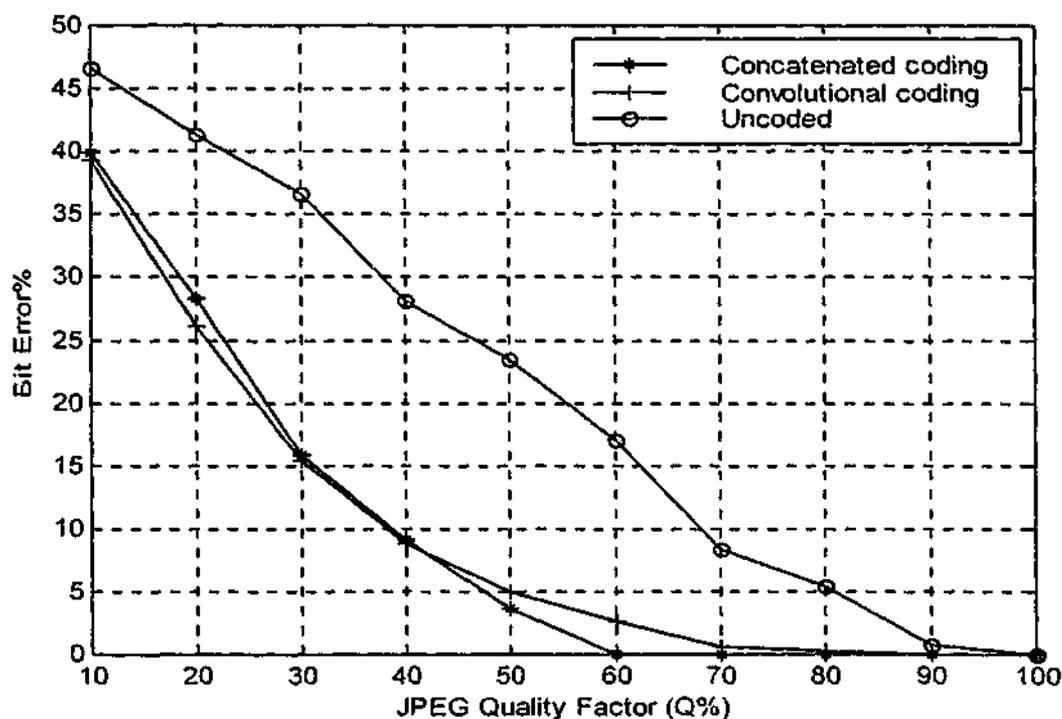


Figure 5.22 Bit error versus JPEG compression for concatenated, convolutional, and uncoded cases.

To compare the results of turbo coding to convolutional codes of similar rate, the same watermark is coded with convolutional code of rate  $\frac{1}{2}$  and the recovered data were obtained using Viterbi soft decoding algorithm. The results are shown in Figure 5.25 for the bit error versus JPEG compression for turbo coding and convolutional code and uncoded case. Similarly, Figure 5.26 shows the bit error of the recovered watermark against the PSNR of the host *Lena* image for the noise addition attack for the two different codes and uncoded watermark. From these figures one can see the improvement of turbo codes over convolutional codes and uncoded case. Notice also that at high compression ratio and low PSNR the three graphs seems to be close but at this point the host image is of no commercial value and therefore the high bit error is not of much concern.

## 5.12 Summary

In this chapter we have given an outline of the advantages that channel coding using convolutional codes brings about in data embedding applications, while the similarities with the detection problem in digital communication have also been pointed out. We have chosen the bit error probability as the reference quality measure. Analysis of the different coding schemes reveals superior performance of convolutional codes for a reasonable complexity. Comparison with uncoded case shows gains of up to 5 dB for simple codes. Note that a coding gain of 3 dB allows doubling the number of hidden information bits for the same  $P_b$ . Soft decision decoding resulted in a better performance than hard decision decoding, because some information is lost as a result of making hard decision instead of soft

decision. We also have given theoretical results (see Figure 5.21) that are presented in a way that becomes independent of the image to be watermarked.

Finally, we have also discussed the benefits of coding, especially when concatenated codes and turbo codes are employed. However, the major disadvantages of turbo codes are its long decoding delays, due to the large block lengths and iterative decoding, and its weaker performance at lower bit errors, due to its low free distance. In spite of that, its performance was superior to convolutional codes with similar rates. Moreover, the encoding complexity and long delays in the case of turbo codes are not an issue, especially if the encoding and decoding is not to be implemented on real time, which is the case for most image watermarking applications. It is worth noting here that the codes used here were chosen for simplicity of implementation and to obtain a qualitative comparison rather than extreme coding gains. Our best scheme (turbo code) can be further improved by using much longer codes.

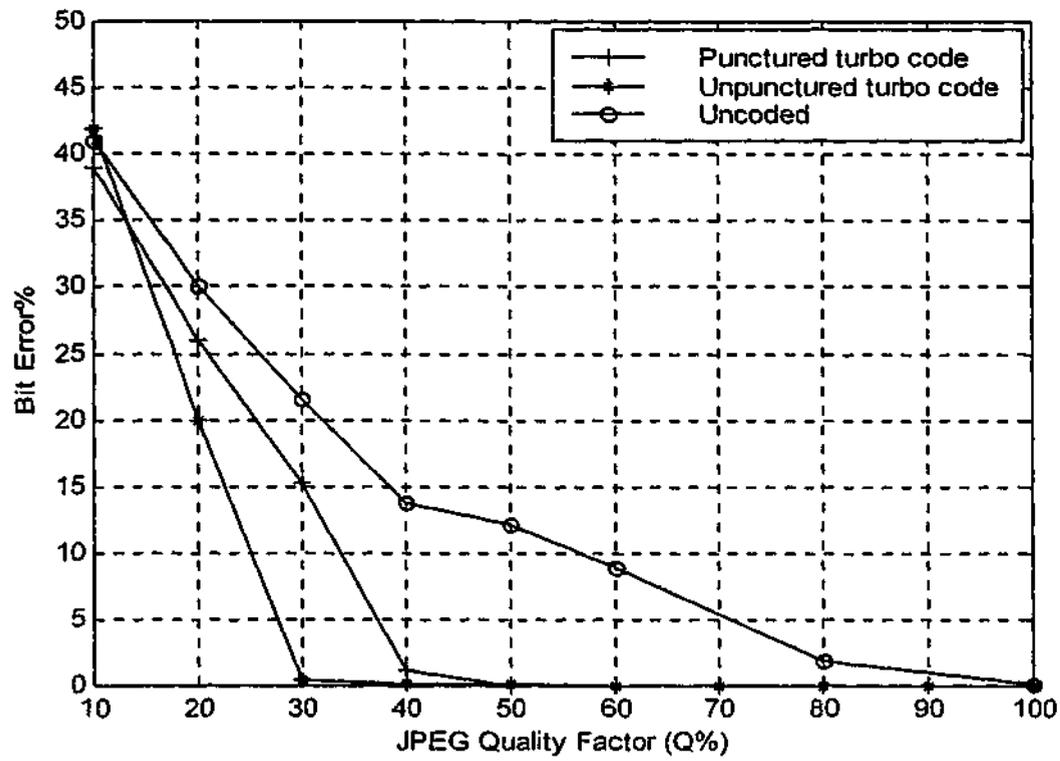


Figure 5.23 Bit error rate versus JPEG compression for turbo codes and uncoded messages.

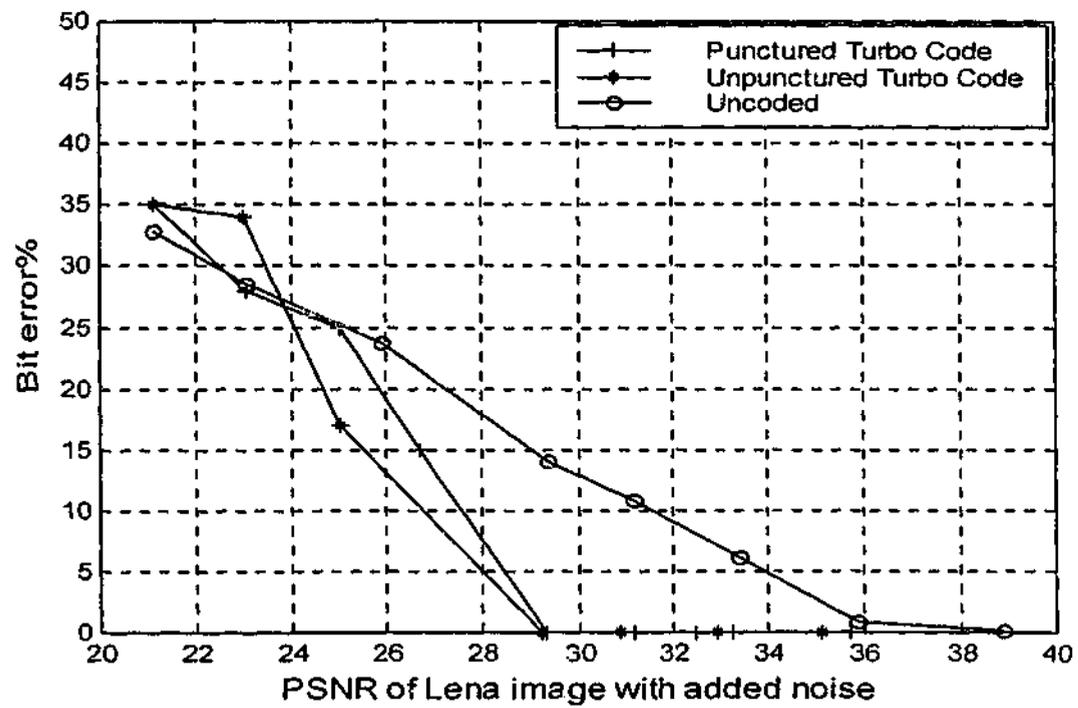


Figure 5.24 Bit error rate versus noise addition attack for turbo coded and uncoded messages.

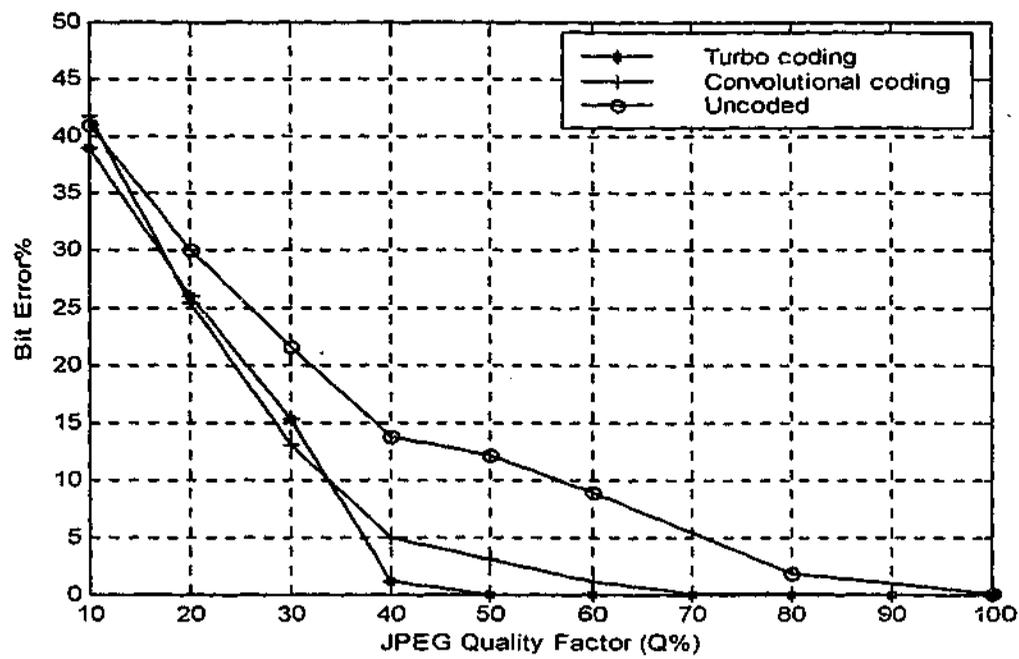


Figure 5.25 Bit error rate versus JPEG compression for turbo codes, convolutional code, and uncoded messages.

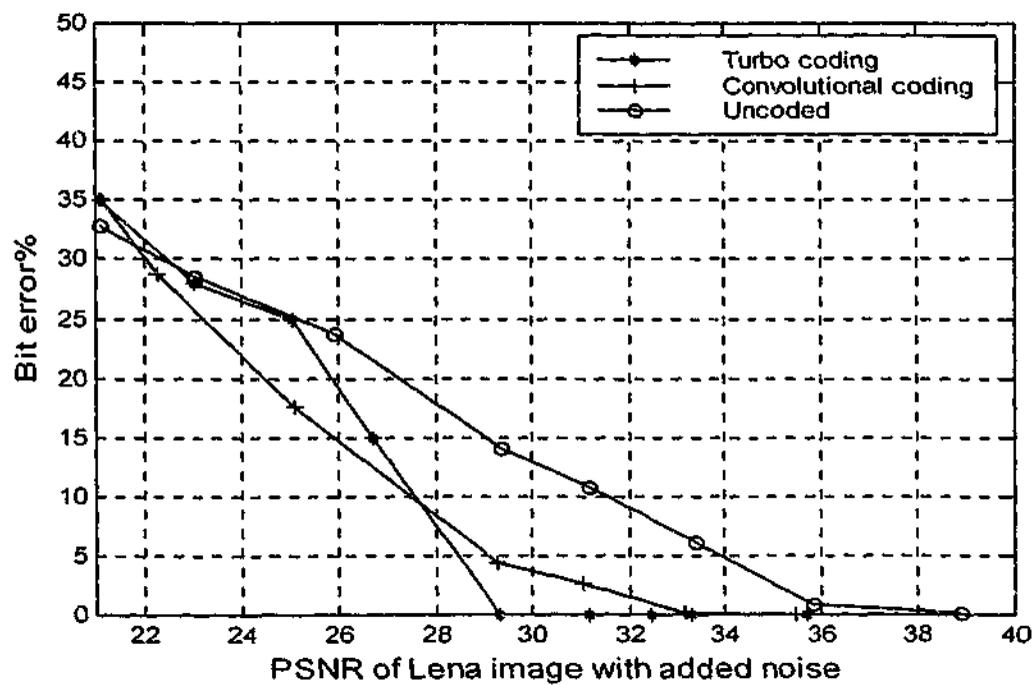


Figure 5.26 Bit error rate versus noise addition attack for turbo coded, convolutional code and uncoded messages.

## **C h a p t e r 6**

### **Hidden Data Extraction without Original Host**

- 6.1 Embedding Technique**
- 6.2 Extraction Technique**
- 6.3 Watermark Detection**
- 6.4 Implementation and Experimental Results**
- 6.5 Summary**

## Hidden Data Extraction without Original Host

---

Much of the prior work in signature authentication and in data hiding assumes that the host source is available. The host signal serves as a major noise source in the detection of the watermark. Hence, the knowledge of host signal characteristics will definitely enhance detection performance. In this chapter we propose an approach to signature recovery that does not require knowledge of the original host by using the discrete wavelet transform. This kind of watermarking is referred to in the literature as *blind watermark detection*. In this case, the interference from host image exists even when there is no noise from processing and other attacks. The most difficult problem associated with blind watermark detection in the frequency domain is to identify the coefficients that have been modified and the embedded watermark values. Since the non-marked image is no longer available at the decoder, it is impossible to determine the position of the coefficients that has been marked. To get around the problem, the mark is always inserted in a predefined set of coefficients. In this paper, we develop a blind watermark detection algorithm by selecting specific coefficients and replacing them with the channel coded indices of the watermark. The choice of the subbands for embedding hidden data is of

important significant and this will be explained in the next Section. Examples of methods that do not require the original host data for signature recovery include [Marvel et. al, 1998], [Piva et. al., 1998c], and [Pereira et. al., 1999]. The main contribution here is a technique that has the potential for embedding a significant amount of data, which can then be recovered without any additional knowledge of the host.

The proposed embedding and extracting methods utilize the discrete wavelet transform domain. The DWT has good energy compaction properties and it is playing an important role in upcoming compression standards such as JPEG2000 and MPEG-4. For this reason it has been used in our data embedding research.

The next section explains the embedding approach for the wavelet coefficients. Sections 6.1 and 6.2 discuss embedding and extraction techniques. Experimental results are given in section 6.3 and conclusions in section 6.4.

## **6.1 Embedding Technique**

We now focus on the issue of choice of subbands for embedding hidden data. In general, hiding data in the lower subbands has several advantages. The nature of current compression algorithms favors better preservation of the lower frequency data than high frequency data. Inserting information in the lower bands, therefore, does not lead to easy destruction of the hidden information or to any significant change in the coding efficiency. On the other hand, inserting data in the higher bands has the advantage that it does not degrade the host image quality significantly. Examples of such operations include low pass filtering for image

enhancement and JPEG lossy compression. A disadvantage, on the other hand, is that distortions to the host image introduced by embedding in the lower bands may be perceptually more severe than other bands.

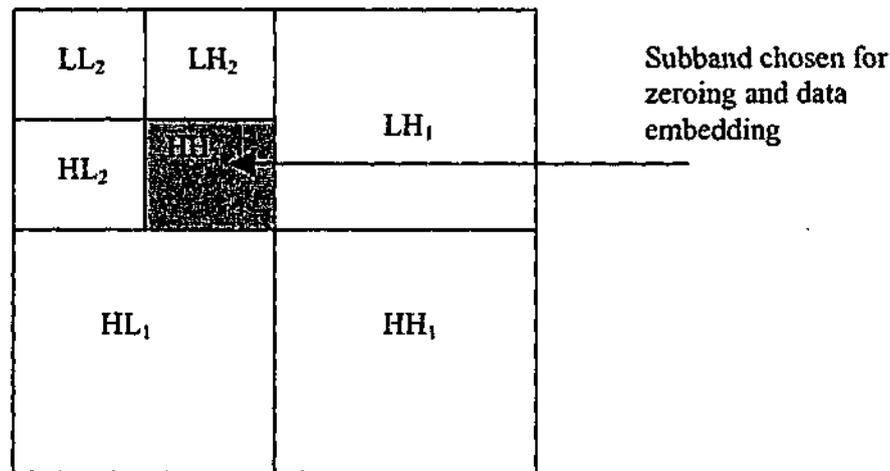
Moreover, as has been pointed earlier, the host image in data hiding serves as a major noise source in blind watermark detection, and the interference from host image exists even when there is no noise from processing and other attacks. In such a case marking only mid-band coefficients will reduce the interference from host image. This results from the observation that low band coefficients generally have much higher power than the mid-band coefficients. Therefore, in order to obtain a tradeoff between perceptual invisibility and robustness to image processing techniques, the lowest subband coefficients are skipped and the watermark is inserted into the intermediary frequency coefficients.

Similar proposals for blind watermarking have been found, in [Barni, et. al., 1998a], [Herrigel, et. al., 1998], and [Wu, et. al., 1999] for embedding in the DCT domain using spread spectrum watermarking, and in [Wang, et. al., 1998], and [Chae and Manjunath, 1999], for embedding in the wavelet domain. Like in [Barni, et. al., 1998a], our algorithm chooses the medium frequency range of the spectrum of the transform coefficients but unlike [Barni, et. al., 1998], our algorithm has the ability to hide large number of bits and casts the watermark in the DWT domain. This is due to the use of different techniques in the embedding process. For example, vector quantizer is used to compress the signature image, and channel codes to eliminate the errors introduced due to compression and noise attacks.

Further, noting that natural images typically have very low energy in the higher bands, we find that for most images, zeroing out some or all of the coefficients in one or more of the high subbands introduces a very low mean square error, and affects detail only hardly noticeable in the perceptual sense. Therefore, if the hidden data is embedded on the zeroed coefficients, the extraction process only needs to use the zero-vector as its estimated base for decoding the noisy vectors it receives. In practice, the exact coefficients that are zeroed out and subsequently used for embedding, are either predetermined, or selected in a pseudo-random manner using a key, or selected image-adaptively based on stable features of the host frame.

The scheme implements a two-stage wavelet transform decomposition of each image is made, and the hidden data is embedded in the coefficients of the middle subband, after zeroing. The DWT decomposition scheme used is shown in Figure 6.1. A two-stage Haar wavelet transform decomposition of host image is made, and the hidden data is embedded in the coefficients of the shaded  $HH_2$  subband, after zeroing. Figures 6.2 and 6.3 shows a schematic diagram for the embedding and extraction mechanism outlined above. The steps in embedding are:

- 1 The signature coefficients are vector quantized according to the method described in section 4.2. The quantized coefficients are encoded using channel codes similar to the ones described in Chapter 5.
- 2 The signature codes are then appropriately scaled by the factor  $\alpha$ .
- 3 The selected host coefficients are then replaced by the scaled signature codes and combined with the original (unaltered) DWT coefficients.



**Figure 6.1** Subband chosen for zeroing and data embedding.

4 The merged coefficients are then inverse transformed to give an embedded image.

The choice of the vector quantizer affects the quantity and quality of embedded data. Choice of the scale factor  $\alpha$  depends on the application. A large value of  $\alpha$  results in a more robust embedding at the cost of quality of the embedded image, i.e., there could be perceivable distortions in the embedded image. A smaller  $\alpha$  may result in poor quality recovered signature when there is a significant compression of the embedded image.

To increase the security of the embedding, an encryption key could be used to pseudo-randomly shuffle the coefficients in the subband chosen for embedding before grouping them into  $n$ -dimensional vectors. The encryption key based shuffling introduces an additional layer of security apart from the security enforced by the already immeasurable variability in the source and channel codebooks chosen. It is practically impossible for unauthorized persons who know the algorithm, to pirate the hidden information, without knowledge of the source

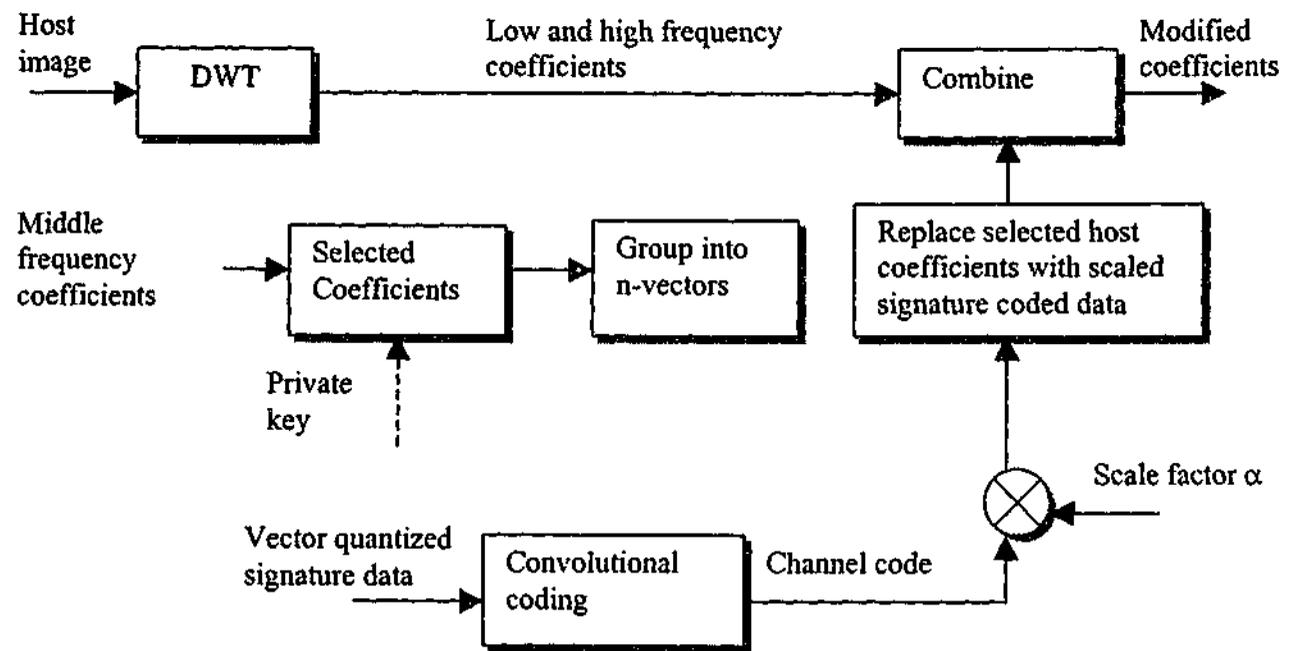


Figure 6.2 Schematic block diagram of the encoder.

codebook, the channel codebook, or the encryption key. In our experiments shown in section 6.4 however, we didn't implement the encryption key for simplicity reasons.

## 6.2 Extraction Technique

Figure 6.3 shows the block diagram of the decoder without the host image.

Signature extraction follows an inverse sequence of operations. The received embedded image is first DWT and the modified coefficients are identified. By appropriately scaling the coefficients corresponding to the signature data, the codes representing the signature data are recovered. To recover the original data a soft-decision decoding using Viterbi algorithm is implemented. Finally, the signature image is obtained from the source codebook of the vector quantizer.

