NOTE TO USERS

This reproduction is the best copy available.

UMI®

IMAGE DATA HIDING

by
Zhicheng Ni

A Dissertation
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy in Electrical Engineering

Department of Electrical and Computer Engineering

January 2005

UMI Number: 3186448

Copyright 2004 by
Ni, Zhicheng

All rights reserved.

INFORMATION TO USERS

The quality of this reproduction is dependent upon the quality of the copy submitted. Broken or indistinct print, colored or poor quality illustrations and photographs, print bleed-through, substandard margins, and improper alignment can adversely affect reproduction.

In the unlikely event that the author did not send a complete manuscript and there are missing pages, these will be noted. Also, if unauthorized copyright material had to be removed, a note will indicate the deletion.

UMI®

UMI Microform 3186448
Copyright 2005 by ProQuest Information and Learning Company.
All rights reserved. This microform edition is protected against
unauthorized copying under Title 17, United States Code.

ProQuest Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

Copyright © 2004 by Zhicheng Ni

ALL RIGHTS RESERVED

APPROVAL PAGE

IMAGE DATA HIDING

Zhicheng Ni

11/2/04

Dr. Yun Q. Shi, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Nirwan Ansari — 11/2/04

Dr. Nirwan Ansari, Dissertation Co-advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Constantine Manikopoulos — 10/4/04

Dr. Constantine Manikopoulos, Committee Member — Date
Associate Professor of Electrical and Computer Engineering, NJIT

Frank Shih — 10/4/04

Dr. Frank Y. Shih, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

EK Wong — 10/4/04

Dr. Edward K. Wong, Committee Member — Date
Associate Professor of Computer and Information Science, POLYTECH

ABSTRACT

IMAGE DATA HIDING

by
Zhicheng Ni

Image data hiding represents a class of processes used to embed data into cover images.

Robustness is one of the basic requirements for image data hiding. In the first part of this dissertation, 2D and 3D interleaving techniques associated with error-correction-code (ECC) are proposed to significantly improve the robustness of hidden data against burst errors.

In most cases, the cover image cannot be inverted back to the original image after the hidden data are retrieved. In this dissertation, one novel reversible (lossless) data hiding technique is then introduced. This technique is based on the histogram modification, which can embed a large amount of data while keeping a very high visual quality for all images. The performance is hence better than most existing reversible data hiding algorithms.

However, most of the existing lossless data hiding algorithms are fragile in the sense that the hidden data cannot be extracted correctly after compression or small alteration. In the last part of this dissertation, we then propose a novel robust lossless data hiding technique based on patchwork idea and spatial domain pixel modification. This technique does not generate annoying salt-pepper noise at all, which is unavoidable in the other existing robust lossless data hiding algorithm. This technique has been successfully applied to many commonly used images, thus demonstrating its generality.

BIOGRAPHICAL SKETCH

Author:	Zhicheng Ni
Degree:	Doctor of Philosophy
Date:	October 2004
Date of Birth:	November, 1971
Place of Birth:	Jiangsu, P.R. China

Undergraduate and Graduate Education:

- Doctor of Philosophy in Electrical Engineering,
  New Jersey Institute of Technology, Newark, NJ, USA 2004
- Master of Engineering in Electrical & Electronic Engineering,
  Nanyang Technological University, Singapore, 2000
- Bachelor of Science in Electrical Engineering,
  Southeast University, Nanjing, P.R. China, 1994

Major: Electrical Engineering

Presentations and Publications:

Zhicheng Ni, Yun Q. Shi, Nirwan Ansari,
"Stirmark Attack Resistant Fractal Transform-based Information Hiding," *The Seventh International Conference on Distributed Multimedia Systems*, Tamkang University, Taipei, Taiwan, September 26-28, 2001.

Yun Q. Shi, Zhicheng Ni, Nirwan Ansari and Jiwu Huang,
"2-D and 3-D Successive Packing Interleaving Techniques and their Applications to Image and Video Data Hiding," *IEEE International Symposium on Circuits and Systems,* Bangkok, Thailand, May 2003.