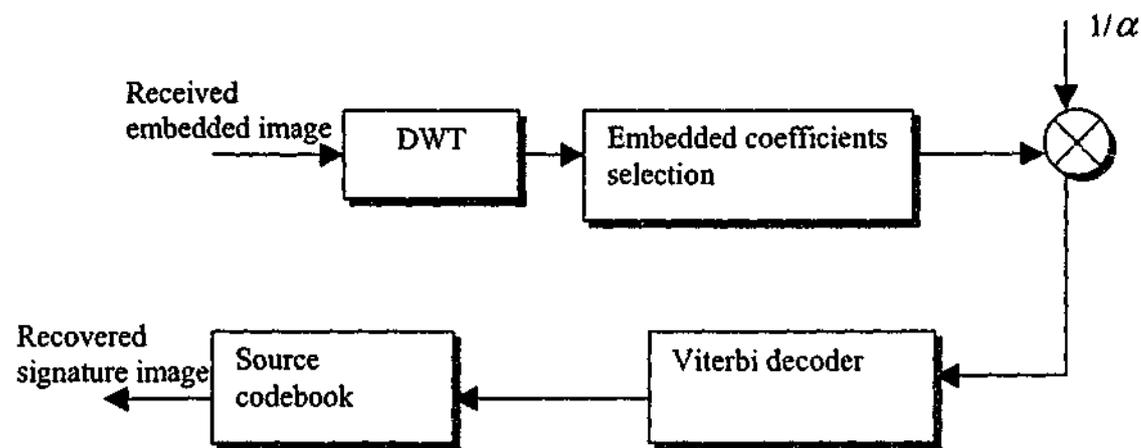


Figure 6.3 Block diagram of the signature extraction.

### 6.3 Watermark Detection

In several applications, such as copyright protection, we are interested in determining whether a given image contains a watermark. This is what we call the watermark detection problem. This should not be confused with the decoding of embedded information that we have analyzed in the previous chapters, since we are interested only in detecting the bare presence of a watermark in the image we are testing. Hence, the watermark detection problem can be expressed as a hypothesis test. Hypothesis testing is the task of deciding which one of two hypothesis  $H_1$  or  $H_0$  is the true explanation for an observed measurement. The two possible hypothesis, namely “the image contains a watermark” ( $H_1$ ) and “the image does not contain a watermark” ( $H_0$ ). In other words, there are two possible probability distributions associated with each of the two hypothesis, denoted by  $f_y(y|H_1)$  and  $f_y(y|H_0)$  respectively. Accepting hypothesis  $H_1$  when  $H_0$  actually is true is called a type I error. Accepting hypothesis  $H_0$  when  $H_1$  actually is true is called type II error. The probability of these two events should be kept as small as possible. A

method of finding the optimum decision of the detector corresponds to the Neyman-Pearson rule [Blahut, 1987], the decision rule is specified in terms of a threshold parameter in which  $H_1$  is decided if

$$\frac{f_y(y|H_1)}{f_y(y|H_0)} > \eta \quad (6.1)$$

Where  $\eta$  is a decision threshold,  $y$  is the watermarked image, and  $f_y(\cdot)$  is the pdf of  $y$ , otherwise  $H_0$  is decided. The pdf in the numerator corresponds to the statistical distribution of the image under test when it contains a valid watermark, whereas the pdf in the denominator corresponds to the statistical distribution when no watermark is present [Cachin, 1998].

In the DWT domain we can use the generalized Gaussian statistical model in Equation (6.1). In this detection test we are not interested in extracting any information that the watermark might carry, as it was the case in previous sections.

In some cases, especially when error correction coding is used, the decision rule in Equation (6.1) can be difficult to implement. However, suboptimal detectors can be used. For instance, a suboptimal decision can be made in two steps. First, a ML decoder obtains an estimate of the message carried by the watermark. Then a hypothesis test similar to that in Equation (6.1) is applied.

#### 6.4 Implementation and Experimental Results

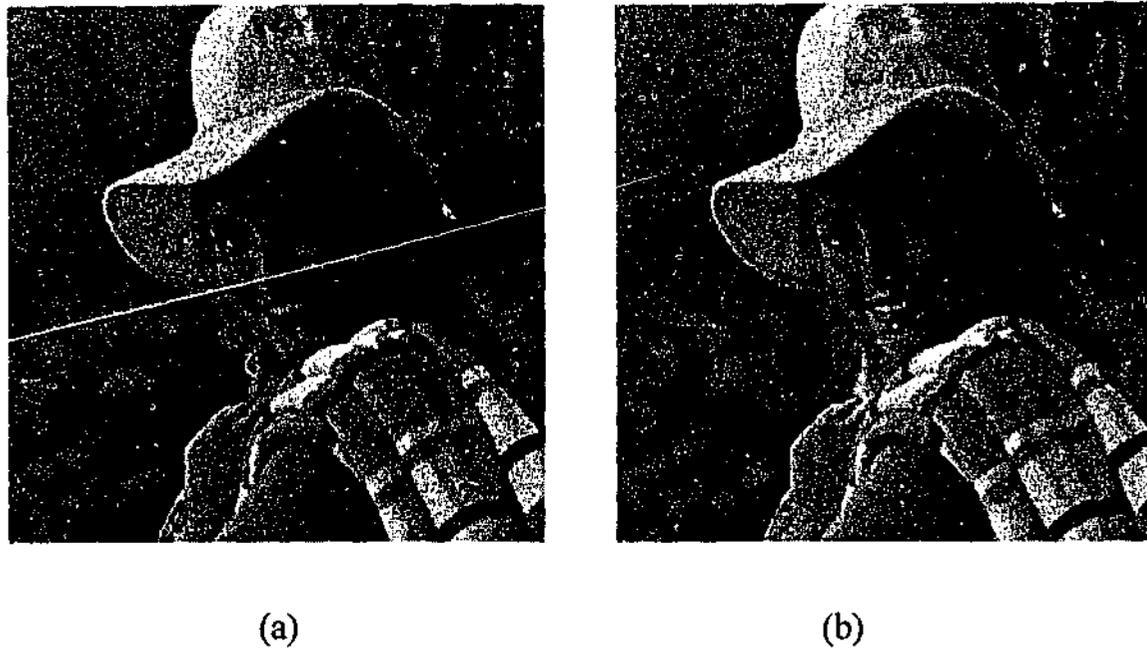
This section provides simulation results to demonstrate the performance of the proposed technique. We specifically make use of the Haar wavelet transform for all simulation. We took the gray image of *Elaine* of size 512 x 512 and watermarked it

with two watermark images of *Bear* and *Bird*, both of size 128 x 128. The watermark is inserted into the DWT of the host image as explained in Section 6.1. A fixed scale factor  $\alpha$  of 10 was used throughout the experiments.

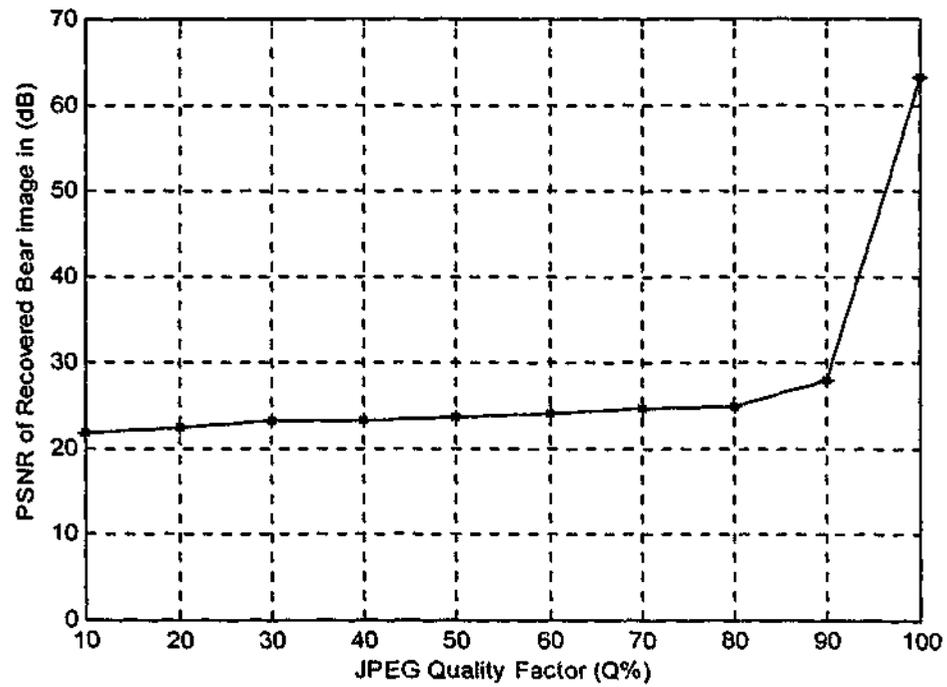
The resulting watermarked signal is distorted using JPEG compression and noise addition. The watermark was then extracted without knowledge of the host image and compared with the original watermark to measure robustness and detection capability of the technique.

The embedding was implemented using vector quantizer to compress the signature images of *Bear* and *Bird* to 1/4 its original size. The indices obtained are then channel coded using convolutional code of rate 2/3 before embedding into the host DWT coefficients. The host image of *Elaine* was transformed to the wavelet domain using 2 levels of Haar wavelet transform and the coefficients of the subband  $HH_2$  are zeroed to allow the insertion of the channel coded signature symbols.

Figure 6.4 shows the host image of *Elaine* and the watermarked image of *Elaine* with a signature gray image of *Bear*. This embedding results in one-sixteenth-size signature image. The PSNR of the watermarked *Elaine* image is 36.81 dB. Figure 6.5 shows the PSNR of the recovered *Bear* image for different JPEG compression. Here the PSNR of the extracted *Bear* image is compared to the original watermark image of *Bear*, higher values would be obtained if it has been compared to the compressed one. Figure 6.6 displays the embedded image of *Elaine* with *Bird* image together with the recovered signature images of *Bird* for different JPEG quality factor compression. It is clear from this figure that even at high JPEG



**Figure 6.4** Original and watermarked images of *Elaine*, (a) original image of size 512 x 512, (b) watermarked image with *Bear* image as watermark.



**Figure 6.5** PSNR of recovered *Bear* image watermark for different JPEG compression. The host is *Elaine* image.

compression ratio (low quality factor  $Q\%$ ) one can recover acceptable quality for the signature image. The images in this figure are shown at a reduced size to 60% of the actual size. The PSNR values shown in this figure were obtained with comparison to the compressed *Bird* image. To check for the presence of watermark in an image, Equation (4.15) is used to calculate the similarity measure between the extracted watermark and original watermark for watermarked and unwatermarked image. The result is shown in Figure 6.7 for varying JPEG compression for the *Bird* image as watermark. As can be seen from this graph, it is easy to find a threshold (e.g. 0.4) for signature detection between unwatermarked and watermarked images. Figure 6.8 shows the bit error as a function of JPEG compression for soft and hard-decision decoding together with the uncoded case. It is obvious that soft-decision decoding superior to hard decoding, nevertheless, the hard decoding is still better than uncoded signature data.

## 6.5 Summary

In summary, we have proposed a robust data hiding technique for embedding images in images. A key component of this scheme is the use of channel codes for encoding the signature image coefficients before inserting them into the host image wavelet coefficients. The hidden data can be recovered without the original host image. Experimental results show that this method is robust to lossy image compression. Similar results can be obtained for concatenated and turbo codes. However, only convolutional codes is shown in this Chapter.



JPEG = 90%  
PSNR = 56.36 dB



JPEG = 70%  
PSNR = 33.61 dB

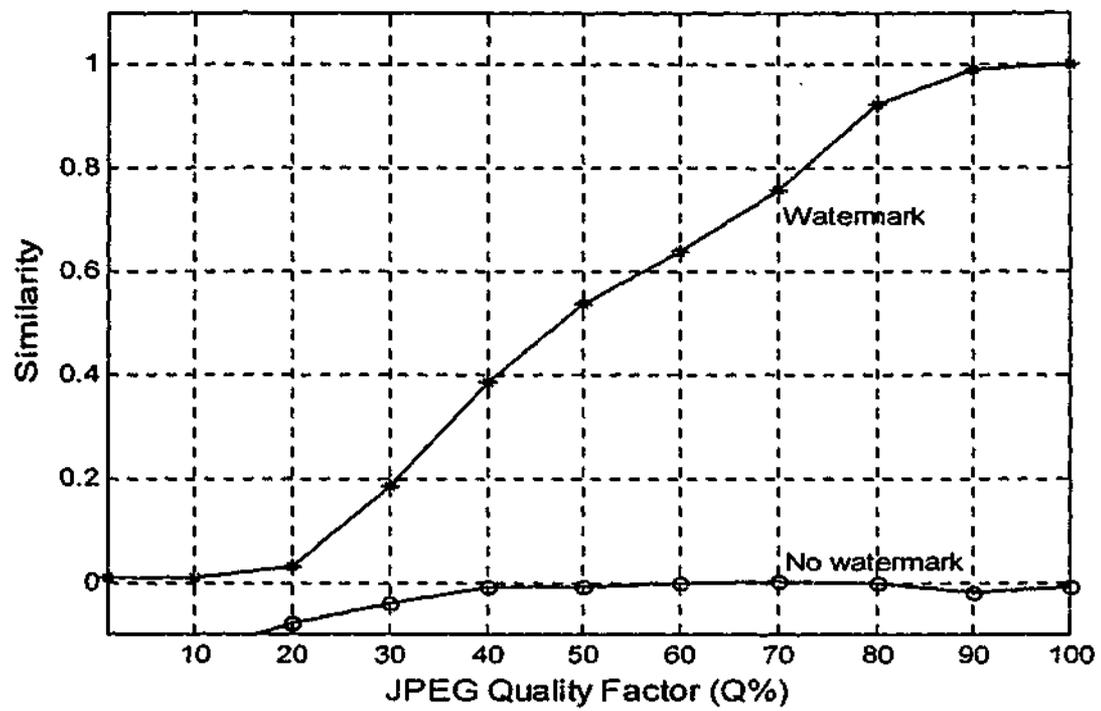


JPEG=50%  
PSNR=31.78 dB

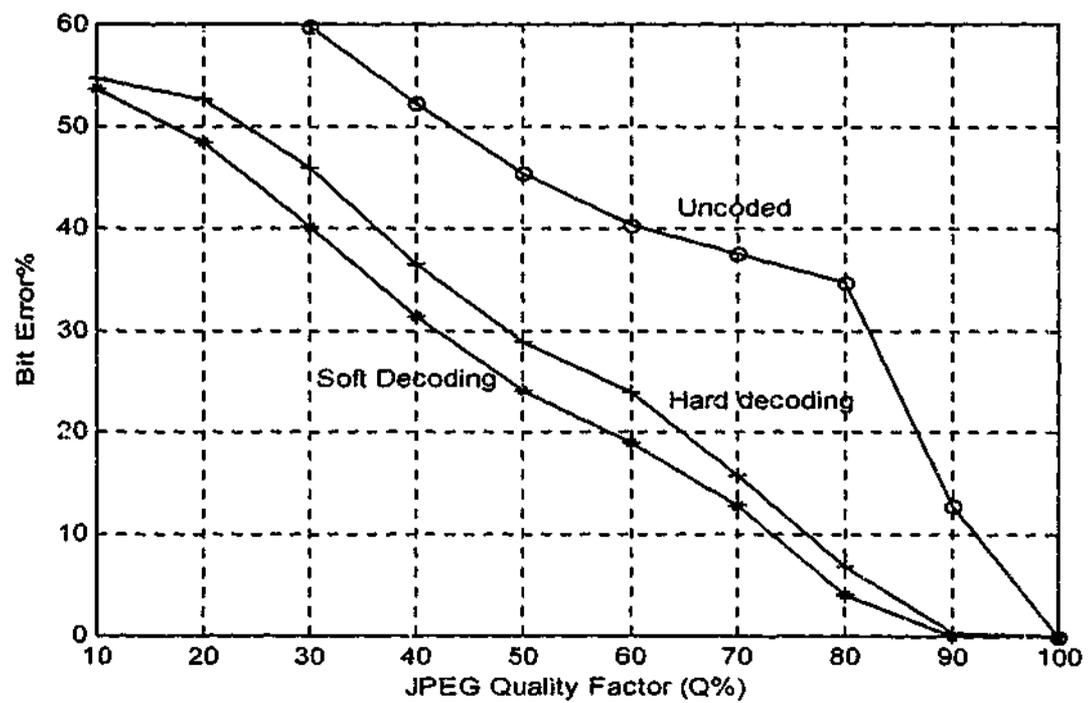


Watermarked *Elaine* image(512x512) with *Bird* image as the watermark.

Figure 6.6 Watermarked *Elaine* image and recovered *Bird* image for different JPEG compression.



**Figure 6.7** Similarity measure of watermarked and unwatermarked *Elaine* image with *Bird* image as the watermark.



**Figure 6.8** Bit error versus JPEG compression for host *Elaine* image and *Bird* image as the watermark image for hard and soft decoding together with uncoded watermark.

## **Chapter 7**

### **Data Embedding in Video**

#### **7.1 Established Work**

#### **7.2 Images in Video**

#### **7.3 Proposed Technique**

#### **7.4 Implementation and Experimental Results**

#### **7.5 Summary**

## Data Embedding in Video

---

This chapter presents an approach for *video data embedding* based on discrete wavelet transform and trellis coding. We address the problem of embedding logos or still images within a host video. The proposed method is an extension to our algorithm in Chapter 6 for data extracting without the original host. The algorithm starts with transforming each frame of the video sequence into the discrete wavelet domain where their coefficients are grouped into vectors equivalent to the codeword length of the signature data. The host video is not necessary for signature extraction. The method repeatedly merges the signature coefficients at various frames of the host video which provides frequency spread of the watermark to provide robustness against widely varying signal distortions including compression and filtering.

### 7.1 Established Work

Previous work on watermarking includes watermarking of still images, audio, and multimedia data in general, has been covered intensively in Chapter 2. While much of the initial work was on watermarking image data, several methods have been

proposed lately for embedding audio and video information into video sequences. One of the earliest work on video watermarking was proposed by Hartung and Girod [Hartung and Girod, 1997]. They provide a watermarking scheme specifically for MPEG encoded video. The basic idea is to add the spread sequence watermark to the line-scanned digital video signal. The watermarking embedding process is slow and it can only handle data at rate of several bytes per second. Another work on video watermarking was proposed by Swanson et al. [Swanson et al., 1997a], and [Swanson et al., 1997b], where they designed a data hiding algorithm to embed watermark data into video. The message data is embedded in the DCT domain, by modifying the projections of the 8x8 host block DCT coefficients onto a pseudo-random direction. The data hiding rate is two bits per 8x8 blocks. The authors demonstrate robustness to additive Gaussian noise and motion JPEG compression. Not long ago, Zhu et al. [Zhu et al., 1998] presented an approach to digital watermarking of images and video based on the discrete wavelet transform. Their watermarking framework embeds the data in the highpass wavelet coefficients of the host video. Such an algorithm is not quite robust against compression and filtering attacks. More recently, [Mukherjee et al. 1998] presents a technique for hiding audio in video. They use multidimensional lattice structures to embed the 8KHz speech signal, and the data hiding rate is about 1%. However, the demodulator for lattice code is too complicated to be implemented in practice [Johannesson, and Zigangirov, 1999].

Meng and Chang [Meng and Chang, 1998] adopted a similar approach to Hartung and Girod of watermarking MPEG- video in the bitstream but of visible watermark.

More recently, [Wu and Yu, 2000] spreads the watermark over the video frames. Spreading the watermark in this way may not be desirable because when someone takes just few frames from the watermarked video (frame dropping), the watermark is no longer retrievable.

Our work differs from most of the above mentioned algorithms in a way that the watermark is hidden in the uncompressed raw multimedia or video data. We believe that such embedding may work better than embedding in the compressed bitstream of the video because:

- The watermarks are always integrated in with the host data.
- It is the original host video that one wishes to copy-right protect, not the compressed version.
- There are many spaces in the multimedia or the video data to embed the watermarks without degrading the quality too much (or even make the watermarks visible)

Moreover, for complete verification there is no advantage to use the watermarking method in a compressed video data. Compression standards, e.g., MPEG or JPEG, have user-defined sections where digital signature can be placed. Because multimedia data are stored or distributed in the file format instead of pixel values, therefore, once the multimedia data is modified, the user-defined section of the original data is usually discarded by the editing software. Further, because there is less space for compressed video to hide watermarks, if we do not want to sacrifice too much visual quality on the video data, there may not be enough information bits to protect the data.

In this chapter, we describe a data hiding technique and demonstrate its robustness to MPEG coding of the embedded video.

## 7.2 Images in Video

While the discussion in the previous chapters has been considerably generic and applies to hiding most kinds of secure data in most kinds of host, in this chapter we modify the scheme to hiding still image in a *video host*.

The general principle of data hiding in video is as follows. Each frame of a video sequence is wavelet transformed, and the transform coefficients are grouped into vectors. The signature data is vector quantized and the indices are embedded into the coefficient vectors in one or more subbands using efficient channel codes. The same hidden data may be repeated in a few successive frames to introduce robustness to frame deletion, duplication, blending and other temporal attacks. Moreover, embedding the same watermark in each frame will reduce the detection complexity due to video data accumulation before actual detection.

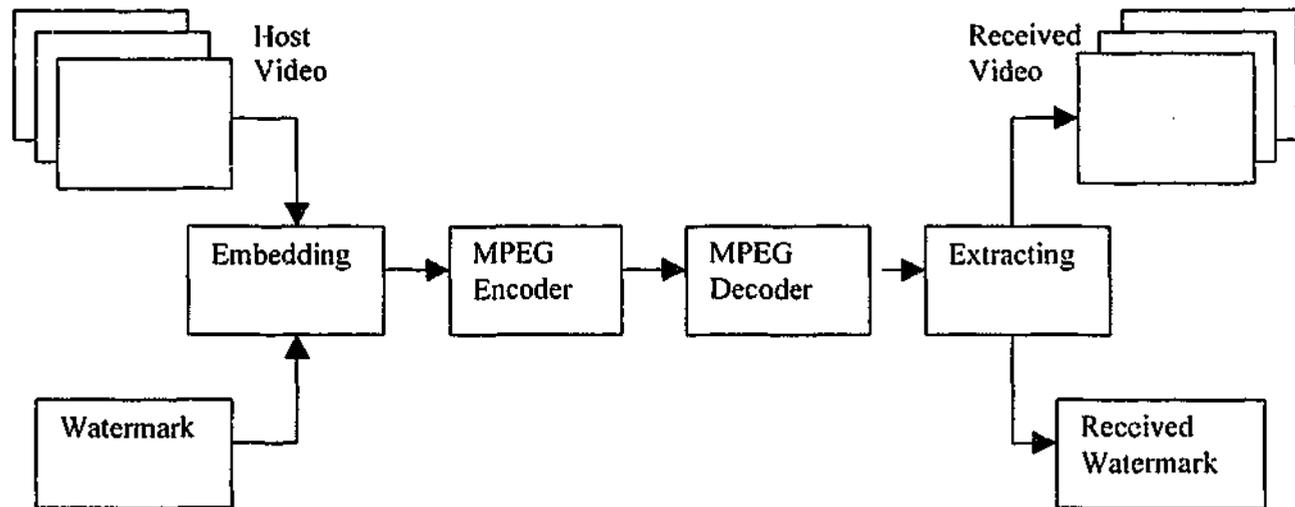
Embedding images in video provides a number of multimedia uses including verification and identification of a given data stream. For the purpose of this investigation, we will focus on the topic of embedding images in video. This process can be broken down into an image-in-image embedding process, as a video stream, prior to compression, is simply a sequence of still images. The topic of digital watermarking bears resemblance to image embedding where copyright protection information is contained within the given image. Typically, those methods yield watermarks that are on the order of 1% of the original image,

whereas image embedding has the potential to store much higher amounts of data. Although data embedding provides a sizable increase in the information embedded, we still wish for the algorithm to be robust to various errors and perturbations that the data will encounter, including lossy compression techniques. Since a video sequence can be broken down into a series of still images, we could use the same robust technique for image-in-image embedding.

The watermark bits embedded using the method mentioned above can represent anything: copyright messages, serial numbers, plain text, control signals, etc. the contents represented by these bits can be compressed and encrypted. In some cases it may be useful to embed a small logo instead of a bit string as a watermark [Langelaar et. al., 2000]. In this work we concentrate on embedding logos or small images into the video frames. Moreover, the use of logo-type watermarks will improve trustworthiness of watermarking in the eyes of non-technical jurors since the mark can be readily displayed and understood [Braudaway, 1997].

### 7.3 Proposed Technique

A schematic of our embedding scheme is shown in Figure 7.1. The embedding is done in the discrete wavelet domain. The motivation of using the wavelet domain opposed to spatial or DCT domains to embed the watermark is because it provides both a simultaneous spatial localization and a frequency spread of the watermark within the host video. The frames of the host video sequence are first wavelet transformed. The signature image is vector quantized and the indices are then



**Figure 7.1** Schematic of video embedding technique.

encoded using channel codes before inserting them into a vector shaped wavelet coefficients of the host video frames. The corresponding inverse transform is applied to form the embedded or watermarked video. More details of the embedding algorithm have been described in Chapters 3-6. The embedded video frames are then MPEG compressed, and the signature image is recovered from the lossy compressed video. Experimental results are given in this chapter.