Zhicheng Ni, Yun Q. Shi, Nirwan Ansari and Wei Su,
"Reversible Data Hiding," *IEEE International Symposium on Circuits and Systems,* Bangkok, Thailand, May 2003.

Guorong Xuan, Jidong Chen, Jiang Zhu, Yun Q. Shi, Zhicheng Ni, Wei Su,
"Distortionless Data Hiding Based on Integer Wavelet Transform,"
IEEE International Workshop on Multimedia Signal Processing, St. Thomas, US Virgin islands, December 2002

G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni and W. Su,
"Distortionless data hiding based on integer wavelet transform,"
IEE Electronics Letters, vol. 38, no. 25, pp. 1646-1648, December 2002.

Y. Q. Shi, X. M. Zhang, Z. Ni and N. Ansari,
"Interleaving for combating bursts of errors,"
IEEE Circuits and Systems Magazine, vol. 4, no. 1, pp. 29-42, First Quarter 2004.

Y. Q. Shi, Z. Ni, D. Zou, C. Liang and G. Xuan,
"Lossless data hiding: Fundamentals, algorithms and applications,"
Proceedings of IEEE International Symposium on Circuits and Systems, vol. II, pp. 33-36, Vancouver, Canada, May 2004.

Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin,
"Robust lossless image data hiding," *2004 IEEE International Conference on Multimedia and Expo,* Taipei, Taiwan, June 2004.

To my parents, Ni, Yiqing and Wang, Haizhen
and my brother Ni, Zhiqiang

ACKNOWLEDGMENT

First of all, I would like to express my deepest appreciation to my academic advisor, Professor Yun Q. Shi, who not only provided me valuable and countless guidance, resources, insight, and intuition, but also constantly gave me support, encouragement, and reassurance throughout the years.

I also would like to express my deep thanks to my co-advisor, Professor Nirwan Ansari for helping me throughout the years. Special thanks are given to Dr. Constantine Manikopoulos, Dr. Frank Y. Shih, and Dr. Edward K. Wong for actively participating in my committee.

I would like to express my special thanks to Dr. Ximin Zhang for his insightful suggestions on my research work and his kind help for my life.

I would also like to thank Dr. Hong Zhao for her invaluable discussion and help. I would like to thank my colleagues in NJ Wireless Center and NJIT, especially, Jiongkuan Hou, Dequan Liu, Dongdong Fu, Jingxuan Liu, Zhiqiang Gao, Zaihe Yu, Dekun Zou, and Yuanqiu Luo.

Finally, I would like to acknowledge my parents who have always been there for their immeasurable devotion and encouragement.

TABLE OF CONTENTS

Chapter **Page**

1 INTRODUCTION... 1

1.1 Motivation 1

1.2 Applications 2

1.3 Contribution of this Dissertation 2

1.4 Outline of this Dissertation 5

2 EXISTING IMAGE DATA HIDING SCHEMES 6

2.1 Spatial Domain Algorithm 6

2.2 DCT Domain Algorithm 7

2.3 Wavelet Domain Algorithm 11

3 2-D AND 3-D SUCCESSIVE PACKING INTERLEAVING TECHNIQUES AND THEIR APPLICATIONS TO DATA HIDING 12

3.1 Background of Successive Packing Interleaving Technique 13

3.2 The Application of 2-D SP Interleaving in Still Image Data Hiding 16

3.3 The Application of 3-D SP Interleaving to Video Data Hiding 19

4 LOSSLESS IMAGE DATA HIDING 23

4.1 Reversible Data Hiding based on the Histogram Modification 24

Algorithm 25

4.2 Some Experimental Results 30

5 ROBUST LOSSLESS IMAGE DATA HIDING 34

5.1 Introduction 34

5.2 A Novel Robust Lossless Image Data Hiding Algorithm 40

TABLE OF CONTENTS
(Continued)