## 7.4 Implementation and Experimental Results

In this section we consider hiding a still image in a video sequence using multilevel DWT and convolutional coding (See Chapter 5). Figure 7.2 (a), shows one frame of *Walter* video of size 256 x 256 pixels and Figure 7.2 (b), shows the signature image *Logo* of size 64 x 64, which is about 1/16-th the size of the video frame.

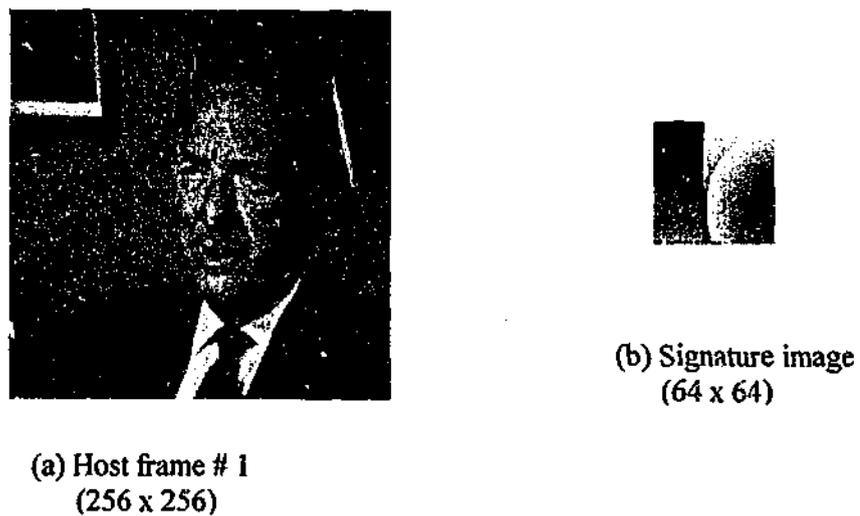


Figure 7.2 Test data. (a) Host video frame # 1. (b) A signature image.

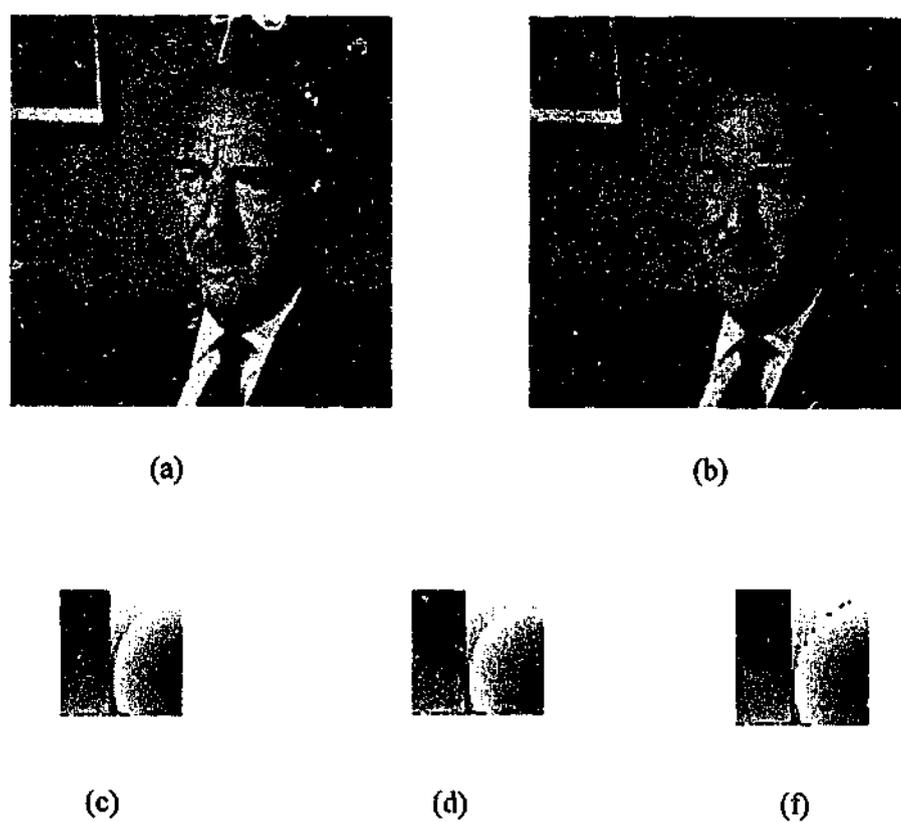


Figure 7.3 (a) Watermarked frame # 1. (b) Frame # 1 after MPEG encoding at a rate 500 k bits/seconds. (c) Recovered *Logo* image. (d) *Logo* image recovered from Frame # 5 (B frame). (f) *Logo* image recovered from Frame # 4 (P frame).

The signature image is first compressed using vector quantizer and the indices obtained are channel coded. Convolutional encoder of rate  $2/3$  is used in this test with the transfer function shown in Figure 5.11. The coded bits are multiplied by a scale factor  $\alpha$  equal 10 before adding it to the DWT coefficients of the host frame. This embedding is repeated in each of the frames of the video. Then the embedded frames are compressed using MPEG-1 with frame pattern set to [IBBPBBPBBPBB] at 500 kbps. Figure 7.3 (a) shows the frame after embedding and Figure (b) is the result after the MPEG encoding.

To recover the watermark image from the compressed video, the video sequence is first MPEG decoded and each frame is subjected to watermark extraction process similar to the one described in Chapter 6 for image in image decoding. Figure (c) shows the recovered *Logo* image. It can be seen that, although the video frame has been heavily distorted, the logo can still be recognized. Moreover, since I-frames in the MPEG-1 sequence are JPEG compressed, the results would be similar to image-to-image embedding described in the previous chapters. The frame #5 shown is a B frame. The reconstructed signature from this B frame is shown in figure (d). The reconstructed signature images from two P frames are shown in figure (e) and figure (f). In general, it appears that the method works reasonably well for hiding still images in video. In the same way, the algorithm can be extended to hide video signature into a video host. Notice the difference in quality between the I-frame embedded and P-frame embedded blocks. For example, the top-left block of the figure is of good quality because this block is embedded directly in an I-frame in the MPEG sequence. Blocks recovered from P- or B-

frames show some amount of visible noise in the reconstructed image. In this experiment, the MPEG bit rate was 500 k bits/second at 30 frames/second.

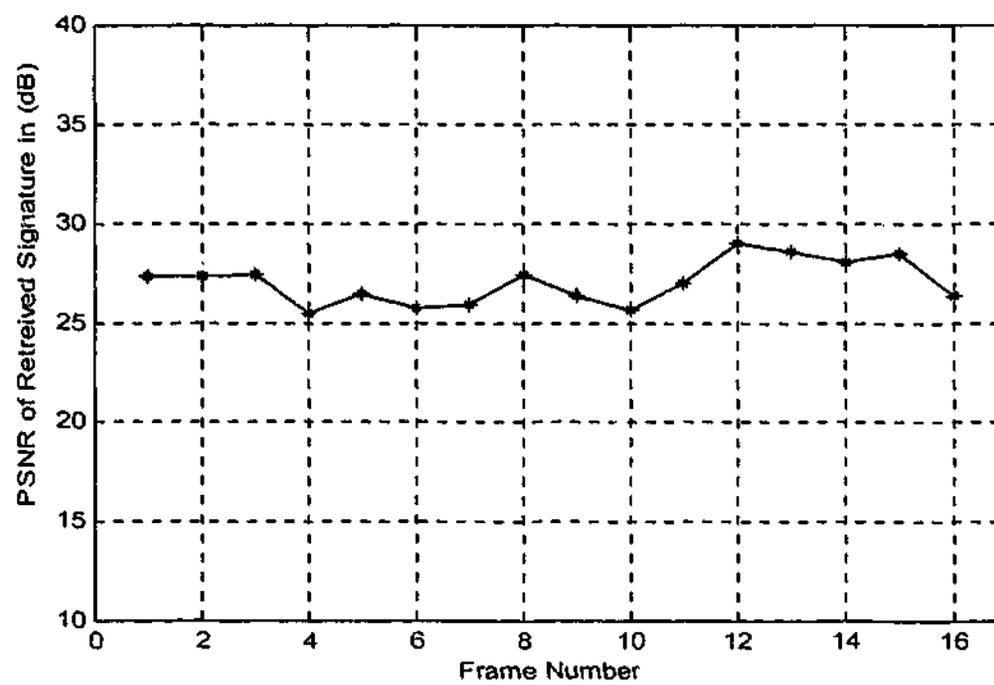
Figure 7.4- (a) shows the PSNR of the retrieved watermark image *Logo* watermarked frames for the first 16 frames of the video for a video rate of 500 kbits/second, (b) is for video rate 750 kbits/second. Figure 7.5 shows the PSNR of the embedded video frames. From Figures 7.4 and 7.5 one can see that the pattern structure I, B, P is not reflected in these results. In other words, the PSNR measure does not show it while human visual perception does. This is another indictment against PSNR as a measure of distortion.

Finally, Figure 7.6 shows Average PSNR of the recovered watermark as a function of compressed video bit rate for video sequence of *Walter* and *logo* image as watermark at 30 frames /second, while Figure 7.7 shows the percentage of bit error rate due to MPEG compression for *Walter* sequence and *Logo* image.

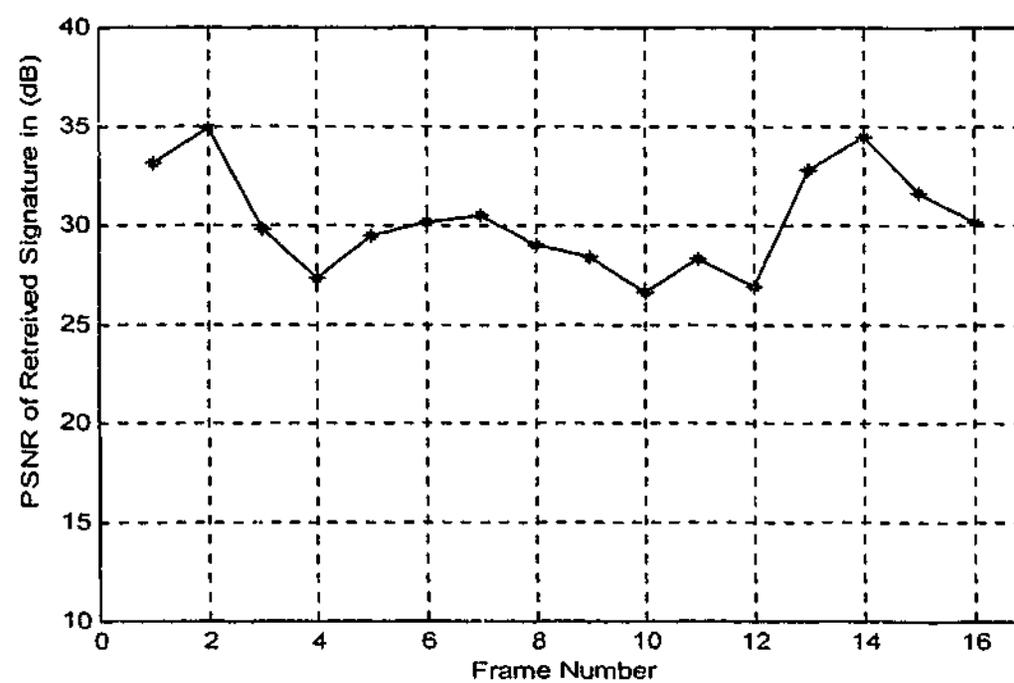
## 7.5 Summary

In conclusion, we note that the methods developed in Chapters 3-5 enable new application domain in multimedia processing. In addition, the embedded watermark is robust to a reasonable level of signal distortion even though the host video was not available for extraction. Moreover, the algorithm is portable to different applications and can hide different types of information robustly within a host signal. This ability has potential applications to multimedia data control and in the emerging MPEG standards such as MPEG-4 and MPEG-7. The easy of access to and manipulation of content makes it all the more important that the multimedia

content be protected from unauthorized uses. MPEG has started working on a new standard for multimedia that enables content to be searched for and delivered based



(a)

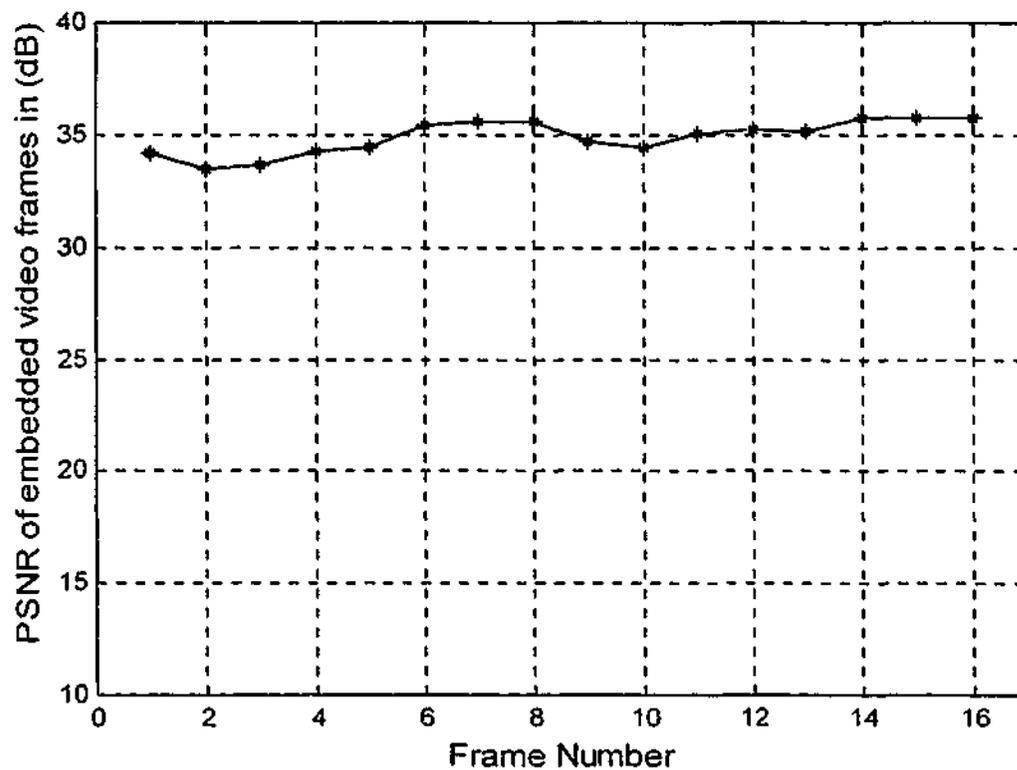


(b)

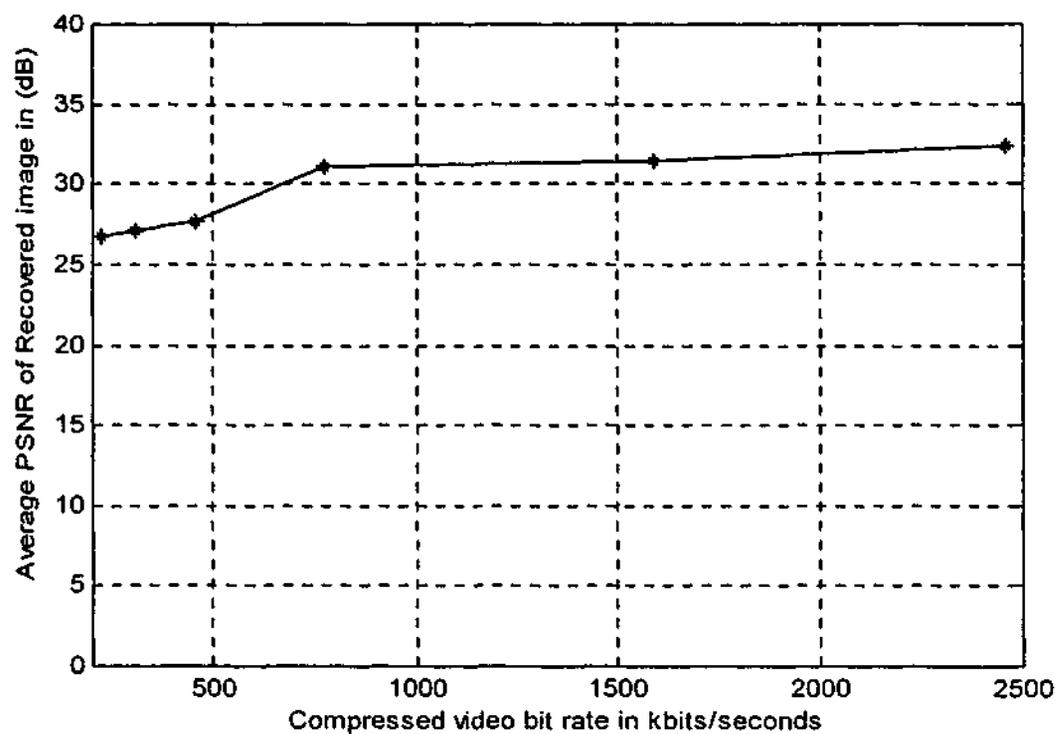
**Figure 7.4** PSNR of retrieved signature image from compressed video sequence at (a) video rate = 500 k bits/sec for 30 frames/sec and (b) video rate=750 k bits/sec.

on usage rights, and data hiding technologies such as the one described in this thesis are very useful for such applications.

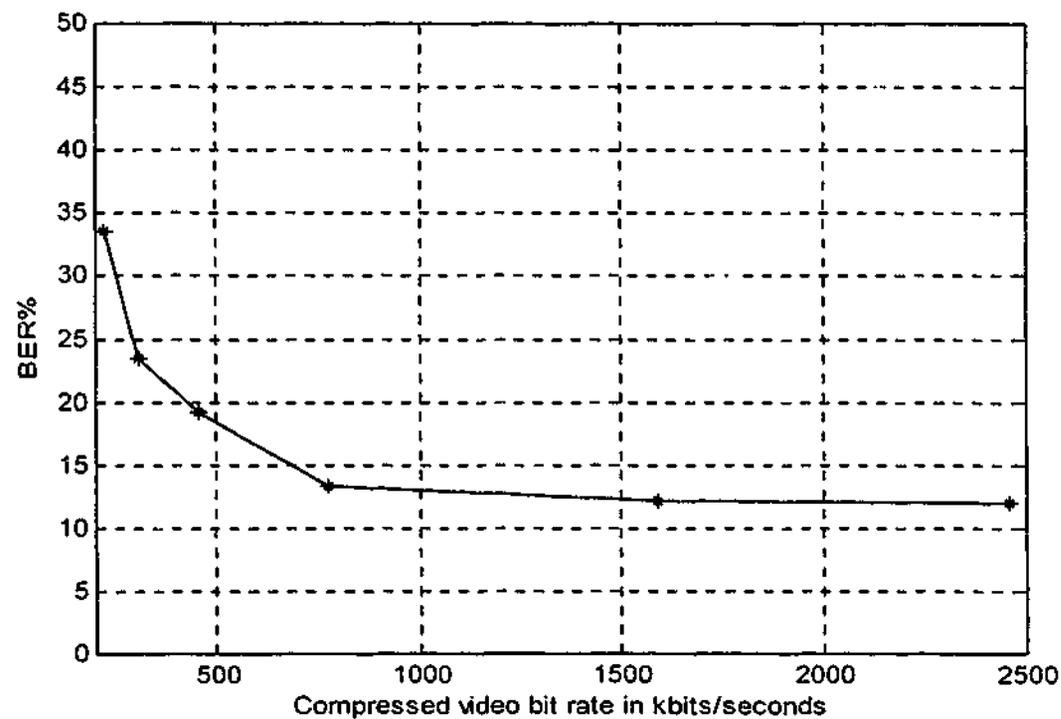
The disadvantage of this algorithm is the excessive computations needed to embed every frame of the video sequence, especially when the video sequence is of considerable length. However, this is a problem only when the embedding process performed in real time, and even this is not a big issue considering the fast improvement in computer technology.



**Figure 7.5** PSNR of embedded video frames of Walter sequence with Logo as watermark at video rate = 500 k bits/sec and 30 frames/sec.



**Figure 7.6** Average PSNR of the recovered watermark as a function of compressed video bit rate for video sequence of *Walter* and *Logo* image as watermark at 30 frames /second.



**Figure 7.7** Bit error of the recovered watermark as a function of compressed video bit rate for *Walter* sequence and *Logo* at 30 frames /second.

## **C h a p t e r 8**

### **Conclusions and Future Work**

#### **8.1 Conclusions**

#### **8.2 Summary of Contributions**

#### **8.3 Directions for Further Research**

## Conclusions and Future Work

---

### 8.1 Conclusions

In this research we have developed several new techniques for robust data embedding of image and video data. These techniques enable embedding large amounts of data and facilitate signature recovery in the absence of the original host. The proposed methods have given an overview of the advantages that channel coding brings about in data embedding applications. The channel codes used in this research comprises of block codes such as BCH and Reed-Solomon, the other codes used were the convolutional code, concatenated codes, and turbo codes. Several interesting applications of these embedding methods to lossless text data recovery from lossy compressed image, and images in video hiding, were presented in Chapters 4-7.

## 8.2 Summary of Contributions

One of the main contributions of this thesis is the development of methodologies for large quantity data embedding. As the result demonstrates, the signature recovery is quite robust to JPEG compression and noise addition. Since these techniques enable large quantity data embedding, one can achieve robustness to image manipulations for watermarking related applications by having redundancy in the signature data.

In Chapter 4 we presented the error-correcting coding technique together with DWT and VQ, that distributes the message data in the wavelet sub-bands. Using this method, one can embed images, which are up to 25% of the host data size. This method is quite robust to JPEG compression and noise addition. Moreover, we demonstrate applications using the previous method for lossless text data hiding. We consider the example of hiding a text message into an image and the message is recovered without error even when the embedded image is lossy compressed.

Chapter 5 extends the wavelet-based embedding technique by implementing convolutional code for error-corrections. Concatenated codes and turbo codes add error resilience to signature recovery. Trellis decoding is implemented using the maximum likelihood decoding algorithm together with the maximum a posteriori (MAP) probability algorithm and soft output Viterbi algorithm (SOVA)

In Chapter 6, we present an algorithm, which combines trellis codes and wavelet transform. In contrast to the previous methods in Chapter 4 and 5, this method does not require the original host to recover the hidden data. Methods that do not require

the original host are very desirable in applications such as hidden communications.

This method is again shown to be quite robust to JPEG compression.

In Chapter 7, we demonstrate that this method is robust to motion compensated coding. This is illustrated by hiding images in video, which is MPEG compressed.

In summary, the methods presented in this dissertation advance the current data hiding technology both in terms of the quantity of the data that can be hidden (up to 25% compared to 1% reported in the literature), and the quality of the embedded and recovered data even under significant JPEG/MPEG compression (of up to 90% in some case). Analysis of the different coding schemes reveals the superior performance of convolutional codes for a reasonable complexity. Comparison with the uncoded case shows gains of about 5 dB for simple codes. Note that increase in coding gain allows increasing the number of hidden information bits for the same probability of error.

Finally, as the work presented here is general, it can be extended to other types of host and signature data.

## **8.3 Directions for Further Research**

### **8.3.1 Robustness to Signal Manipulation**

The primary emphasis in this dissertation is on large quantity data embedding that is robust to data compression. There is a trade-off between the quantity of the data that can be embedded and the robustness of the hidden data to signal processing. In

digital watermarking for authentication, intentional or unintentional attacks may include, in addition to signal compression, scaling, cropping, rotation of images, and digital-to-analog and analog-to-digital conversions. No single technique can be resistant to all these attacks simultaneously. However, the methods proposed in this dissertation have the advantage of embedding large amounts of data. Since watermarks typically require very few bits compared to the host data size, one may be able to distribute these bits intelligently so that the embedded data is resistant to specific attacks than compression. This needs further investigations.

### **8.3.2 Information-Theoretic Model**

In this thesis, we have illustrated basic similarities and differences between watermarking and traditional communications. The host signal has been viewed as a form of noise, and signature or watermark signals treated as transmissions with very low signal-noise-ratios. However, observation knowledge of the host data as side information at the transmitter allows the design of more powerful data embedding algorithms [Moulin, et. al., 2000]. In order to utilize this communication model in the design of watermark insertion algorithm, it is necessary to have knowledge of the underlying statistics of the content and the distortion it is likely to experience. In particular, it could be possible to calculate the robustness of watermarked data to subsequent attacks, and to maximize robustness within a specified distortive constraint.

While these observations promise an enhancement in the performance, there is still much room for future work. This might include further improvement by optimizing

the encoder to adapt itself to the host signal and uses it to his advantage by, choosing codewords in the direction of the host signal [Costa, 1983]. Certainly the use of error correcting codes that has been implemented in this thesis will be an added benefit to such models.

### 8.3.3 Applications

Large quantity data embedding and lossless recovery of hidden data open up a domain of new applications, including image/video quality control, and embedding control information in multimedia data.

Other potential applications include multimedia databases where the objects in the data base "contain" self-information that can be used in navigating the database, or in providing different levels of access to the users depending on the service that is requested. For example, object based representations (using region masks) could be embedded into the video stream that would enable object based functionality using existing video data formats such as MPEG-2. While many standard bit streams allow for header information where such control data bits could be stored, embedding the control data in the host data stream has the advantage that it can not be accidentally or otherwise stripped off the host data. Recently, the MPEG has started working on developing a new standard MPEG-21 "Multimedia Framework". The scope of the standard can be described as the integration of two critical technologies: how consumers can search for and get content by themselves or through the use of intelligent agents and how content can be

decoded for consumption according to usage rights associated with the content (quoted from a web paper by Leonardo Chiariglione, Convener, MPEG; ([http://www.telecomitalia.com/threports\\_e.htm](http://www.telecomitalia.com/threports_e.htm)). We believe that the technologies developed in this dissertation would enable such applications.

## References

[Abdulaziz and Pang, 2000a]

N. K. Abdulaziz, and K. K. Pang, "Performance Evaluation of Data Hiding System using Wavelet Transform and Error-Control Coding," Proceedings of IEEE International Conference on Image Processing 2000 (ICIP' 2000), 10-13 September 2000, Vancouver, Canada.

[Abdulaziz and Pang, 2000b]

N. K. Abdulaziz, and K. K. Pang, "Robust Data Hiding for Images," Proceedings of the International Conference on Communications Technology (ICCT 2000), World Computer Congress WCC 2000, 21-25 August 2000, Beijing, China, vol. 1, pp. 380-383.

[Abdulaziz and Pang, 2000c]

"Source and Channel Coding Approach for Data Hiding," Proceedings of SPIE, Visual Communications and Image Processing 2000 Conference, 20-23 June 2000, Perth, Australia, pp. 1526-1535.

[Abdulaziz and Pang, 2001]

N. K. Abdulaziz, and K. K. Pang, "Data Embedding using Trellis Coding," International Conference on Communication, Computer and Power (ICCCP'01), February 12-14, 2001, Muscat, Sultanate of Oman., pp. 228-232.

[Akansu and Smith, 1996]

A. N. Akansu, and M. J. T. Smith, Editors, "Subband and wavelet transforms: Design and application", Kluwers Academic Publications, Boston, 1996.

[Anderson and Petitcolas, 1998]

R. J. Anderson, and F. A. P. Petitcolas, "On The Limit of Steganography", IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection, vol. 16, no. 4, May 1998, pp. 474-481.

[Antonini et. al., 1992]

M. Antonini, M. Barlaud, P. Mathieu, and I. Daubeshies, "Image coding using the wavelet transform," IEEE transactions on Image Processing, vol. 1, 1992, pp. 205-220.

[Bahl, et. al., 1974]

L. Bahl, S. Cocke, F. Jeinken, and J. Raviv, "Optimal Decoding of Linear Codes for Minimizing Symbol Error rate," IEEE Transactions on Information Theory, vol. 20, March 1974, pp. 248-287.

[Barlaud, 1994]

M. barlaud, Wavelets in Image Communications, Elsevier Science B.V., The Netherlands, 1994.

[Barnett and Pearson, 1998]

R. Barnett, and D. Pearson, "Frequency mode LR attack operator for digitally watermarked images", Electronics Letters vol. 34, no. 19, Sep. 1998, pp. 1837-1839.

[Barnett, 1999]

R. Barnett, "Digital Watermarking: Applications, Techniques and Challenges," Electronics and Communication Engineering Journal, vol. 11, no. 4, August 1999, pp. 173-183.

[Barni et. al., 1998a]

M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-Domain System for Robust Image Watermarking", Signal Processing (Special Issue on Watermarking), vol. 66, no. 3, 1998, pp. 357-372.

[Barni et. al., 1998b]

M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "Copyright Protection of Digital Images by Embedded Unperceivable Marks", Image and Vision Computing, vol. 16, no. 12-13, 1998, pp. 897-906.

[Barni et. al., 1999]

M. Barni, F. Bartolini, V. Cappellini, A. Lippi, and A. Piva, "A DWT-based algorithm for spatio-frequency masking of digital signaturs", Proceedings of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging'99, San Jose, CA, Jan 25-27 1999, vol. 3657.

[Bas, et. al., 1998]

P. Bas, J-M. Chassery, and F. Davoine, "Using the Fractal Code to Watermark images," Proceedings of the IEEE International Conference on Image Processing 1998 (ICIP98), October 4-7, 1998, Chicago, Illinois, USA, pp. 469-473.

[Bateman, 1999]

A. Bateman, Digital Communications: Design for the Real World, Addison-Wesley, Essex, England, 1999.

[Bender et. al., 1995]

W. Bender, D. Gruhl, and N. Morimoto, "Techniques for Data Hiding", Proceedings of the SPIE Conference on Storage and Retrieval for Image and Video Databases III, vol. 2420, San Jose, CA, February 1995, pp. 164-173.

[Bender et. al., 1996]

W. Bender, D. Gruhl, N. Morimoto, and A. Li, "Techniques for Data Hiding," IBM Syst. Journal, vol. 35, no.3 and 4, 1996, pp. 313-336.

[Berrou, et. al., 1996]

C. Berrou, A. Glavieux, and P. Thitimajshima, "Near Shannon Limit error-Correcting Coding and Decoding: Turbo Code," IEEE Transactions on Communication, October 1996.

[Biglieri and Luise, 1992]

E. Biglieri, and M. Luise (Editors), Coded modulation and Bandwidth-Efficient Transmission, Elsevier Science Publishers B. V., The Netherlands, 1992.

[Biglieri et. al., 1991]

E. Biglieri, D. Divsalar, P. J. McLane, and M. K. Symon, Introduction to Trellis-Coded Modulation with Applications, Macmillan publishing company, NY, USA, 1991.

[Blahut, 1984]

R. Blahut: Theory and Practice of Error Control Codes. Addison Wesley Publishing, Reading, MA, USA, 1984.

[Blahut, 1987]

R. Blahut, Principles and Practice of Information Theory, Addison-Wesley Publishing Company, USA, 1987.

[Blahut, 1990]

R. Blahut, Digital Transmission of Information, Addison-Wesley publishing Company, N. Y., 1990.

[Bors and Pitas, 1996]

A. G. Bors, and I. Pitas, "Image Watermarking Using DCT Domain Constraints", Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland, vol. 3, Sept. 16-19, 1996, pp. 231-234.

[Braudaway, 1997]

G. W. Braudaway, "Protecting Publicly-Available Images with an Invisible Image Watermark," Proceedings of IEEE International Conference on Image Processing, vol. 1, October 1997, pp. 524-527.

[Bull et. al., 1999]

D. Bull, N. Canagarajah, and A. Nix, Insight into Mobile Multimedia Communications, Academic Press, UK, 1999.

[Cachin, 1998]

C. Cachin, "An information-Theoretic Model for Steganography," Proceedings of 2<sup>nd</sup> Workshop on Information Hiding, Lecture notes in Computer Science, Springer, 1998.

[Chae and Manjunath, 1998]

J. J. Chae, and B. S. Manjunath, "A Robust Embedded Data from Wavelet Coefficients," Proceedings of SPIE, Storage and Retrieval for Image and Video data base, Electronic Imaging EI'98, San Jose, 1998, vol. 3312, pp. 308-317.

[Chae, et. al., 1998]

J. J. Chae, D. Mukherjee and B. S. Manjunath, "Color Image Embedding using Multidimensional Lattice Structures," Proceedings of IEEE International Conference on Image Processing ICIP'98, Chicago, October 1998, vol. 1, pp. 460-464.

[Chae and Manjunath, 1999]

J. J. Chae and B. S. Manjunath, "A Technique for Data Hiding and Reconstruction without Host Image," SPIE, Electronic Imaging 99, Security and watermarking of Multimedia Content, January 1999, San Jose, California.

[Charles-Lee, 1997]

L. H. Charles Lee, Convolutional Coding: Fundamentals and Applications, Artech House Publisher, Boston, USA, 1997.

[Chen and Wornell, 1999]

B. Chen and G. W. Wornell, "Dither Modulation: A New Approach to Digital Watermarking and Information Embedding," in Proceedings of SPIE, Security and Watermarking of Multimedia Contents, vol. 3657, January 1999.

[Conway and Sloane, 1993]

J. H. Conway and N. J. A. Sloane, "Sphere Packings, Lattices and Groups", Second Edition, Springer-Verlag, New York, 1993.

[Costa, 1983]

M. Costa, "Writing on a Dirty Paper," IEEE Trans. On Information Theory, vol. IT-29, no. 3, May 1983.

[Cox et. al., 1996]

I. J. Cox, J. Kilian, T. Lieghton, and T. Shamoan, "A Secure, Robust Watermark for Multimedia", Workshop on Information Hiding, Newton Institute, Univ. of Cambridge, May 1996.

[Cox and Miller, 1997]

I.J. Cox and M. L. Miller, "A review of watermarking and the importance of perceptual modelling," Proceedings of the SPIE International Conference on Human Vision and Electronic Imaging II, Feb. 10-13, 1997, San Jose, CA, USA, pp.92-99.

[Cox et. al., 1997]

I. J. Cox, J. Kilian, T. Lieghton, and T. Shamoan, "Secure Spread Spectrum Watermarking for Multimedia", IEEE Transactions on Image Processing, vol. 6, no. 12, December 1997, pp. 1673-1687.

[Cox and Linnartz, 1998]

I. J. Cox, and J-P. M. G. Linnartz, "Some general methods for Tampering with Watermarks," IEEE Selected Areas of Communications, vol. 16, no. 4, 1998, pp. 587-593.

[Cox, et. al., 1999]

I. J. Cox, M. L. Miller, and A. L. McKellips, "Watermarking as Communications with Side Information," Proceedings of IEEE, vol. 87, no. 7, 1999, pp. 1127-1141.

[Cox, et. al., 2000]

I. J. Cox, M. L. Miller, and J. A. Bloom, "Watermarking Applications and Their Properties," Proceedings of the IEEE International Conference on Information Technology, Coding and Computing, March 27-29, 2000, Las Vegas, Nevada.

[Craver et. al., 1998]

S. Craver, N. Memon, B-L. Yeo, and M. M. Yeung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications", IEEE Journal on Selected Areas In Communications, vol. 16, no. 4, May 1998, pp. 573-586.

[Darven and Scott, 1996]

P. Darven and M. Scott, "Fractal based image steganography," Information Hiding, First International Workshop, Editor R. Anderson, vol. 1174 of Lecture Notes in Computer Science, Springer-Verlag, Berlin, 1996, pp. 279-294.

[Daubechies, 1992]

I. Daubechies, Ten Lectures on wavelets, SIAM, Philadelphia, 1992.

[Dixon, 1984]

R. C. Dixon, "Spread spectrum systems", John Wiley, 1984.

[Doufexi et. al., 2000]

[82] A. Doufexi, A. Nix, and D. Bull, "Robust wireless image transmission using Jointly-Optimized Modulation and Source Coding", Proceedings of IEEE 51<sup>st</sup> Vehicular Technology Conference, VTC 2000-Spring Tokyo, May 15-18, Japan.

[Duric et. al., 1999]

Z. Duric, N. F. Johnson, and S. Jajodia, "Recovering Watermarking from Images," Information and Software Engineering Technical Report ISE-TR-99-04, George Mason University, April, 1999.

[Fridrich, 1997]

I. Fridrich, "Methods for data hiding", Working paper, 1997, Center for Intelligent Systems & Department of Systems Science and Industrial Engineering, SUNY Binghamton, Binghamton, NY, <http://ssie.binghamton.edu/~jirif/>.

[Forney, 1966]

G. D. Forney, Concatenated Codes, MIT Press, Cambridge, Massachusetts, 1966.

[Fossorier, et. al., 1998]

M. P. C. Fossorier, Z. Xiong, and K. Zeger, "Joint Source-Channel Image Coding for a Power Constraint Noisy Channel," Proceedings of IEEE International Conference on Image Processing, ICIP '98, vol. 1, Chicago, October, 1998.

[Fu and Au, 2000]

M. S. Fu, and O. C. Au, "Hiding Data in Halftone Image using Modified data hiding error Diffusion," Proceedings of SPIE, Visual Communications and Image processing 2000, vol. 4067, part 3, pp. 1671-1680.

[Gersho and Gray, 1992]

A. Gersho, and R. M. Gray, Vector Quantization and Signal Compression, Kluwer Academic Publishers, Boston, 1992.

[Girod, 1989]

B. Girod, "The Information theoretical significance of Spatial and Temporal Masking in Video Signals," Proceedings of SPIE, Human Vision processing and Digital Display, vol. 1077, 1989, pp. 178-187.

[Hagenauer, 1996]

J. Hagenauer, "Iterative Decoding of Binary Block and Convolutional Codes," IEEE Transactions on Information Theory, vol. 42, March 1996, pp. 429-445.

[Hartung and Girod, 1997]

F. Hartung, and B. Girod, "Digital Watermarking of MPEG-2 Coded Video in the Bitstream Domain", Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal processing, April 1997, Munich, Germany, vol. 4, pp. 2621-2624.

[Haykin, 1994]

S. Haykin, "Communication systems", John Wiley and Sons, 1994.

[Hernandez et. al 1998a]

J. R. Hernandez, F. Perez-Gonzalez, J. M. Rodriguez, and G. Nieto, "Performance Analysis of a 2-D Multipulse Amplitude Modulation scheme for Data Hiding and Watermarking of Still Images," IEEE Journal of Selected Areas in Communications, vol. 16, May 1998, pp.510-524.

[Hernandez et. al., 1998b]

J. R. Hernandez, F. Perez-Gonzalez and J. M. Rodriguez, "The Impact of Channel Coding on the performance of Spatial Watermarking for Copyright Protection," Proceedings of International Conference on Acoustics, Speech, and Signal Processing ICASSP'98, Seattle, Washington, USA, May 1998, vol. 5, pp. 2973-2976.

[Herrigel, et. al., 1998]

A. Herrigel, J. O. O'Ruanaidh, H. Petersen, S. Pereira, and T. Pun, "Secure Copyright protection Techniques for Digital Images," Second information Hiding Workshop, 1998.

## [Image Database]

<http://sipi.usc.edu/services/database/Database.html>.

## [Inoue, et. al., 1998]

H. Inoue, A. Miyazaki, A. Yamamoto, and T. Katsura, "A Digital Watermark Based on the Wavelet Transform and its Robustness to Image Compression," Proceedings of IEEE International Conference on Image Processing, ICIP '98, vol. 1, Chicago, October, 1998, pp. 391-395.

## [Jayant &amp; Noll, 1984]

N. S. Jayant and P. Noll, "Digital Coding of waveforms" Prentice Hall, 1984.

## [Johannesson, and Zigangirov, 1999]

R. Johannesson, and K. Zigangirov, Fundamentals of Convolutional Coding, IEEE Press, IEEE Series on Digital and Mobile Communication, NY, USA, 1999.

## [Johnson and Jajodia, 1998]

N. J. Johnson, and S. Jajodia, "Exploring Steganography: Seeing the Unseen," IEEE Computer, vol. 31, no. 2, February 1998, pp. 26-34.

## [Kesal, et. al., 2000]

M. Kesal, M. K. Mihcak, R. Koetter, and P. Moulin, "Iteratively Decodable Codes for Watermarking Applications," Proc. 2<sup>nd</sup> Int. Symp. On Turbo Codes and Related Topics, Brest, France, September 2000.

## [Kim et. al., 1998]

S. W. Kim, S. Suthaharan, H. K. Lee, and K. R. Rao, "A watermarking scheme using human visual model and BN distribution", The 6<sup>th</sup> IEEE International Workshop on Intelligent Signal Processing and Communication Systems, 5-6 November 1998, Melbourne, Australia, vol.1, pp. 396-400.

## [Koch and Zhao, 1995]

E. Koch, and J. Zhao, "Towards Robust and Hidden image Copyright Labeling", Proceedings of the IEEE Workshop on Nonlinear Signal and

Image Processing, Neos Marmaras, Halkidiki, Greece, June 20-22, 1995, pp. 452-455.

[Kuhn, 1997]

M. G. Kuhn, "StirMark", available at <http://www.cl.cam.ac.uk/~mgk25/stirMark/>, Security Group, Computer Lab, Cambridge University, UK (E-mail: [mkuhn@acm.org](mailto:mkuhn@acm.org)), 1997.

[Kunder and Hatzinakos, 1997]

D. Kunder, and D. Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion," Proceedings of the IEEE International Conference on Image Processing, Oct. 26-29, 1997, Santa Barbara, CA, vol. 1, pp. 544-547.

[Kunder and Hatzinakos, 1998]

D. Kunder, and D. Hatzinakos, "Digital watermarking using multiresolution wavelet decomposition," Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing, Seattle, Washington, May 1998, vol. 5, pp. 2969-2972.

[Kunder and Hatzinakos, 1999]

D. Kunder, and D. Hatzinakos, "Digital watermarking for Telltale Tamper-Proofing and Authentication," Proceedings of the IEEE-Special Issue on Identification and Protection of Multimedia Information, July 1999, pp. 1167-1180.

[Langelaar et. al., 1996]

G. Langelaar, J. van der Lobbe, and J. Biemond, (1996), "Copy protection for multimedia based on labeling techniques", Available WWW:[http://www.it.et.tudelft.nl/pda/smash/public/benelux\\_cr.html](http://www.it.et.tudelft.nl/pda/smash/public/benelux_cr.html).

[Langelaar et. al., 1997]

G. C. Langelaar, J. van der Lubbe, and R. L. Lagendijk, "Robust Labelling Methods for Copy Protection of Images", Proceedings of the SPIE International Conference on Storage and Retrieval for Image and Video Databases V, vol. 3022, San Jose, CA, February 13-14, 1997, pp. 298-309.

[Langelaar, et. al., 1999]

G. C. Langelaar, R. L. Lagendijk, and J. Biemond, "Watermarking by DCT Coefficients Removal: A Statistical Approach to Optimal parameter

Settings," Proceedings of SPIE Electronic Imaging' 99, Security and Watermarking of Multimedia Contents, January 1999, San Jose (CA), USA.

[Langelaar et. al., 2000]

G. C. Langelaar, I. Setyawan, and R. L. Lagendijk, "Watermarking Digital Image and Video Data," IEEE Signal Processing Magazine, September 2000, pp. 20-46.

[Leduc, 1994]

J. -P. Leduc, Digital moving Pictures Coding and transmission on ATM Networks, Elsevier Science B. V., The Netherlands, 1994.

[Lin and Costello, 1983]

S. Lin, and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, USA, 1983.

[Lin, et. al., 1998]

S. Lin, T. Kasami, T. Fujiwara, and M. Fossorier, Trellises and trellis-Based Decoding Algorithms for Linear Block Codes, Kluwer Academic Publishers, MA, USA, 1998.

[Linde et. al., 1989]

Linde Y., Buzo A., and Gray R. M., "An algorithm for vector quantizer design", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp. 553-559, 1989.

[Maes, et. al., 2000]

M. Maes, T. Kalker, J-P. M. G. Linnartz, J. Talstra, G. F. G. Depovere, and J. Haitsma, "Digital Watermarking for DVD Video Copy Protection," IEEE Signal Processing magazine, September 2000, pp.47-57.

[Mallat, 1989]

S. Mallat, "A Theory for Multiresolution Signal Decomposition: The Wavelet Representation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 11, no. 7, July 1989.

[Marvel et. al., 1998]

L. M. Marvel, C. G. Boncelet, Jr., and C. T. Retter, "Reliable blind information hiding for images," Proceedings of Second International Workshop, IH'98, Portland, Oregon, USA, April 1998.

[Mason and Zimmermann, 1960]

S. J. Mason, and H. J. Zimmermann, Electronic Circuits, Signals and Systems, John Wiley and Sons, INC., 1960.

[MATLAB]

Matlab software, The Mathworks, Inc., Web: [www.mathworks.com](http://www.mathworks.com).

[Maxemchuck and Low, 1997]

N. F. Maxemchuck, and S. Low, "Marking Text Documents," Proceedings of IEEE International Conference on Image Processing, ICIP '97, Santa Barbara, California, 1997.

[Meng and Chang, 1998]

J. Meng and S-F Chang, "Embedding Visible Video watermarking in the Compressed Domain," Proceedings of IEEE International Conference on Image Processing, ICIP '98, vol. 1, Chicago, October, 1998, pp. 474-477.

[Mintzer et. al., 1997]

F. Mintzer, G. W. Braudaway, and M. M. Yeung, "Effective and Ineffective Digital Watermark," Proceedings of IEEE International Conference on Image Processing, ICIP '97, vol. 3, Santa Barbara, California, 1997.

[Moulin, et. al., 2000]

P. Moulin, M. Kivanc Mihcak, and G.-I. Lin, "An Information-Theoretic Model for Image Watermarking and Data Hiding," Proceedings of IEEE International Conference on image Processing, Vancouver, B.C., Canada, 2000.

[Nievergelt, 1999]

Y. Nievergelt, Wavelets Made Easy, Birkhauser, Boston, 1999.

[O'Ruanaidh et. al., 1995]

J. J. K. O'Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking Digital Images for Copyright Protection", Proceedings of the International Conference on Image Processing and its Applications, Edinburgh, Scotland, July 1995, pp. 321-326.

[O' Ruanaidh and Pun, 1998]

J. J. K. O' Ruanaidh, and T. Pun, "Rotation, scale and translation invariant spread spectrum digital image watermarking," Signal Processing (Special Issue on Watermarking), vol. 66, no. 3, 1998, pp. 357-317.

[O' Ruanaidh et. al., 1996a]

J. J. K. O' Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking Digital Images for Copyright protection," IEE Proceedings on Vision, Image and Signal Processing, vol. 143, no. 4, 1996, pp. 250-256.

[O' Ruanaidh et. al., 1996b]

J. J. K. O' Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase watermarking of digital images," Proceedings of the IEEE International Conference on Image Processing, Lausanne, Switzerland, Sept. 16-19, 1996, vol. 3, pp. 239-242.

[Papoulis, 1991]

A. Papoulis, Probability, Random Variables and Stochastic Processes, Third Edition, McGraw-Hill, 1991.

[Pattan, 2000]

B. Pattan, Robust Modulation Methods and Smart Antennas in Wireless Communications, Prentice Hall PTR, 2000.

[Pereira et. al., 1999]

S. Pereira, J. K. O'Ruanaidh, and T. Pun, "Secure robust digital watermarking using the Lapped Orthogonal Transform", Proceedings of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging'99, San Jose, CA, Jan 25-27 1999, vol. 3657.

[Pereira et. al., 2000]

S. Pereira, S. Voloshynovskiy, and T. Pun, "Effective Channel Coding for DCT Watermarks," Proceedings of IEEE International Conference on Image Processing 2000 (ICIP' 2000), 10-13 September 2000, Vancouver, Canada.

[Petitcolas and Anderson, 1998]

F. A. P. Petitcolas, and R. J. Anderson, "Weakness of copyright marking systems," Multimedia and Security Workshop at ACM Multimedia'98, Bristo, U.K., September, 1998.

[Petitcolas et. al., 1998]

F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn, "Attacks on copyright marking systems," in the Second Workshop on Information Hiding, Portland, Oregon, USA, 14-17 April, 1998, pp. 211-214.

[Petitcolas, et. al., 1999]

F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding-A Survey", proceedings of the IEEE, Special Issue on Protection of Multimedia Contents, vol. 87, no. 7, July 1999, pp. 1062-1078.

[Petitcolas, 2001]

F. A. P. Petitcolas, Digital Watermarking News, <http://www.cl.cam.ac.uk/~fapp2/steganography/news.html>.

[Petitcolas]

[http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/image\\_database.html#Lena](http://www.cl.cam.ac.uk/~fapp2/watermarking/benchmark/image_database.html#Lena).

[PictureMarc]

PictureMarc, Digimarc, <http://www.digimarc.com>.

[Pitas, 1996]

I. Pitas, "A Method for Signature casting on digital images", In International Conference on Image Processing, vol. 3, Sept. 1996, pp. 215-218.

[Pitas, 1998]

I. Pitas, "Robust Image Watermarking in the Spatial Domain," *Signal Processing*, vol. 66, 1998, pp. 385-403.

[Piva et. al., 1997]

A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "DCT-based Watermark Recovery without Resorting to the Uncorrupted Original Image", *Proceedings of the IEEE International Conference on Image Processing*, October 26-29, 1997, Santa Barbara, CA, vol. 1, pp. 520-523.

[Piva et. al., 1998a]

A. Piva, M. Barni, F. Bartolini, "Copyright Protection of Digital Images by Means of Frequency Domain Watermarking", *Proc. SPIE Mathematics of data / Image Coding, Compression, and Encryption* vol. 3456, pp.25-35, San Diego, California, 1998.

[Piva et. al., 1998b]

A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "Application-driven requirements for digital watermarking technologies", *European Multimedia, Microprocessor Systems and Electronic Commerce EMMSEC 98*, Bordeaux, France, September 28-30, 1998 in *Technologies for the Information Society: Developments and Opportunities*, J. -Y. Roger et al. (Eds) IOS Press, 1998, pp.513-520.

[Piva et. al., 1998c]

A. Piva, M. Barni, F. Bartolini, and V. Cappellini, "Threshold selection for correlation-based watermark detection," *Proceedings of COST 254 Workshop on Intelligent Communications*, L'Aquila, Italy, June 4-6, 1998, pp. 67-72.

[Plataniotis and Venetsanopoulos, 2000].

K. N. Plataniotis, and A. N. Venetsanopoulos, *Color Image Processing and Applications*, Springer-Verlag Berlin, 2000.

[Podilchuck and Zeng, 1997a]

C. I. Podilchuck, and W. Zeng, "Perceptual watermarking of still images," *IEEE Signal Processing Society 1997 Workshop on Multimedia Signal Processing*, June 23-25, 1997, Princeton, New Jersey, USA

[Podilchuck and Zeng, 1997b]

C. I. Podilchuk, and W. Zeng, "Watermarking of the JPEG bitstream," Proceedings of the International Conference on Imaging Science, Systems, and Technology, Las Vegas, Nevada, June 30-July 3, 1997, pp. 253-260.

[Podilchuck and Zeng, 1998]

C. I. Podilchuck, and W. Zeng, "Image-adaptive watermarking using visual model," IEEE Journal of Selected Areas in Communications, Special Issue on Copyright and Privacy Protection, vol. 16, no. 4, May 1998, pp. 525-539.

[Proakis, 1995]

J. Proakis, Digital Communications, McGraw-Hill Publisher, NY, USA, 3<sup>rd</sup> Edition, 1995.

[Ramkumar and Akansu, 1998]

M. Ramkumar and A. N. Akansu, "Information Theoretic Bounds for Data Hiding in Compressed Images," Proceedings of IEEE Signal processing Society 1998 Workshop on Multimedia Signal Processing, December 7-9, 1998, Los Angeles, California, USA.

[Reed and Chen, 1999]

I. S. Reed, and X. Chen, Error-Control Coding for Data Networks, Kluwer Academic Publishers, Norwell, Massachusetts, USA, 1999.

[Qiao and Nahrstedt, 1998]

L. Qiao, and K. Nahrstedt, "Watermarking Schemes and Protocols for Protecting Rightful Ownership and Customer's Rights", Academic Press, Journal of Visual Communication and Image Representation, 1998.

[Schlegel, 1997]

C. Schlegel, Trellis Coding, Chapter 8, pp. 233, IEEE Press, NJ, USA, 1997.

[Schneider and Chang, 1996]

M. Schneider, and S.-F. Chang, "A robust Content Based Digital Signature for Image Authentication," Proceedings of IEEE International Conference on Image Processing, ICIP'96, Lausanne, 1996, vol. III, pp.227-230.

[Schyndel et. al., 1994]

R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne, "A Digital Watermark", Proceedings of the International Conference on Image Processing, vol. 2, 1994, pp. 86-90.

[Servetto et. al., 1998]

S. D. Servetto, C. I. Podilchuk, and K. Ramachandran, "Capacity issues in digital image watermarking," in Proc. ICIP'98, IEEE Int. Conf. On Image Processing, Chicago, Illinois, USA, 4-7 October 1998, pp. I: 445-449.

[Shannon, 1948]

C. Shannon, "A Mathematical Theory of Communication", Bell System Technical Journal, vol. 27, 1948, pp. 379-423 and 623-656.

[Simon, et. al., 1989]

M. K. Simon, J. K. Omura, R. A. Scholtz, and B. K. Levitt, "Spread spectrum communications", Computer Science Press, 1989.

[Smith and Comiskey, 1996]

J. R. Smith, and B. O. Comiskey, "Modulation and Information Hiding in Images", Proceedings of the First Information Hiding Workshop, Isaac Newton Institute, University of Cambridge, UK, May 1996, Springer-Verlag Lecture Notes in Computer Science, vol. 1174.

[Stego Dos]

Stego Dos, Blocl Wolf's Picture Encoder v0.9B,  
<ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/stegodos.zip>.

[Su et. al., 1999]

J. K. Su, F. hartung, and B. Girod, "A channel model for a watermark attack", Proceedings of SPIE, Security and Watermarking of Multimedia Contents, Electronic Imaging'99, San Jose, CA, Jan 25-27 1999, vol. 3657.

[SureSign]

SureSign, Signum Technologies, <http://www.signumtech.com>.

[Swanson et. al., 1996a]

M. D. Swanson, B. Zhu, and A. H. Tewfik, "Transparent robust image watermarking," Proceedings of the 1996 International Conference on Image Processing, Lausanne, Switzerland, Sept. 16-19, 1996, vol. 3, pp. 211-214.

[Swanson et. al., 1996b]

M. D. Swanson, B. Zhu, and A. H. Tewfik, "Robust data hiding for images," In IEEE Digital Signal Processing Workshop (DSP 96), Loen, Norway, September, 1996, pp. 37-40.

[Swanson et. al., 1997a]

M. D. Swanson, B. Zhu, and A. H. Tewfik, "Multiresolution scene-based video watermarking using perceptual models", Proceedings of the IEEE International Conference on Image Processing, Oct. 26-29, 1997, Santa Barbara, CA, vol. 2, pp. 558-561.

[Swanson et. al., 1997b]

M. D. Swanson, B. Zhu, and A. H. Tewfik, "Data hiding for Video-in-Video," Proceedings of the IEEE International Conference on Image Processing, October 26-29, 1997, Santa Barbara, CA, vol. 2, pp. 676-679.

[Swanson et. al., 1998]

M. D. Swanson, M. Kobayashi, and A. Tawfik, "Multimedia Data-Embedding and Watermarking Technologies", Proceedings of the IEEE, vol. 86, no. 6, June 1998, pp. 1064-1087.

[Tao and Dickinson, 1997]

B. Tao, and B. Dickinson, "Adaptive watermarking in the DCT domain ", Proceedings of the 1997 IEEE International Conference on Acoustics,

Speech, and Signal Processing, April 21-24, Munich, Germany, vol. 4, pp. 2985-2988.

[Taub and Schilling, 1986]

H. Taub, and D. L. Schilling, Principles of Communication Systems, McGraw-Hill Book Company, 1986.

[Tewfik, 2000]

A. H. Tewfik, "Digital Watermarking," IEEE Signal processing Magazine, September 2000.

[Tirkel et. al., 1993]

A. Z. Tirkel, G. A. Rankin, R. M. van Schyndel, W. J. Ho, N. R. A. Mee, and C. F. Osborne, "Electronic Watermark", Digital Image computing, Technology and Applications—DICTA '93, Macquarie University, Sydney, 1993, pp. 666-673.

[Ungerboeck, 1982]

G. Ungerboeck, "Channel Coding with Multilevel/Phase Signals," IEEE Transactions on Information Theory, vol. IT-28, no. 1, January 1982.

[Ungerboeck, 1987]

G. Ungerboeck, "Trellis Coded Modulation with Redundant Signal Set," IEEE Communications Magazine, vol. 27, february, pp.5-21.

[Vetterli and Kvacevic, 1995]

M. Vetterli, and J. Kovacevic, Wavelets and Subband Coding, Prentice Hall PTR, Englewood Cliffs, NJ, 1995.

[Viterbi, 1979]

A. J. Viterbi, Principles of Digital communication and Coding, NY, McGraw-Hill, 1979.

[Vucetic and Yuan, 2000]

B. Vucetic, and J. Yuan, Turbo Codes: Principles and Applications, Kluwer Academic Publishers, MA, USA, 2000.

[Watson, 1992]

A. B. Watson, "DCT quantization matrices visually optimized for individual images," in Proceedings SPIE Conference on Human Vision, Visual Processing and Digital Display IV, vol. 1913, San Jose, 1992, pp. 202-216.

[Wilson et. al., 1995]

T. A. Wilson, S. K. Rogers, and L. R. Myers, "Perceptual based hyperspectral image fusion using multiresolution analysis," Optical Engineering, vol. 34, no. 11, November, 1995, pp. 3154-3164.

[Wolfgang and Delp, 1997a]

R. B. Wolfgang and E. J. Delp, "Overview of image security techniques with applications in multimedia systems," Proceedings of the SPIE International Conference on Multimedia Networks: Security, Display, Terminals, and Gateways, November 4-5, 1997, Dallas, Texas, vol. 3228, pp.297-308.

[Wolfgang and Delp, 1997b]

R. B. Wolfgang, and E. J. Delp, "A Watermarking Technique for Digital Imagery: Further Studies", Proceedings of the International Conference on Imaging Science, Systems, and Technology, Las Vegas, Nevada, June 30-July 3, 1997, vol. 1, pp.279-287.

[Wu and Liu, 1998]

M. Wu, B. Liu, "Watermarking for Image Authentication," Proceedings of the International Conference on Image Processing (ICIP98), October 4-7, 1998, Chicago, Illinois, USA, pp. 437-441.

[Wu, et. al., 1999]

M. Wu, H. H. Yu, and A. Gelman, "Multi-Level Data Hiding for Digital Image and Video," SPIE Photonics East'99, Boston, 1999.

[Wu and Yu, 2000]

M. Wu, and H. H. Yu, "Video Access Control Via Multi-Level data Hiding," proceedings of IEEE International Conference on Multimedia and Expo. (ICME'00), New York City, 2000.

[Xia et. al., 1997]

X. -G. Xia, C. G. Boncelet, G. R. Arce, "A multiresolution watermark for digital images," Proceedings of the IEEE International Conference on Image Processing, Santa Barbara, CA, Oct. 26-29, 1997, vol. 3, pp. 548-551.

[Xia et. al., 1998]

X.-G. Xia, C. G. Boncelet, and G. Arce, "Wavelet transform based watermark for digital images", The Electronic Journal for the Optical Society of America, Optics Express, vol. 3, no. 12, December 1998, pp. 497-511.

[Yoshida, 1999]

J. Yoshida, "Digital Watermarking Companies Set to Merge," Electrical Engineering Times, 16 June, 1999.

[Zeng and Liu, 1997]

W. Zeng, and B. Liu, "On Resolving Rightful Ownerships of Digital images by Invisible watermarks", Proceedings of the IEEE International Conference on Image Processing, Oct. 26-29, 1997, Santa Barbara, CA, vol. 1, pp. 552-555.

[Zhao, et. al., 1998]

J. Zhao, E. Koch, and C. Luo, "In Business Today and Tomorrow," Communications of the ACM, vol. 41, no. 7, Jily 1998, pp. 67-72.

[Zhu et al. 1998]

W. Zhue, Z. Xiong, and Y-Q. Zhang, "Multiresolution Watermarking for Images and Video: A Unified Approach," Proceedings of IEEE International Conference on Image Processing ICIP'98, Chicago, Illinios, USA, 4-7 October 1998.

## **Published Papers**

- (1) N. Abdulaziz and K. K. Pang, "Source and channel coding approach to data hiding," Proceedings of SPIE, Visual Communications and Image Processing 2000 Conference, 20-23 June 2000, Perth, Australia, pp. 1526-1535.
- (2) N. Abdulaziz and K. K. Pang, "Robust data hiding for images," Published at World Computer Congress WCC 2000, International Conference on Communications Technology (ICCT 2000), 21-25 August 2000, Beijing, China, vol. 1, pp. 380-383.
- (3) N. Abdulaziz and K. K. Pang, "Performance evaluation of data hiding system using wavelet transform and error-control coding," IEEE International Conference on Image Processing 2000 (ICIP 2000), 10-13 September 2000, Vancouver, Canada.
- (4) N. Abdulaziz and K. K. Pang, "Data embedding using trellis coding," International Conference on Communication, Computer and Power (ICCCP'01), February 12-14, 2001, Muscat, Sultanate of Oman.

# Source and channel coding approach to data hiding

N. K. Abdulaziz\*<sup>1</sup> and K. K. Pang\*<sup>1</sup>

Dept of Electrical and Computer Systems Engineering  
Monash University  
Clayton, VIC 3168, Australia

## ABSTRACT

In this work, a robust data embedding scheme, which uses a source and channel coding framework for data hiding is implemented. The data to be embedded, referred to as the signature data, comprises of two different data types, text messages and images as the signature data. The first data type used was the text message, where the text message is converted into bits and these bits are coded using Reed-Solomon codes, and the resulting code is hidden into the wavelet transformed coefficients of the host image. For hiding images as signature data, an image is used as large as 128 x 128 to be hidden into a host image of size 256 x 256. The perturbations are controlled by a maximum allowable visible distortion that can be introduced in the host using a model of the human visual perception. This method could be used for both digital watermarking related applications as well as for data hiding purposes.

**Keywords:** Data hiding, digital watermarking, error correcting codes, wavelet transform, vector quantization

## 1. INTRODUCTION

Motivated by the overwhelming desire for Internet data security, digital watermarking has recently emerged as an important area of research in multimedia data processing<sup>1-2</sup>. A digitally watermarked image is obtained by invisibly hiding a signature information into the host image. The signature is recovered using an appropriate decoding process. The challenge is to simultaneously ensure that the watermarked image is perceptually indistinguishable from the original, and that the signature be recoverable even when the watermarked image has been compressed or transformed by standard image processing operations.

Several interesting data hiding techniques for images have been proposed. The most common approach is to add fixed amplitude noise to the host image<sup>3-7</sup>. All of the schemes described utilize the fact that digital media contain perceptually insignificant components, which may be replaced or modified to embed data. While most of the research on watermarking concentrates on copyright protection in internet data distribution<sup>8-9</sup>, a different kind of watermarking, commonly known as data hiding, is at present receiving considerable attention. Data hiding is intended to hide larger amounts of data into host source, rather than just to check for authenticity and copyright information<sup>10-11</sup>. In other words, the problem of watermarking or copyright protection is a special case of the generic problem of data hiding, where a small signature is embedded with greater robustness to noise.

In this work, the data to be embedded, referred to as the signature data, comprises of two different data types, text messages and images. The first data type used was the text message, where the text message is converted into bits and these bits are coded using Reed-Solomon codes, and the resulting code is hidden into the wavelet transformed coefficients of the host image. For hiding images as signature data, an image is used as large as 128 x 128 to be hidden into a host image of size 256 x 256. To accommodate for the large number of bits to be hidden, the signature data is first compressed using vector quantization and the indices obtained in the process are embedded in the wavelet transform coefficients of the host image.

---

\*Correspondence: Email: (nidhal.abdulaziz, Khce.k.Pang) @eng.monash.edu.au

The transformed coefficients of the host data are grouped into vectors, and the vectors are perturbed using noise-resilient channel codes derived from error correcting codes<sup>12</sup>. Error correcting codes provide coders with a tool to recover lost information such as errors, erasures and deletions.

It's worth noting here that the watermarking problem is a special case of the data hiding algorithm. In watermarking applications, the allowable distortion is very small, and the requirement of robustness is very strict. As a result, the amount of data that can be hidden reliably is small. In this work, the focus is on the more generic problem where any amount of data may be hidden, but in a manner such that for a given allowable distortion the robustness against data transformations, compression, or attacks, is maximized.

The perturbations are controlled by a maximum allowable visible distortion that can be introduced in the host using a model of the human visual perception. The spatial masking model of Girod<sup>13</sup> was used to adjust the watermark strength (or the scaling factor  $\alpha$ ), so that the watermarked image is perceptually identical to the original image. This method could be used for both digital watermarking related applications as well as for data hiding purposes. The scale factor  $\alpha$  controls the relative amount of host and signature image data in the embedded image. A large scale factor can be used for data hiding where it is desirable to maintain perceptual quality of the embedded image. A lower scale factor is better suited for watermarking where robustness to typical image processing operations is needed.

Compared to prior work in digital watermarking, the proposed scheme can handle a significantly large quantity of data such as a gray scale images. A trade-off between the quantity of hidden data and the quality of the watermarked image is achieved by varying the number of quantization levels for the signature, the code word length, and the scale factor for embedding. Moreover, the proposed scheme here focuses on hiding the signature mostly in the low frequency DWT bands, and stable reconstruction can be obtained even when the images are transformed, quantized (as in JPEG), or otherwise modified by enhancement or low pass filtering operations.

In this paper, we use the vector quantization scheme for compressing the signature image, the compressed indices are injected into the wavelet coefficients of the host image in a vector based perturbation and the watermarked host is subjected to JPEG compression for manipulation of the watermarked image before attempting retrieval. As our experimental-results indicate, there are no visible distortions in the watermarked image, and the recovered signature is similar to the original signature even after 5% JPEG lossy compression quality factor.

In the next section a discussion of the proposed embedding and extracting algorithm is presented.

## 2. DATA EMBEDDING

The host data is first orthogonally transformed, and the transform coefficients are then perturbed in a definite fashion to represent hidden information. Note that the use of a transform is not essential to this approach because a raw image is by itself an expansion on the standard bases. However, orthogonal transformations may yield a subset of coefficients which when perturbed, either result in a lower probability of erroneous detection after a particular kind of transformation, or yields less perceptually significant distortions, or strike a compromise of both.

Moreover, it is well known that embedding in the low-frequency bands is more robust to manipulations such as enhancement and image compression. However, changes made to the low frequency components may result in visible artifacts. Modifying the data in a multiresolution framework, such as a wavelet transform<sup>14</sup>, appears quite promising for obtaining good quality embedding with little perceptual distortion.

The embedding procedure is explained by means of the diagram in Figure 1. The data to be embedded is first source coded, either losslessly or lossily depending on the nature of the data to generate a sequence of symbols. For hiding large amount of data, as it is the case for hiding images, the signature image is first quantized using vector quantization with Linde-Buzo-Gray (LBG) algorithm<sup>15</sup>. Vector quantization (VQ) transforms the vectors of data into indices that represents the clusters of vectors. For example, for the case of Bear image of size 128 x 128, The image is first decomposed into 4-dimensional image vectors; in this case the signature image is divided into 2 x 2 blocks. Each vector is compared with a collection of representative codevectors taken from a previously generated codebook (the source codebook). Best match codevector is chosen using a minimum distortion rule. After the minimum distortion codevector has been found, the index  $i$  is used to represent the signature vector. For 4-dimensional vector quantizer, the codebook contains 16 levels or in another word, the quantization level  $\beta = 16$ . A high value of quantization level ( $\beta$ ) quantizes the signature finely, but  $\alpha$  must now be higher too so that the probability of error is sufficiently low. This in turn degrades the transparency of the watermarked image. The

choice of the parameters  $\alpha$  and  $\beta$  determines the trade-off between the transparency and the quality of the hidden data. In order to embed these indices; each index is coded using channel coding like the error-correcting codes before embedding in the transformed host image. Error-control coding techniques are used to detect and correct errors that occur in the data transmission in a digital communication system. The transmitting side of the error-control coding adds redundant bits or symbols to the original information sequence. The receiving side of the error-control coding uses these redundant bits or symbols to detect and correct the errors that occurred during transmission.

For hiding these codes into the host image, a single level of the discrete wavelet transform (DWT) decomposition of the host image is made before data embedding, where a group of  $N$  transformed coefficients are used to form an  $N$ -dimensional vector. These vectors are then modified by the coded indices after it has been scaled by a factor  $\alpha$ . The parameter  $\alpha$  determines the transparency constraint. That is, if  $\vec{v}$  represents a vector of host DWT coefficients after grouping, and the index of the vector quantized signature image is  $i$ , then the perturbed vector  $\vec{w}$  is given by:

$$\vec{w} = \vec{v} + \alpha \cdot \vec{C}(s_i) \quad (1)$$

where  $\vec{C}(s_i)$  represent the channel code corresponding to the symbol  $s_i$ , where  $i = 1, \dots, \beta$ . Each index of the signature image is hidden into  $N$  coefficients in the LL band of the host; the remaining indices are hidden in the other subbands of the host (HL, LH, and HH). The scale factor for embedding is chosen in a way that assure an acceptable quality for the watermarked image. In addition to the traditional Peak Signal to Noise Ratio (PSNR) measure for assessing this quality, a model of the human visual perception is also used. The spatial masking model of Girod<sup>16</sup> was used to adjust the watermark strength (or the scaling factor  $\alpha$ ), so that the watermarked image is perceptually identical to the original image. The spatial masking model of Girod is based on the physics of human visual perception and accurately describes the visibility of artefacts around edges and in flat areas in digital images. The model is a general model and can be applied to videos. In this case, only the spatial portion of the model is used.

For security in copyright protection, we can select special regions in the transform domain to embed data, or randomly group the coefficients to form a vector using a private key. It is to be noted, however, that in general, the less the quantity of data hidden, and the more secure it can be made.

### 3. EXTRACTING DATA

If the original host image is available, then the operation of data injection and retrieval are, in fact, very similar to the channel coding and decoding operations in a typical digital communication systems. In watermarking in the transform domain, the original host data is transformed, and the transformed coefficients are perturbed by a small amount in one of several possible ways in order to represent the signature data. When the watermarked image is compressed or modified by image processing operations, this is equivalent to adding noise to the perturbed coefficients. The retrieval operation subtracts the received coefficients from the original ones to obtain the noisy perturbations. The true perturbations that represent the injected data are then estimated from the noisy data as best as possible.

Recovering the hidden data starts with the same DWT of the received watermarked image that was used to embed the data. The true host image coefficients (known to the receiver) are then subtracted from the coefficients of the received image to obtain the noisy perturbations. Note that these perturbations can be "noisy", because of various possible transformations of the watermarked data. These coefficients are then grouped into groups of  $N$  in the same manner as they were grouped during encoding to obtain a vector  $\vec{z}$ , which is scaled by the factor  $1/\alpha$ . The resulting vector  $1/\alpha \cdot \vec{z}$  is then decoded to find the index  $i$ . From the index  $i$ , and using a duplicate codebook and a table lookup, the signature data can be recovered. Figure 2 shows the details of symbol recovery and signature extraction.

### 4. EXPERIMENTAL RESULTS AND CONCLUSIONS

Different images and text messages were used in testing this algorithm and different parameters were implemented as a measure such as the Peak Signal to Noise Ratio (PSNR) of the reconstructed image as a function of the JPEG coding quality factor ( $Q$ ) for different codes and different scale factor  $\alpha$ , the similarity of the reconstructed image to the original signature image for various levels of  $Q$ , and the Bit Error Rate (BER) as a function of  $Q$ .

One of the test images used in this work is shown in Figure 3. The host image, Lena a 256 x 256 gray scale, and the signature images Bear, and Peppers, all are 128 x 128 gray scale. A 1-stage discrete Haar wavelet transform is used for both the encoder and the decoder in this work.

Figure 1 illustrates the data hiding technique on Lena image, 256 x 256 grayscale. Using this scheme, the text "We are investigating data hiding using Reed-Solomon codes" is embedded in the host Lena image. Girods' model<sup>16</sup> indicated that  $\alpha = 10$  produced watermarks with less than 1% of pixels with visible changes and PSNR = 38.3 dB for the case of hiding text messages of length 448 bits. The image with the hidden data is shown in Figure 3 (c) with PSNR = 35.56 dB, and a scaling factor  $\alpha = 40$ , which produced watermarks with less than 3% of pixels with visible changes. Figure 4 shows Lena image watermarked with bear image using LBG vector quantizer with blocks of 2 x 2 and 16 levels and BCH coding of (7,4) at various scale factor, without any compression. Note that the scale factor  $\alpha$  controls the relative weight of host and signature image contributions to the fused image. As the value of  $\alpha$  increases, the quality of the watermarked image degrades. For example, in Figure 4, one can see artifacts in the background for  $\alpha = 20$ .  $\alpha = 10$  appear to be a reasonable value in terms of the trade-off between quality of the watermarked image and robustness to signature recovery under image compression.

Figure 5 shows the compressed watermarked image of Baboon and the recovered Peppers image for different levels of JPEG compression. The recovered image at Q = 5% is clear even though the watermarked image of Baboon has degraded a lot and is of no commercial value.

Figure 6 shows the signature images recovered from the watermarked Lena image after 100%, 80%, 50%, and 5% JPEG quality factor. In general, most of the recovered signature images are of high quality, when the scale factor  $\alpha = 10$ .

In Figure 7 a plot of bit error (BER) for different levels of JPEG coding at different quality factors is shown for hiding text into image. From this figure, it is clear that the quality factor at which the watermark is lost is when Q = 30% which is much better than previous results published in reference<sup>1</sup> that was obtained using both spatial and frequency data hiding techniques. Similarly, Figure 8 shows the plot of BER versus JPEG coding at different quality settings for hiding bear image into the host image. The algorithm works well under high quality coding conditions yet degrades more rapidly when the coding becomes too lossy. Figure 9 shows the PSNR of the recovered bear image for different values of JPEG compression. The quality of the recovered signature with a large scale factor  $\alpha$  is obviously much better than those with a smaller  $\alpha$ . On the other hand, the number of quantization levels  $\beta$  determines the coarseness of quantization and therefore the quality of the signature image hidden in the host.

The technique presented here is a new technique that allows the correction of channel errors due to compression attack; it also has the ability to hide large amount of data into the host image. However, There are still some work to be done and the watermarked image need to be subjected to other attacks, for example, noise addition, filtering and other image processing techniques and see how it will affect the recovered signature image.

#### ACKNOWLEDGMENTS

The author would like to thank University of Southern California, Signal and Image processing Institute (USC-SIPI) Image Database for using their classic images (Lena, Baboon, and Peppers), and also to thank Mr. Robert Barber from Barber Nature Photography for using Black Bear Image.

#### 5. REFERENCES

1. M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia data-Embedding and Watermarking Technologies", Proceedings of the IEEE, *vol. 86, no. 6*, June 1998, pp. 1064-1087.
2. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding-A Survey", Proceedings of the IEEE, Special Issue on Protection of Multimedia Contents, *vol. 87, no. 7*, July 1999, pp. 1062-1078.
3. Bender W., Gruhl D., and Morimoto N., "Techniques for Data Hiding", Technical Report, MIT Media Lab, 1994.
4. Pitas I., and Kaskalis T., "Applying Signatures on Digital Images", Proceedings 1995 IEEE Nonlinear Signal Processing Workshop, pp. 460-463, 1995.
5. Van Schyndel R. G., Tirkel A. Z., and Osborne C. F., "A Digital Watermark", Proceedings of IEEE Int. Conf. on Image Processing *vol. II*, pp. 86-90, 1994.
6. Cox I., Kilian J., Leighton T., and Shamoon T., "Secure Spread-Spectrum watermarking for Multimedia", Technical Report 95-10, NEC Research Institute, 1995.

7. Hartung F., and Girod B., "Digital Watermarking of Raw and Compressed Video", Proceedings of the SPIE Dig. Comp. Tech. And Systems for Video Communication, vol. 2952, pp. 205-213, Oct. 1996.
8. Craver S., Memon N., Yeo B., and Yeoung M., "Can Invisible Watermarks Resolve Rightful Ownership?" Proceedings of the SPIE, Storage and Retrieval for Image and Video database V, vol.3022, pp. 310-321, 1997.
9. Craver S., Memon N., Yeo B., and Yeoung M., "Resolving rightful Ownerships with Invisible watermarking Techniques: Limitations, attacks, and Implications", IBM Research Report RC20755, March 1997.
10. Swanson M. D., Zhu B., and Tewfik A. H., "Data Hiding for Video-in-Video", IEEE International Conference of Image Conference, vol. II, pp. 676-679, Santa Barbara, Oct. 1997.
11. Chae J. J., and Manjunath B. S., "A Robust Embedded Data from Wavelet Coefficients". Proceedings of the SPIE EI'98, vol. 3312, pp. 308-317, San Jose, Feb.1998.
12. S. Lin, and D. J. Costello, Jr., *Error Control Coding: Fundamentals and Applications*, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, USA, 1983.
13. B. Girod, "The Information Theoretical Significance of Spatial and Temporal masking in Video Signals", Proceedings of the SPIE Human Vision, Visual Processing, and Digital Display, 1989, vol. 1077, pp. 178-187.
14. Vitterli M., and Kovacevic J. *Wavelets and Subband Coding*, Prentice Hall, New Jersey, 1995.
15. Linde Y., Buzo A., and Gray R. M., "An algorithm for vector quantizer design", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp. 553-559, 1989.
16. J. Fridrich, and M. Goljan, "Comparing robustness of watermarking techniques." Proceedings of SPIE (Security and Watermarking for Multimedia Contents), vol. 3657, San Jose, Jan 25-27, 1999.

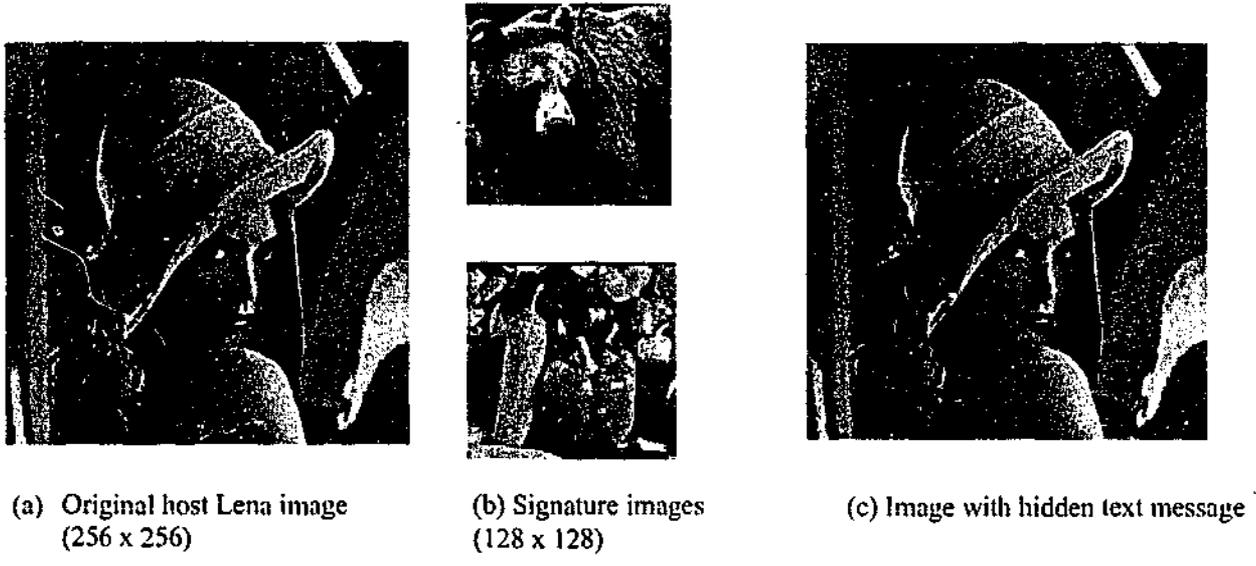


Figure 3: Test images (a) Host Lena image, (b) Bear and Peppers signature images, (c) Image with hidden text message for  $\alpha = 40$ .

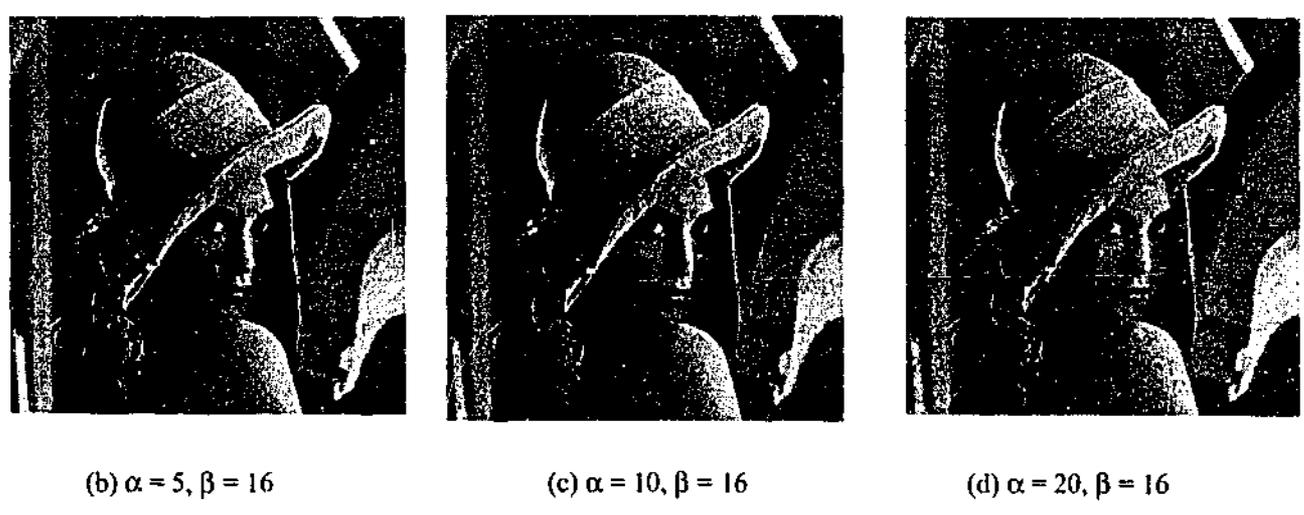


Figure 4: Host Lena with embedded bear image for various scale factors.

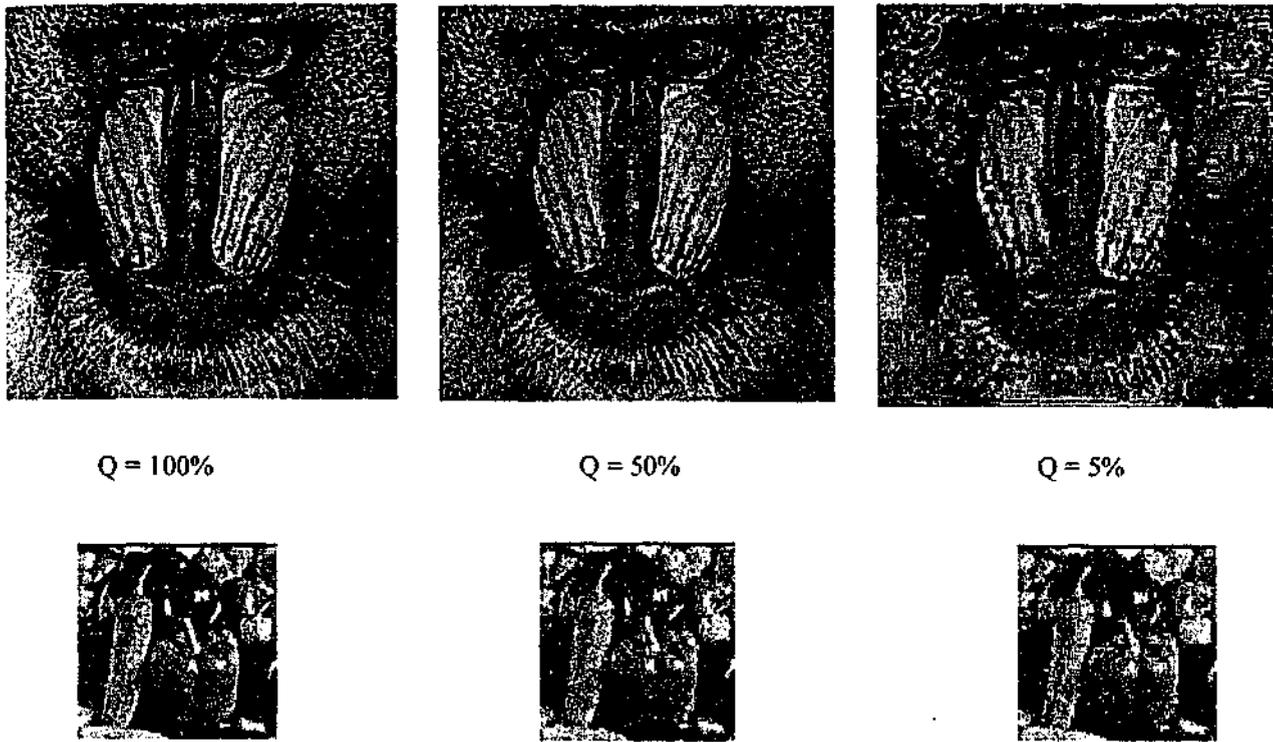


Figure 5 Embedded image using Baboon as the host image and Peppers image as the signature for different quality factor (Q%) of JPEG compression using BCH (7,4) and scale factor  $\alpha = 10$ , and quantization levels  $\beta = 16$ .

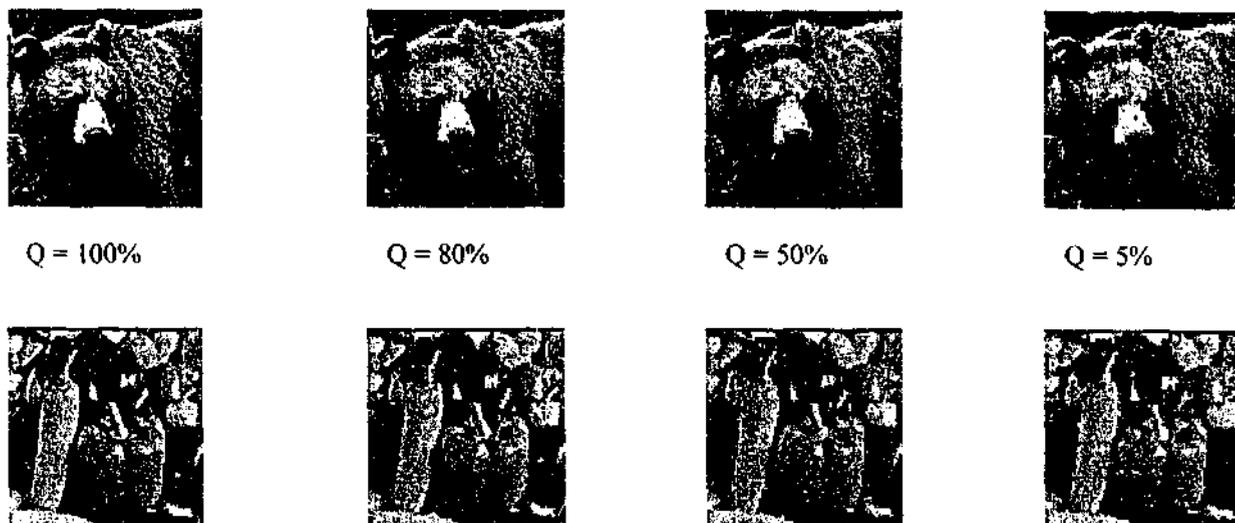


Figure 6: Recovered signature images for different JPEG Quality factor (Q) using Lena image as the host and BCH coding,  $\alpha = 10$ , and  $\beta = 16$ .

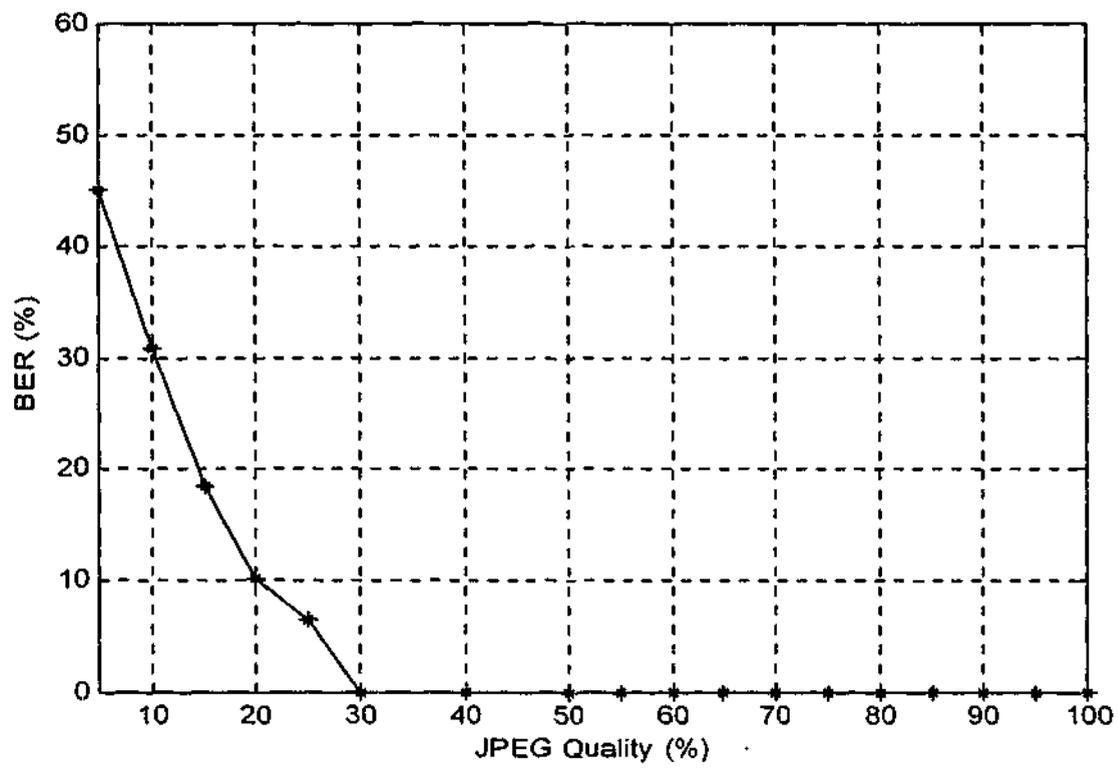


Figure 7: Bit error rate versus JPEG coding at different qualities using Reed-Solomon coding for the hidden text message and  $\alpha = 40$ .

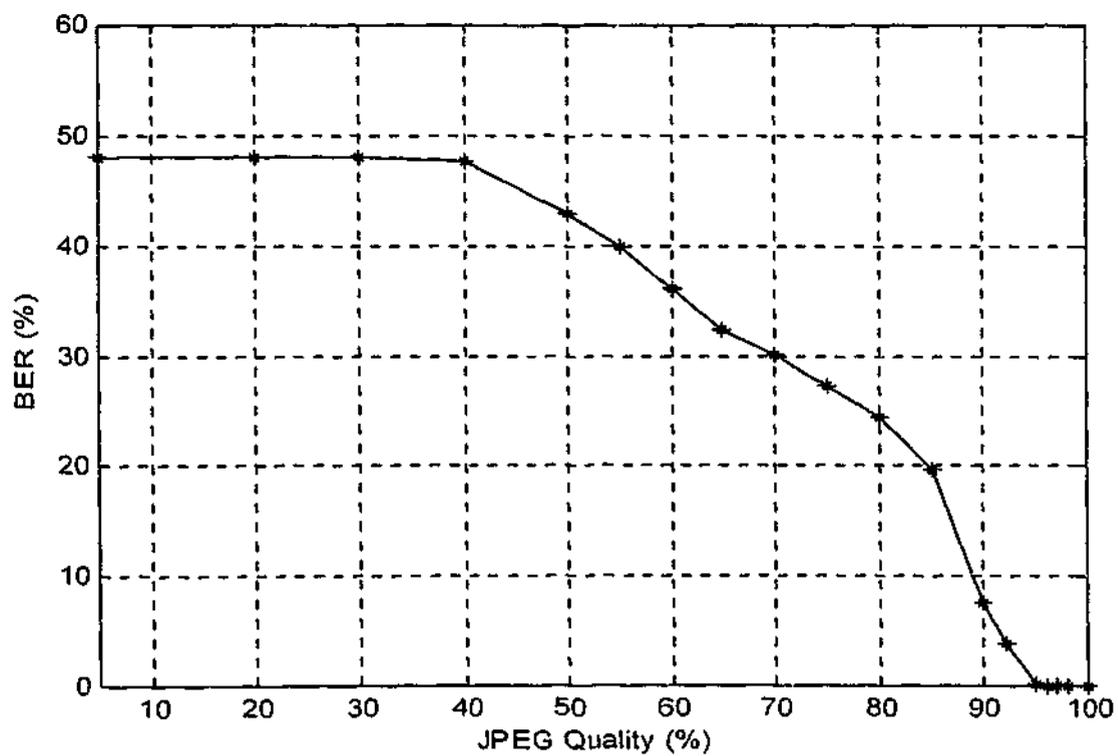


Figure 8: Bit error rate vs JPEG coding at different qualities using BCH code and  $\alpha = 10$ . The hidden signature is bear image (128 x 128).

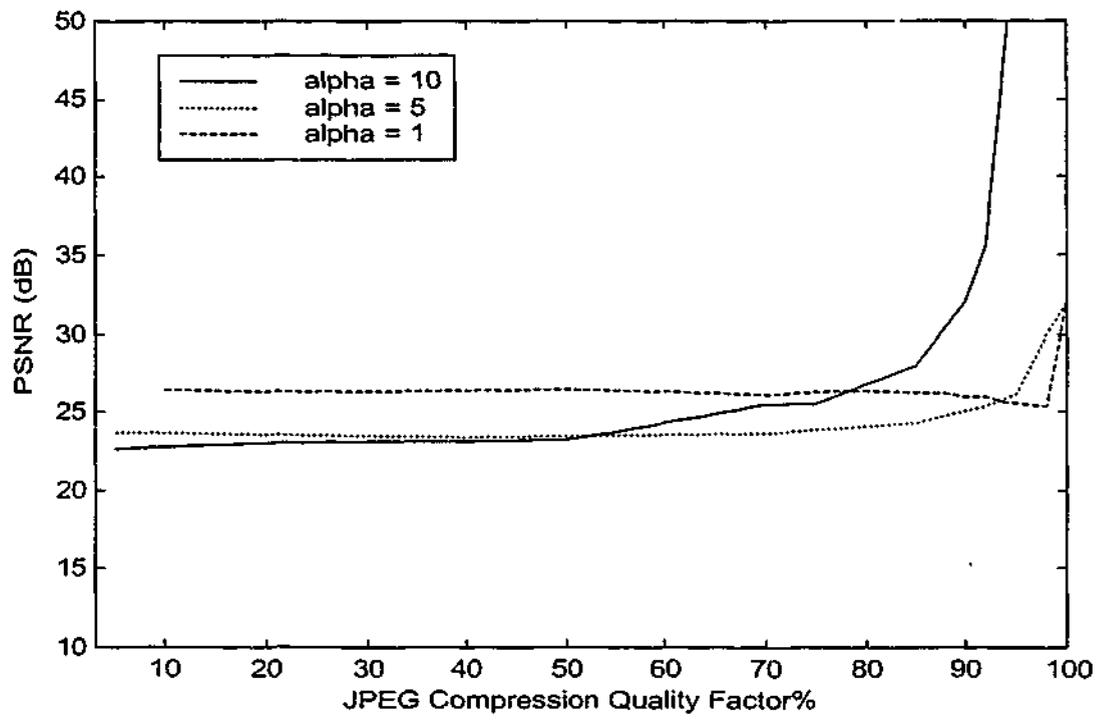


Figure 9: The PSNR of the recovered Bear image for different scaling factor  $\alpha$  and different values of JPEG compression.

# Robust Data Hiding for Images

N. K. Abdulaziz, K. K. Pang  
Dept of Electrical and Computer Systems Engineering  
Monash University  
Clayton, VIC 3168, Australia  
Email: (nidhal.abdulaziz, Khee.k.Pang) @eng.monash.edu.au

## Abstract

This paper describes a robust data embedding scheme, which uses a source and channel coding framework for data hiding. The data to be embedded, referred to as the signature data, is source coded by vector quantization and the indices obtained in the process are embedded in the transform coefficients of the host image. Transform coefficients of the host are grouped into vectors and perturbed using error-correcting codes derived from BCH codes. Compared to prior work in digital watermarking, the proposed scheme can handle a significantly large quantity of data such as a gray scale images. A trade-off between the quantity of hidden data and the quality of the watermarked image is achieved by varying the number of quantization levels for the signature, the code word length, and the scale factor for embedding. Experimental results on signature recovery from JPEG compressed watermarked images are included.

## Keywords

Digital watermarking, data hiding, wavelet transform, error-correcting codes, vector quantization.

## 1. Introduction

Motivated by the overwhelming desire for Internet data security, digital watermarking has recently emerged as an important area of research in multimedia data processing [1]. A digitally watermarked image is obtained by invisibly hiding a signature information into the host image. The signature is recovered using an appropriate decoding process. The challenge is to simultaneously ensure that the watermarked image is perceptually indistinguishable from the original, and that the signature be recoverable even when the watermarked image has been compressed or transformed by standard image processing operations. Several interesting data hiding techniques for images have been proposed. The most common approach is to add fixed amplitude noise to the host image [2][3][4][5][6]. All of the schemes described utilize the fact that digital media contain perceptually insignificant components, which may be replaced or modified to

embed data. While most of the research on watermarking concentrates on copyright protection in internet data distribution [7][8], a different kind of watermarking, commonly known as data hiding, is at present receiving considerable attention. Data hiding is intended to hide larger amounts of data into host source, rather than just to check for authenticity and copyright information [9][10]. In other words, the problem of watermarking or copyright protection is a special case of the generic problem of data hiding, where a small signature is embedded with greater robustness to noise. This paper proposes a robust data hiding technique using channel codes derived from the error-correcting codes. In particular we use BCH error-correcting codes with different code words where a gray-scale image is embedded by perturbing the host wavelet coefficients. In this paper, we use the vector quantization scheme for compressing the signature image, the compressed indices are injected into the wavelet coefficients of the host image in a vector based perturbation and the watermarked host is subjected to JPEG compression for manipulation of the watermarked image before attempting retrieval. As experimental-results indicate, there are no visible distortions in the watermarked image, and the recovered signature is similar to the original signature even after 5% JPEG lossy compression quality factor.

In the next section a discussion of the proposed embedding and extracting algorithm is presented.

## 2. Data embedding

It is well known that embedding in the low-frequency bands is more robust to manipulations such as enhancement and image compression. However, changes made to the low frequency components may result in visible artifacts. Modifying the data in a multiresolution framework, such as a wavelet transform [11], appears quite promising for obtaining good quality embedding with little perceptual distortion. Figure 1 shows a schematic diagram of the embedding procedure. In this work, a vector-based approach is adopted to hidden data injection, where a

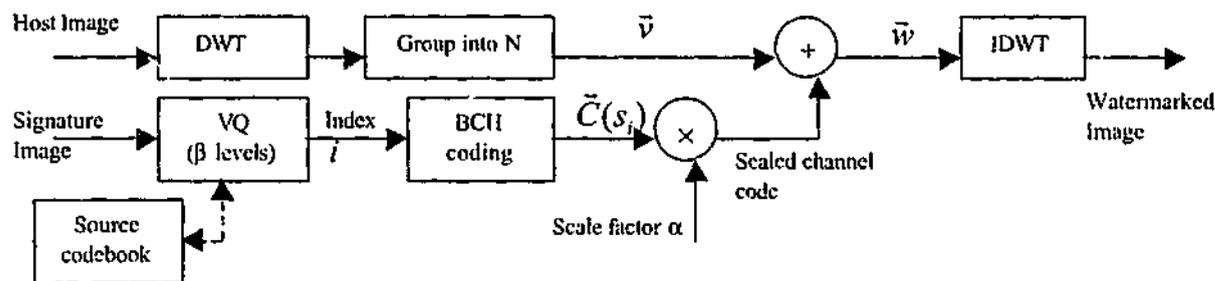


Figure 1. Block diagram of the encoder in the embedding state.

group of  $N$  transformed coefficients are used to form an  $N$ -dimensional vector. These vectors are modified by codes that represent the data to be embedded. The codes used for channel coding are error-correcting codes. Error correcting codes provide coders with a tool to recover lost information such as errors, erasures and deletions. In this work, block codes known as the BCH code was used. The coefficient vectors perturbed in our implementation are of dimension  $N$  and the channel code used to embed the data is BCH codes of  $(N, K)$  [12]. In this algorithm, a single level of the discrete wavelet transform (DWT) decomposition of the host image is made before data embedding. The signature image is quantized using vector quantization with Linde-Buzo-Gray (LBG) algorithm [13]. Vector quantization (VQ) transforms the vectors of data into indices that represents the clusters of vectors. For BCH code of  $(7,4)$ , the signature image is decomposed into 4-dimensional image vectors; in this case the signature image is divided into  $2 \times 2$  blocks. Each vector is compared with a collection of representative codevectors taken from a previously generated codebook (the source codebook). Best match codevector is chosen using a minimum distortion rule. After the minimum distortion codevector has been found the index  $i$  is used to represent the signature vector. In order to embed the indices, the DWT coefficients of the host image is grouped to form an  $N$ -dimensional vector, and the vector is then perturbed by the coded indices after it has been scaled by a factor  $\alpha$ . If  $\bar{v}$  represents a vector of host DWT coefficients after grouping, and the index of the vector quantized signature image is  $i$ , then the perturbed vector  $\bar{w}$  is given by:

$$\bar{w} = \bar{v} + \alpha \cdot \bar{C}(s_i)$$

where  $\bar{C}(s_i)$  represent the channel code (BCH code) corresponding to the symbol  $s_i$  where  $i = 1, \dots, \beta$ .

Each index of the signature image is hidden into  $N$  coefficients in the LL band of the host; the remaining indices are hidden in the other subbands of the host (HL, LH, and HH). The scale factor for embedding is

chosen in a way that assure an acceptable quality for the watermarked image. The choice of the parameters  $\alpha$  and  $\beta$  determines the trade-off between the transparency and the quality of the hidden data. For security in copyright protection, we can select special regions in the transform domain to embed data, or randomly group the coefficients to form a vector using a private key. It is to be noted, however, that in general, the less the quantity of data hidden, and the more secure it can be made.

### 3. Extracting data

If the original host image is available, then the operation of data injection and retrieval are, in fact, very similar to the channel coding and decoding operations in a typical digital communication systems. In watermarking in the transform domain, the original host data is transformed, and the transformed coefficients are perturbed by a small amount in one of several possible ways in order to represent the signature data. When the watermarked image is compressed or modified by image processing operations, this is equivalent to adding noise to the perturbed coefficients. The retrieval operation subtracts the received coefficients from the original ones to obtain the noisy perturbations. The true perturbations that represent the injected data are then estimated from the noisy data as best as possible.

Recovering the hidden data starts with the same DWT of the received watermarked image that was used to embed the data. The true host image coefficients (known to the receiver) are then subtracted from the coefficients of the received image to obtain the noisy perturbations. Note that these perturbations can be "noisy" because of various possible transformations of the watermarked data. These coefficients are now grouped into groups of  $N$  in the same manner as they were grouped during encoding to obtain a vector  $\bar{e}$ , and then scaled by the factor  $1/\alpha$ . The resulting vector  $1/\alpha \bar{e}$  is then BCH decoded to find the index  $i$ . From the index  $i$ , and using a duplicate codebook and a table lookup, the signature image can be recovered. Figure 2

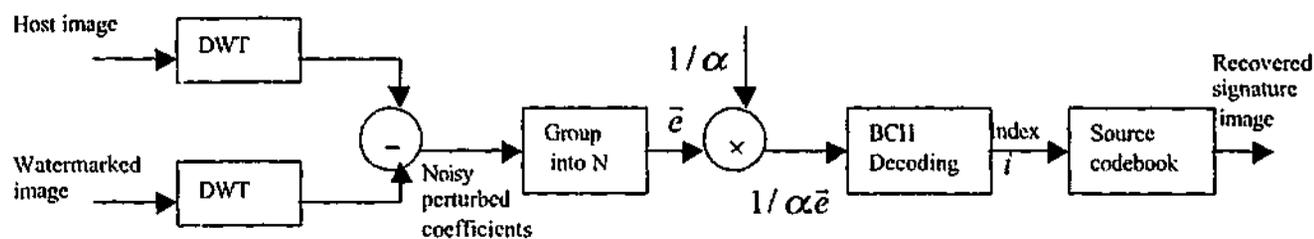


Figure 2. Block diagram of the decoder.

shows the details of symbol recovery and signature extraction.

#### 4. Experimental results

The test images used in this work are shown in Figure 3. The host image, Lena a 256 x 256 gray scale, and the signature images Bear, and peppers, all are 128 x 128 gray scale. A 1-stage discrete Haar wavelet transform is used for both the encoder and the decoder in this work.

Figure 4 shows Lena image watermarked with bear image using LBG vector quantizer with blocks of 2 x 2 and 16 levels and BCH coding of (7,4) at various scale factor, without any compression. Note that the scale factor  $\alpha$  controls the relative weight of host and signature image contributions to the fixed image. As the value of  $\alpha$  increases, the quality of the watermarked image degrades. For example, in Figure 4, one can see artifacts in the background for  $\alpha = 20$ .  $\alpha = 10$  appear to be a reasonable value in terms of the trade-off between quality of the watermarked image and robustness to signature recovery under image compression.

Figure 5 shows the signature images recovered from the watermarked image after 100%, 80%, 50%, and 5% JPEG quality factor. In general, most of the recovered signature images are of high quality, when the scale factor  $\alpha = 10$ . Figure 6 shows the PSNR of the recovered bear image for different values of JPEG compression. The quality of the recovered signature with a large scale factor  $\alpha$  is obviously much better than those with a smaller  $\alpha$ . On the other hand, the number of quantization levels  $\beta$  determines the coarseness of quantization and therefore the quality of the signature image hidden in the host.

#### 5. Conclusions

The technique presented here is a new technique that allows the correction of channel errors due to compression attack; it also has the ability to hide large amount of data into the host image. There are still some work to be done and questions to be answered, among others:

- Does the method perform better with longer codes? If so, what are the most appropriate?
- Is there an optimum in the trade-off among the scale factor  $\alpha$ , quantization levels  $\beta$ , and channel code length (N,K).

Moreover, the watermarked image need to be subjected to other attacks, for example, noise addition, filtering and other image processing techniques and see how it will affect the recovered signature image.

#### 6. References

1. Petitcolas F. A. P., Anderson R. J., and Kuhn M. G., "Information hiding - a survey", Proceedings of the IEEE, vol. 87, no. 7, pp. 1062--1078, July 1999.
2. Bender W., Gruhl D., and Morimoto N., "Techniques for Data Hiding", Technical. Report, MIT MediaLab, 1994.
3. Pitas I., and Kaskalis T., "Applying Signatures on Digital Images", Proceedings 1995 IEEE Nonlinear Signal Processing Workshop, pp. 460-463, 1995.
4. Van Schyndel R. G., Tirkel A. Z., and Osborne C. F., "A Digital Watermark", Proceedings of IEEE Int. Conf. on Image Processing vol. II, pp. 86-90, 1994.
5. Cox I., Kilian J., Leighton T., and Shamoon T., "Secure Spread-Spectrum: watermarking for Multimedia", Technical Report 95-10, NEC Research Institute, 1995.
6. Hartung F., and Girod B., "Digital Watermarking of Raw and Compressed Video", Proceedings of the SPIE Dig.Comp. Tech. And Systems for Video Communication, vol. 2952, pp. Tech. And Systems for Video Communication, vol. 2952, pp. 205-213, Oct. 1996.
7. Craver S., Memon N., Yeo B., and Yeoung M., "Can Invisible Watermarks Resolve Rightful Ownership?" Proceedings of the SPIE, Storage and Retrieval for Image and Video database V, vol.3022, pp. 310-321, 1997.
8. Craver S., Memon N., Yeo B., and Yeoung M., "Resolving rightful Ownerships with Invisible watermarking Techniques: Limitations, attacks, and aimplications," IBM Research Report RC20755, March 1997.

9. Swanson M. D., Zhu B., and Tewfik A. H., "Data Hiding for Video-in-Video", IEEE International Conference of Image Conference, vol. II, pp. 676-679, Santa Barbara, Oct. 1997.

10. Chae J. J., and Manjunath B. S., "A Robust Embedded Data from Wavelet Coefficients", Proceedings of the SPIE EI'98, vol. 3312, pp. 308-317, San Jose, Feb.1998.

11. Vitterli M., and Kovacevic J. wavelets and Subband Coding, Prentice Hall, New Jersey, 1995.

12. S. Lin, and D. J. Costello, Jr., Error Control Coding: Fundamentals and Applications, Prentice-Hall, Inc. Englewood Cliffs, New Jersey, USA, 1983.

13. Linde Y., Buzo A., and Gray R. M., "An algorithm for vector quantizer design", IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, pp. 553-559, 1989.

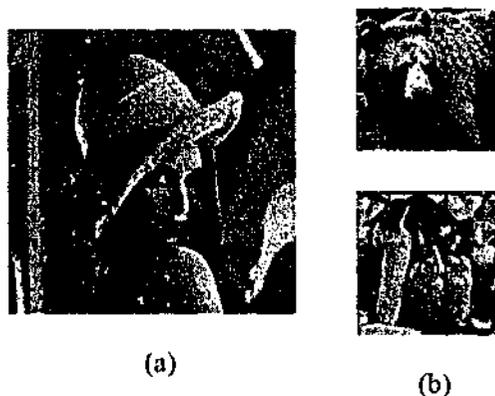


Figure 3: Test images (a) Host Lena image, (b) Bear and Peppers signature images.

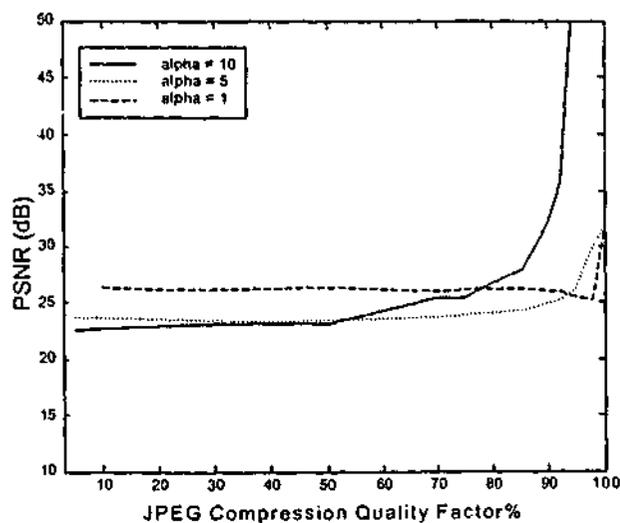


Figure 6: The PSNR of the recovered bear image for different values of JPEG compression and BCH(7,4).

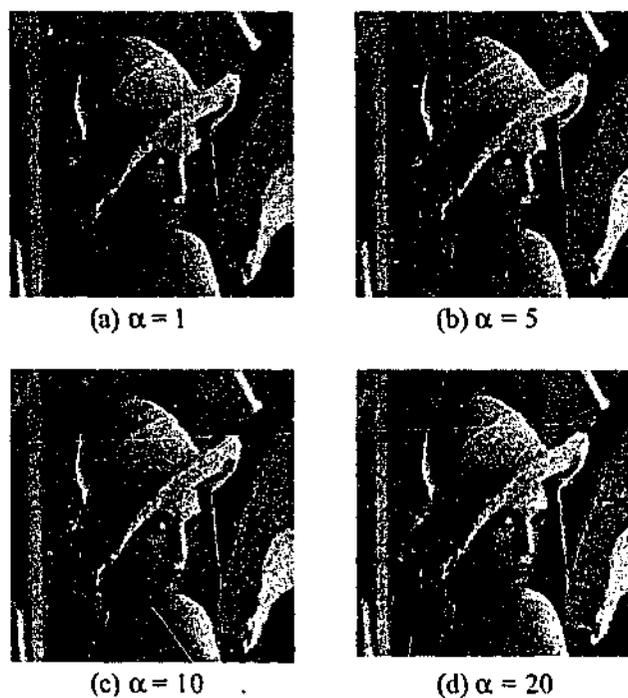


Figure 4: Host Lena with embedded bear image for various scale factors and  $\beta = 16$ .

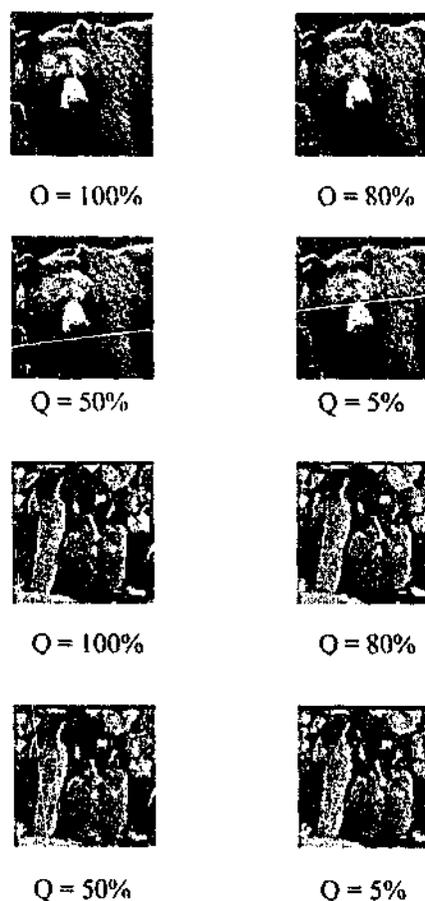


Figure 5: Recovered signature images for different JPEG Quality factor (Q).

# PERFORMANCE EVALUATION OF DATA HIDING SYSTEM USING WAVELET TRANSFORM AND ERROR-CONTROL CODING

*N. K. Abdulaziz and K. K. Pang*

Dept of Electrical and Computer Systems Engineering  
Monash University

Clayton, VIC 3168, Australia

Email: (nidhal.abdulaziz, khee.pang) @eng.monash.edu.au

## ABSTRACT

This paper proposes an algorithm of data hiding for image signal based on the wavelet transform and error-correcting codes. The data to be embedded, referred to as the signature data, is coded using error-correcting codes and the resulting code is hidden into the wavelet transformed coefficients of the host image in a vector based perturbation. When the amount of hidden data is large, as it is the case when the signature data is an image, the signature data is first compressed using vector quantization and the indices obtained in the process are embedded. Information data are detected using both original and watermarked images. Simulation results demonstrates the high robustness of the algorithm to image degradations such as JPEG and additive noise.

## 1. INTRODUCTION

Digital watermarking and information embedding systems have a number of important multimedia applications [1,2]. Many of these applications relate to copyright notification and enforcement for multimedia content such as audio, video, and images that are distributed in digital formats. In data hiding, the focus is on hiding large amounts of data in a host, for a wider range of applications than just copyright protection. Data hiding has several applications, such as transmission of secret information over an insecure but available medium such as the Internet. Another application is in secure transmission of control information along with data in a commercial delivery system. In general, data hiding makes possible invisible mixing of different kinds of secure data transmission, allowing only authorized to retrieve the additional hidden information.

Previous work on embedding invisible signatures can be grubbed into spatial domain and transform domain methods. Targeted applications include watermarking for copyright protection or authentication. Typically, the data used to represent the digital watermarks are a very small fraction of the host image data. Such

signatures include, for example, pseudo-random numbers, trademark symbols, and binary images. Much work has also been done in modifying the data in the transform domain. These include DCT domain techniques [3] and wavelet transforms [4,5,6]. This paper presents a data embedding scheme that is suitable for both watermarking and image data hiding. While watermarking requires robustness to image manipulation, data hiding requires that there is very little visible distortion in the host image. While much of the previous work used signature data that is a small fraction of the host image data, the proposed approach can easily handle gray-scale images that could be as much as 25% of the host image. In some of the recent work on using wavelets for digital watermarking, the signatures were encoded in high and middle frequency bands. Such an embedding is sensitive to operations such as low pass filtering, JPEG lossy compression, and the Laplacian removal attack which has been found to be effective against several digital watermarking schemes that modify the mid to high frequency spectral components of the original image [7]. In contrast, the proposed scheme here focuses on hiding the signature mostly in the low frequency DWT bands, and stable reconstruction can be obtained even when the images are transformed, or otherwise modified by low pass filtering operations.

The paper is organized as follows: the next section describes the proposed algorithm in detail and experimental results are provided in section 3. Discussions are concluded in section 4.

## 2. THE PROPOSED METHOD

The embedding procedure is explained by means of the diagram in Figure 1. The data to be embedded is first source coded, either losslessly or lossily depending on the nature of the data to generate a sequence of symbols. For hiding large amount of data, as it is the case for hiding images, the signature image is first quantized using vector quantization with Linde-Buzo-Gray (LBG) algorithm. Vector quantization (VQ) transforms the vectors of data into indices ( $i$ ) that

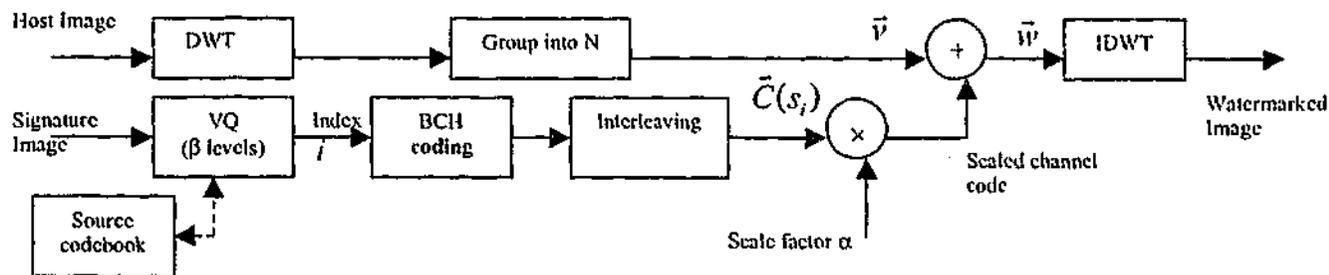


Figure 1. Encoder.

represents the clusters of vectors. Details of this procedure can be found in [8]. These indices are then BCH coded, interleaved, and scaled before adding it to the host DWT coefficients. The host image is wavelet transformed and a group of  $N$  transformed coefficients are used to form an  $N$ -dimensional vector. These vectors are then modified by the coded indices after it has been scaled by a factor  $\alpha$ . The parameter  $\alpha$  determines the transparency constraint. That is, if  $\vec{v}$  represents a vector of host DWT coefficients after grouping, and the index of the vector quantized signature image is  $i$ , then the perturbed vector  $\vec{w}$  is given by:

$$\vec{w} = \vec{v} + \alpha \vec{C}(s_i) \quad (1)$$

where  $\vec{C}(s_i)$  represent the channel code corresponding to the symbol  $s_i$ , where  $i = 1, \dots, \beta$ . Each index of the signature image is hidden into  $N$  coefficients in the LL band of the host; the remaining indices are hidden in the other subbands of the host (HL, LH, and HH). The scale factor for embedding is chosen in a way that assure an acceptable quality for the watermarked image. The scale factor alpha controls the relative amount of signature data used in the embedding scheme.

Recovering the hidden data starts with the same DWT of the received watermarked image that was used to embed the data. The true host image coefficients (known to the receiver) are then subtracted from the coefficients of the received image to obtain the noisy perturbations. Note that these perturbations can be "noisy", because of various possible transformations of the watermarked data. These coefficients are then grouped into groups of  $N$  in the same manner as they were grouped during encoding to obtain a vector  $\vec{e}$ , which is scaled by the factor  $1/\alpha$ . The resulting vector  $1/\alpha \vec{e}$  is then de-interleaved and BCH decoded to find the index  $i$ . From the index  $i$ , and using a duplicate codebook and a table lookup, the signature data can be recovered. Figure 2 shows the details of symbol recovery and signature extraction.

### 3. EXPERIMENTAL RESULTS

The test images used in this paper are: "Lena", "Baboon", and "Fishingboat". All images were gray scale images with  $256 \times 256$  pixels. Two types of signature were used, image data (Bear image), of size  $128 \times 128$  pixels gray scale and ASCII text file of length 890 bits. All the experiments described below use the discrete Haar wavelet basis. To measure the robustness, we used the bit-error rate, the peak signal-noise ratio PSNR of the recovered watermark image, the similarity measure, and the attack is JPEG compression and noise addition.

**Bit-Error versus Attack Strength Graph:** This graph relates the watermark robustness to the attack, where the bit-error rate is plotted as a function of the attack strength for a given visual quality. This evaluation allows the direct comparison of the watermark robustness and shows the overall behaviour of the method towards attacks. Figure 3 shows this graph for our example. In this figure, a plot of bit error (BER) for different levels of JPEG coding at different quality factors is shown for hiding Bear image into the three test images. The algorithm works well under high quality coding conditions yet degrades more rapidly when the coding becomes too lossy.

Similarly Figure 4 shows the plot of the PSNR of the recovered Bear image versus JPEG coding at different quality settings.

The bit error rate for embedding text message is shown in Figures 5 as a function of JPEG compression and in Figure 6 as a function of noise addition.

In checking for the presence of a signature, the quality of the signature is not an issue. A binary decision for the presence or absence of a signature needs to be made. We use a measure similar to the one defined in [3] to compute the cross correlation between the recovered signature  $\hat{s}(m, n)$  and the original signature  $s(m, n)$  in the wavelet transform domain. This similarity is defined as:

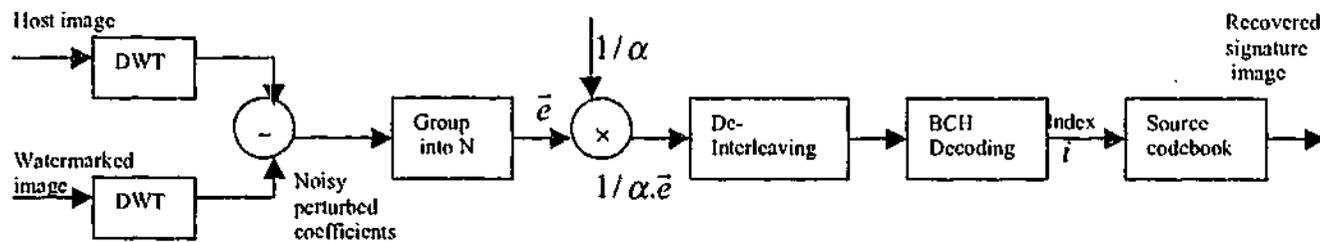


Figure 2. Decoder.

$$S = \frac{\sum_{m,n} s^*(m,n)s(m,n)}{\sum_{m,n} (s^*(m,n))^2} \quad (2)$$

A graph of this similarity for varying JPEG compression is shown in Figure 7, as can be seen from this graph, it is easy to find a threshold for signature detection between unwatermarked and watermarked images. Moreover, Figure 8 shows the bit error rate for both watermarked and unwatermarked Lena image for the case of hiding text message.

#### 4. DISCUSSION

In this method, a scheme of embedding large amount of data is presented. This approach could be used for both digital watermarking related applications as well as for data hiding purposes. The scale factor controls the relative amount of host and signature data in the embedded image. Experimental results demonstrate that good quality signature recovery and authentication is possible when the images are JPEG compressed and under noise addition.

The method performs very well against the JPEG compression attack and achieves 0% BER for quality factor as low as 30% as can be seen from Figure 5. There are relatively little differences in performance for different images. Overall, the method performs better in JPEG compression attack than in noise addition.

#### 5. REFERENCES

- [1] M. D. Swanson, M. Kobayashi, and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies", Proceedings of the IEEE, vol. 86, no. 6, June 1998, pp. 1064-1087.
- [2] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding-A Survey", proceedings of the IEEE, Special Issue on Protection of Multimedia Contents, vol. 87, no. 7, July 1999, pp. 1062-1078.
- [3] I. J. Cox, J. Kilian, T. Lieighton, and T. Shamoan, "Secure Spread Spectrum 6, no. Watermarking for Multimedia", IEEE Transactions on Image Processing, vol. 12, December 1997, pp. 1673-1687.
- [4] H.-J. M. Wang, P.-C. Su, and C.-C. Jay Kuo, "Wavelet-based digital image watermarking", The Electronic Journal for the Optical Society of America, Optics Express, vol. 3, no. 12, December 1998, pp. 491-496.
- [5] X.-G. Xia, C. G. Bonchelet, and G. Arce, "Wavelet transform based watermark for digital images", The Electronic Journal for the Optical Society of America, Optics Express, vol. 3, no. 12, December 1998, pp. 497-511.
- [6] D. Kundur, and D. Hatzinakos, "A robust digital image watermarking method using wavelet-based fusion," Proceedings of the IEEE International Conference on Image Processing, Oct. 26-29, 1997, Santa Barbara, CA, vol. 1, pp. 544-547.
- [7] R. Barnett, and D. E. Pearson, "Frequency mode LR attack operator for digitally watermarked images," Electronics Letters, vol. 34, no. 19, 17<sup>th</sup> September 1998, pp. 1837-1839.
- [8] N. K. Abdulaziz, and K. K. Pang, "Source and channel coding approach to data hiding", To be presented at the conference Visual Communications and Image processing 2000, 20-23 June 2000, Perth.

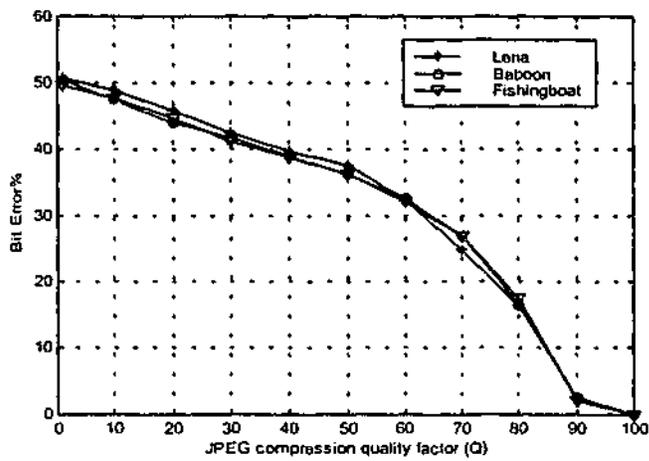


Figure 3. Bit error rate versus JPEG coding using Bear image as the signature.

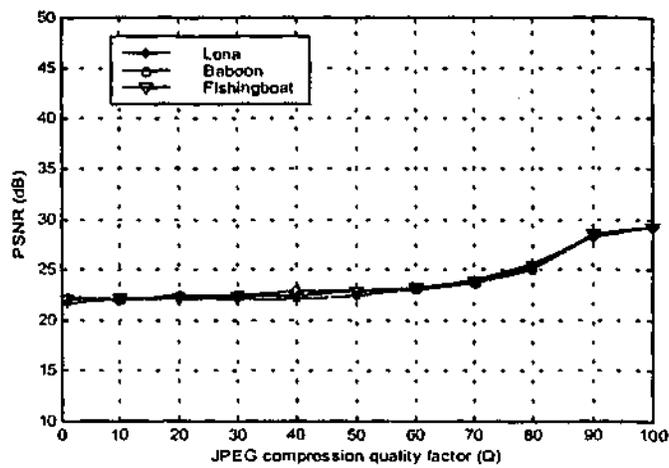


Figure 4. PSNR of recovered Bear image versus JPEG coding.

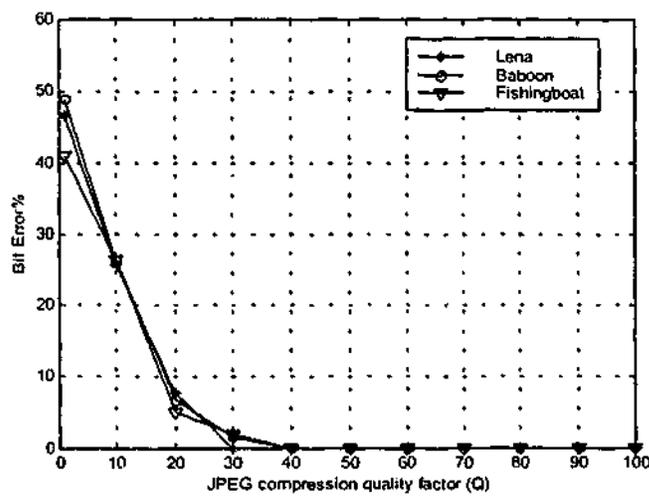


Figure 5. Bit error rate versus JPEG coding using text message as the signature.

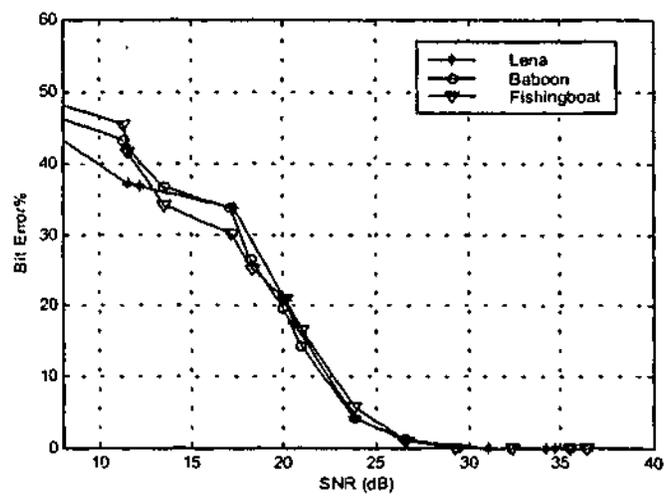


Figure 6. Bit error rate versus SNR using text message as the signature.

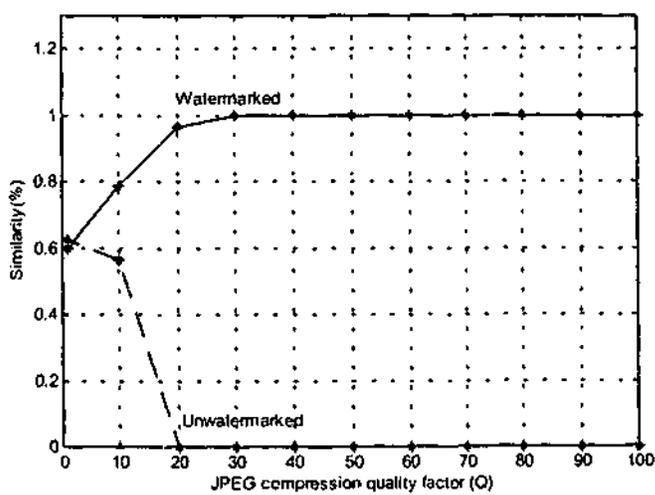


Figure 7. Similarity measure for watermarked and unwatermarked images.

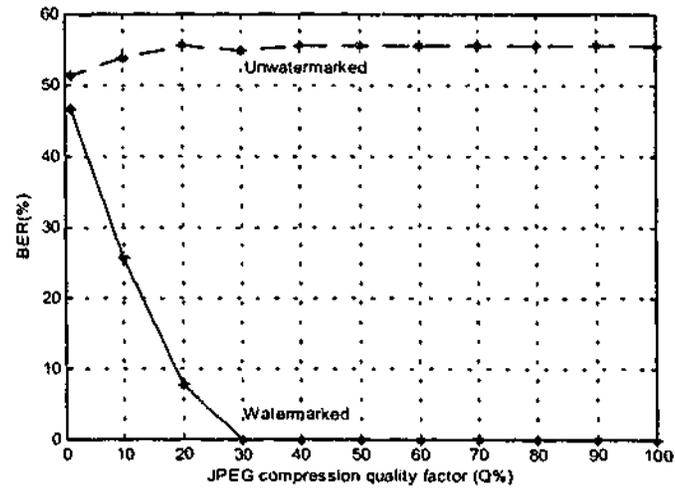


Figure 8. Bit error rate for watermarked and unwatermarked images.

# DATA EMBEDDING USING TRELLIS CODING

N. K. ABDULAZIZ AND K. K. PANG  
Dept of Electrical and Computer systems Engineering  
Monash University, Clayton,  
Australia

## ABSTRACT

An approach to embedding gray scale images using a discrete wavelet transform and trellis coding is proposed. The proposed scheme enables using signature images that could be as much as  $\frac{1}{4}$  of the host image data and hence could be used both in digital watermarking as well as image/data hiding. In digital watermarking the primary concern is the recovery or checking for signature even when the embedded image has been changed by image processing operations. Thus the embedding should be robust to typical operations such as low-pass filtering and lossy compression. On the other hand, for data hiding applications it is important that there should not be any visible changes to the host data that is used to transmit a hidden image. Moreover, it is desirable that it is difficult or impossible for unauthorized persons to recover the embedded signatures. The proposed method make use of channel codes that are implemented in digital communication for fading channels such as trellis codes. Experimental results demonstrate that high quality recovery of the signature data is possible and that it is robust to operations such as JPEG compression and noise addition.

**Keywords:** digital watermarking, data hiding, trellis coding.

## 1. INTRODUCTION

As multimedia data, such as images audio and video, becomes wide spread on the Internet, there is a need to address issues related to the security and protection of such data [1,2,3]. Digital watermarking is one approach to managing this problem by encoding user or other copyright information directly in the data. In order to be effective, an invisible watermark should be secure, reliable, and resistant to common signal processing operations and intentional attacks. Recovering the signature from the watermarked media could be used to identify the rightful owners and the intended recipients as well as to authenticate the data.

Previous work on embedding invisible signature can be broadly grouped into spatial domain and transform domain methods. Targeted applications include watermarking for copyright protection or authentication. Typically the data used to represent the digital watermarks are very small fraction of the host image data. Such signatures include, for example, pseudo-random numbers, trademark symbols and binary images. Spatial domain methods usually modify the least-significant bit of the host image [3,4], and are, in general, not robust to operations as low-pass filtering. Much work has also been done in modifying the data in the transform domain. These include DCT domain techniques [5, 6], and wavelet transforms [7, 8]. While much of the previous work used signature data that is a small fraction of the host image data, the proposed approach can easily handle grey-scale images that could be as much as 25% of the host image. In recovering the signature image, it is assumed that the original host image is available.

The proposed scheme distributes the signature information in the discrete wavelet transform (DWT) domain of the host image. Spatial distribution of the DWT coefficients helps to recover the signature even when the images are compressed using JPEG lossy compression. In some of the recent work on using wavelets for digital watermarking, the signatures were embedded in all DWT bands. Such an embedding is sensitive to operations that change the high frequency content without degrading the image quality significantly. Examples of such operations include low pass filtering for image enhancement and JPEG lossy compression. In contrast, the proposed scheme here focuses on hiding the signature mostly in the low frequency DWT bands, and stable reconstruction can be obtained even when the images are transformed, quantized (as in JPEG), or otherwise modified by enhancement or low pass filtering operations. Moreover, the algorithm makes use of channel coding that is often implemented to overcome fading in mobile communications such as trellis coding. The paper is organized as follows: the next section describes how data embedding is similar to digital communication system. The embedding and extraction algorithm of data is explained in detail in sections 3 and 4 and experimental results are provided in section 5. Discussions are concluded in section 6.

## 2. PARRALLEL WITH DATA COMMUNICATION

Digital watermarking of multimedia can be viewed as a communications problem. A message (information to be embedded) is converted into a signal (the watermark), which is then sent through a channel to the receiver. The receiver must locate the watermark signal and attempt to recover the message from it. The channel is referred to as the watermark-channel to distinguish it from a conventional broadcast channel.

An attack is an operation, performed on the watermarked document, that may degrade a watermark and possibly make the watermark unreliably detected. From a communications point of view, even coincidental manipulations such as lossy compression or cropping, are attacks. Attacks are assumed to occur only in the channel. While there are various noise models available for various kinds of channels, we will assume that the noise is of an Additive White Gaussian Noise (AWGN). This particular noise model approximates many real channels and also makes analysis simpler. The task of the receiver is then to estimate the symbols transmitted from the noisy waveform that is received. In a noise free channel, the received signal point is exactly the same point as the one transmitted. However, as a result of noise, the received vector is different from the one transmitted. Using a maximum-likelihood decoder in then yields a decoding rule, which for every vector received, chooses the symbol to whose channel code is closest in Euclidean distance.

## 3. EMBEDDING PRINCIPLE

In this section we explain the generic embedding principle by means of the diagram in Figure 1. The secure data is first source-coded either losslessly or lossily depending on the nature of the data, to generate a sequence of symbols. The embedding process injects one symbol into each coefficient vector  $v_j$  of the DWT coefficients of the host image. The coding is usually obtained from a noise-resilient channel code by scaling it by a parameter  $\alpha$ , which determines the transparency constraint. That is, the perturbed vectors  $\tilde{v}_j$  are obtained as follows:

$$\tilde{v}_j = v_j + \alpha C(s_j),$$

where the set of vectors  $C(s_j)$ , constitute a channel codebook. The perturbed coefficients are inverse transformed back to the host before transmission or distribution.

For hiding large amount of data, as it is the case for hiding images, the signature image is first quantized using vector quantization with Linde-Buzo-Gray (LBG) algorithm. Vector quantization (VQ) transforms the vectors of data into indices ( $i$ ) that represents the clusters of vectors. Details of this procedure can be found in [9]. For reliable extraction of the indices, Several channel codes could be implemented. In previous work we have investigated the use of block codes such as BCH and Reed-Solomon codes as error-control coding to improve the performance of the embedding algorithm and results can be found in [10, 11]. In this work, trellis codes is used as the channel codes. These codes can be generated from convolutional coding together with set partitioning which is known as trellis coded modulation (TCM). It is worth noting here that only the mapping and not the modulation is used in the embedding. The TCM used comprises of a convolutional coder with a rate of  $\frac{1}{2}$ . The inputs bits  $k = 2$ , one bit is used as input to the convolutional coder and the other bit is left uncoded. The output bits  $n = 3$ , is then mapped to 8 signal point constellation that is used as the channel code for the signature data. Interleaver is used to convert burst errors into random errors.

## 4. EXTRACTION PRINCIPLE

The extraction principle is outlined in Figure 2. Let us say that the  $j$ th perturbed vector  $\tilde{v}_j$ , corresponding to a hidden symbol  $s_j$ , has been received as  $w_j$ , as a result of additive noise  $n_j$  due to compression and other transformations:

$$w_j = \tilde{v}_j + n_j$$

The process of extraction is then formulated as a statistical estimation problem that estimates the transmitted symbol from the noisy version received. The extraction process uses its knowledge of the original host to decode, from each received vector, the symbol within whose decision boundaries the received perturbation lies. In other words, a nearest neighbor search with an appropriate distance measure is used. The sequence of the extracted symbols are then source-decoded to obtain the extracted watermark.

Recovering the hidden data starts with the same DWT of the received watermarked image that was used to embed the data. The true host image coefficients (known to the receiver) are then subtracted from the coefficients of the received image to obtain the noisy perturbations. Note that these perturbations can be "noisy", because of various possible transformations of the watermarked data. These coefficients are then grouped into groups of  $N$  in the same manner as they were grouped during encoding to obtain a vector  $\tilde{x}$ ,

which is scaled by the factor  $1/\alpha$ . The resulting vector  $1/\alpha \cdot \bar{z}$  is then de-interleaved and Viterbi decoded to find the index  $i$ . From the index  $i$ , and using a duplicate codebook and a table lookup, the signature data can be recovered.

## 5. EXPERIMENTAL RESULTS

The test images used in this paper are gray scale image of "Lena" with 256 x 256 pixels. Two types of signature were used, image data (Bird image), of size 128 x 128 pixels gray scale and random data of length 2000 bits. All the experiments described below use the discrete Haar wavelet basis. To measure the robustness, we used the bit-error rate and the peak signal-noise ratio PSNR of the recovered watermark image, the attack is JPEG compression and noise addition.

**Bit-Error versus Attack Strength Graph:** This graph relates the watermark robustness to the attack, where the bit-error rate is plotted as a function of the attack strength for a given visual quality. This evaluation allows the direct comparison of the watermark robustness and shows the overall behavior of the method towards attacks. It also shows the improvement of coded data over uncoded messages. Figure 3 shows this graph for our example. In this figure, a plot of bit error (BER) for different levels of JPEG coding at quality factor  $\alpha = 10$  is shown for hiding random data into the test image for both coded and uncoded cases. The algorithm works well under high quality coding conditions yet degrades more rapidly when the coding becomes too lossy.

The bit error rate for embedding random message is shown in Figures 4 as a function of noise addition for both coded and uncoded data. Similarly Figure 5 shows the plot of the PSNR of the recovered Bird image versus JPEG coding at different quality settings. Figure 6 shows the original and watermarked Lena image and Figure 7 shows the recovered Bird image from the watermarked host after it has undergone JPEG compression of different quality.

## 6. CONCLUSION

In this method, a scheme of embedding large amount of data is presented. This approach could be used for both digital watermarking related applications as well as for data hiding purposes. The scale factor controls the relative amount of host and signature data in the embedded image. Experimental results demonstrate that good quality signature recovery and authentication is possible when the images are JPEG compressed and under noise addition. The method performs very well against the JPEG compression attack and the improvement of the

algorithm is quite significant compared to uncoded data as seen from Figure 3. At JPEG compression of 70%, the bit error is almost zero for coded data while its around 13% for uncoded data, and the difference between the two curves increases as the image is severely compressed. Overall, the method performs better in JPEG compression attack than in noise addition. Moreover, from Figure 7 its clear that the recovered watermark image of Bird is quite recognizable even at high compression ratio when the watermarked host is of no commercial value.

## REFERENCES

- [1] M. D. Swanson, M. Kobayashi, and A. Tawfik, "Multimedia Data-Embedding and Watermarking technologies," Proceedings of the IEEE, vol.86, no. 6, June 1998, pp. 1064-1087.
- [2] F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Information Hiding- A survey," Proceedings of the IEEE, Special Issue on Protection of Multimedia Contents, vol. 87, no. 7, July 1999, pp. 1062 -1078.
- [3] Bender W., Gruhl D., and Morimoto N., "Techniques for Data Hiding," Technical Report, MIT Media Lab, 1994.
- [4] R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne, "A Digital Watermark," Proceedings of IEEE International Conference on Image Processing, vol.2, pp. 86-90, Austin, Nov. 1994.
- [5] Cox I., Kilian J., Leighton T., and Shamoon T., "Secure spread-Spectrum watermarking for Multimedia," Technical Report 95-10, NEC Research Institute, 1995.
- [6] M. Barni, F. Bartolini, V. Cappellini, and A. Piva, "A DCT-domain System for Robust Image Watermarking," Signal Processing (Special Issue on Watermarking), vol.66, no. 3, 1998, pp.357-372.
- [7] D. Kundur, and D. Hatzinakos, "A Robust Digital Image watermarking Method using Wavelet-Based Fusion," Proceedings of the IEEE International Conference on Image Processing, Oct. 26-29, 1997, Santa Barbara, CA, vol. 1, pp. 544-547.
- [8] C. I. Podilchuck, and W. Zeng, "Image-Adaptive watermarking using Visual Model," IEEE Journal of selected Areas in Communications, Special Issue on Copyright and Privacy Protection, vol. 16, no. 4, May1998, pp. 525-539.
- [9] Linde Y., Buzo A., and Gray R. M., "AN Algorithm for vector Quantizer Design," IEEE Transactions on Acoustics, Speech, and Signal Processing, vol. 37, 1989, pp. 553-559.
- [10] N. Abdulaziz and K. K. Pang, "Source and Channel Coding Approach for Data Hiding," Proceedings of SPIE, Visual Communications and Image Processing 2000 Conference, 20-23 June 2000, Perth, Australia, pp. 1526-1535.
- [11] N. Abdulaziz and K. K. Pang, "Performance Evaluation of Data Hiding System using Wavelet Transform and Error-Control Coding," In IEEE International Conference on Image Processing 2000 , ICIP 2000, 10-13 September 2000, Vancouver, Canada.

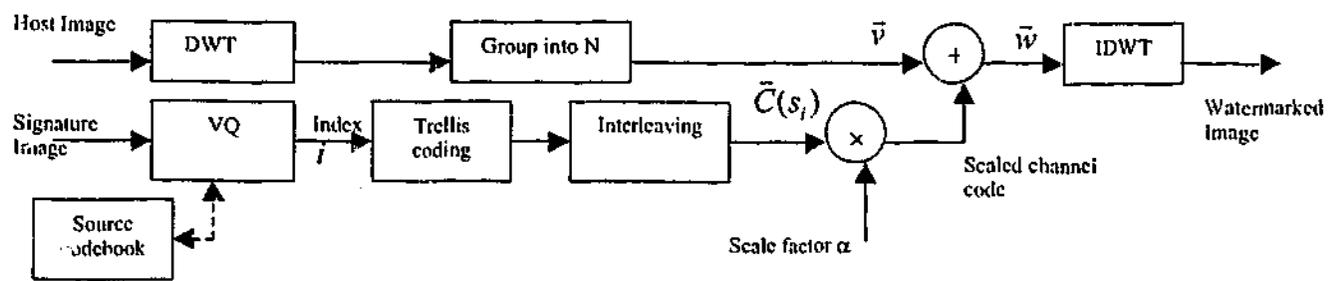


Figure 1. Block diagram of the encoder.

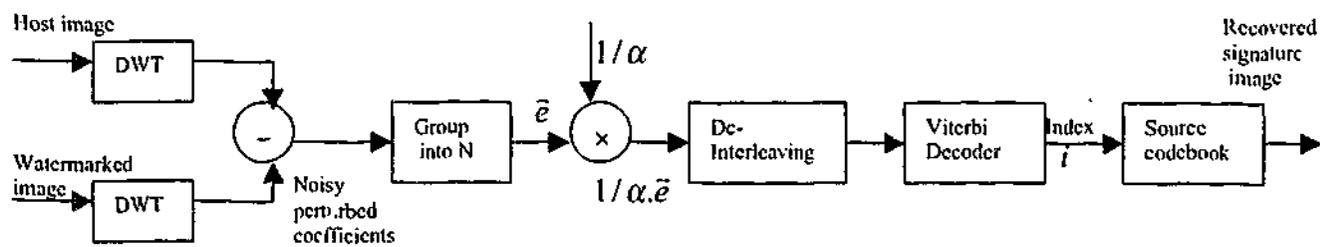


Figure 2. Block diagram of the decoder.

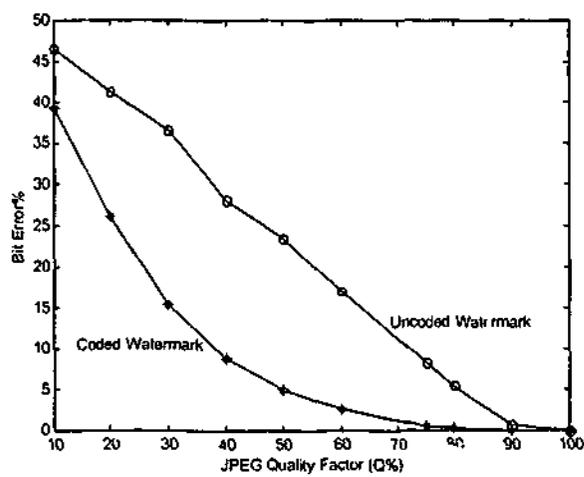


Figure 3. Bit error vs. JPEG compression attack for different quality factors.

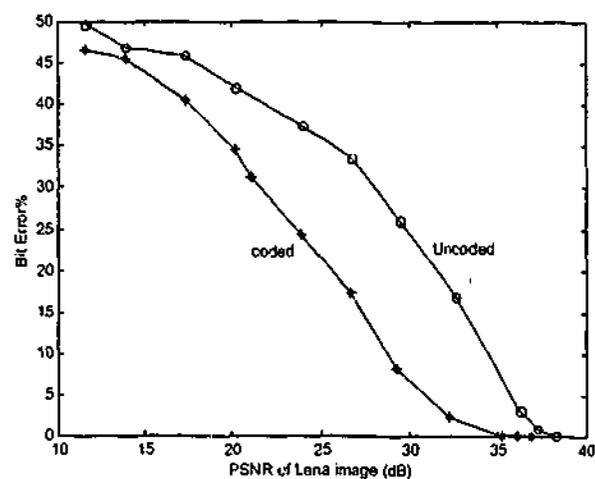


Figure 4. Bit error vs. PSNR of the host image for the noise addition attack.

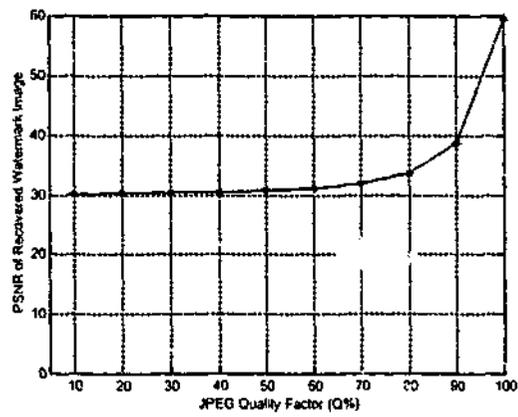


Figure 5. PSNR of recovered watermark image (Bird) vs. JPEG compression attack.

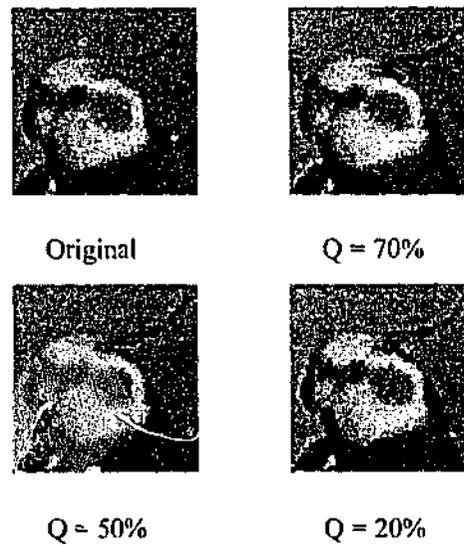


Figure 7. The original and recovered watermark image (Bird) for different JPEG compression.



(a)

(b)

Figure 6. (a) Original Lena image. (b) Watermarked Lena image.