Chapter **Page**

5.2.1 Main Idea 40

5.2.2 Bit Embedding Strategy 42

5.2.3 Error Correction Code 49

5.2.4 Chaotic Mixing 49

5.2.5 Data Embedding Diagram 50

5.2.6 Data Extraction 51

5.3 Experimental Results 52

5.4 Authentication Application 56

5.4.1 Content Signing 56

5.4.2 Signature Embedding 57

5.4.3 Authentication 57

5.5 Conclusion 58

6 CONCLUSIONS AND FUTURE WORK 59

REFERENCES 61

LIST OF FIGURES

Figure **Page**

2.1 Spatial domain algorithm 7

2.2 Data embedding diagram 9

2.3 Data extraction diagram 10

2.4 Data embedding algorithm 11

3.1 Results for block burst error attack 17

3.2 Results for "stirmark" cropping attack 18

3.3 Results for "Stirmark" jitter attack 19

3.4 Results for frame loss attack 20

3.5 Results for 3-D error burst attack 21

4.1 Histogram of Lena image 25

4.2 Lena image 26

4.3 Histogram of marked Lena image 26

4.4 Histogram of one medical image. (Note that there are two huge peaks, 164,683 and 20,739, in the histogram, located at gray levels 0 and 255, respectively. They have been scaled down on purpose in order to show more details for the vast majority of other gray levels.) 27

4.5 Data embedding algorithm 29

4.6 Data extracting algorithm 29

4.7 Watermark (a binary image of 171 × 93) 30

4.8 Airplane image 31

4.9 Histogram 32

4.10 Other eight marked images 32

LIST OF FIGURES
(Continued)

Figure **Page**

5.1 Histogram mapping onto a circle 35

5.2 Data embedding diagram 35

5.3 A medical image, Mpic1 38

5.4 A JPEG2000 test image, Woman 38

5.5 Difference pair pattern 40

5.6 The distribution of difference value α 42

5.7 Block histogram for category 1 43

5.8 Embedding a bit ‘1’ 44

5.9 Embedding a bit ‘1’ 44

5.10 Block histogram for category 2 45

5.11 Embedding a bit ‘1’ 45

5.12 Embedding a bit ‘1’ 46

5.13 Increase threshold or block size to embed bit ‘0’ 46

5.14 Block histogram for category 3 47

5.15 Block histogram for category 4 47

5.16 Embedding a bit ‘0’ 48

5.17 The whole block is intact 48

5.18 Chaotic mixing of Baboon image by A^n, a) original image, b) n=1, c) n=5, d) n=10, where n is the numbers the chaotic mixing has applied 50

5.19 Block diagram of data embedding 51

5.20 Extracting bit ‘1’ 52

LIST OF FIGURES
(Continued)

Figure **Page**

5.21 Block diagram of data extraction ... 52

5.22 A medical image, Mpic1 ... 54

5.23 A CoralDraw image ... 54

5.24 A JPEG2000 test image, Woman ... 55

5.25 Block diagram of content signing ... 57

5.26 Block diagram of authentication ... 57

LIST OF TABLES

Table **Page**

4.1 Experimental Results 33

4.2 Comparison between Three Reversible Marking Methods [14,15,16] and our Proposed Method Objective 33

5.1 Test Results for Eight Medical Images 39

5.2 Test Results for Eight JPEG2000 Test Images 39

5.3 Test Results for Commonly used 512×512×8 Grayscale Images 55

5.4 Test Results for 1000 Images in CorelDraw Database 55

5.5 Test Results for Eight Medical Images 56

5.6 Test Results for Eight JPEG2000 Color Test Images 59

CHAPTER 1

INTRODUCTION

In this chapter, the background of this dissertation is first introduced, which includes the fundamentals of data hiding, the applications of data hiding. Then some contributions of this dissertation are presented, which include three proposed image data hiding techniques. Finally, future research is discussed and dissertation is concluded.

1.1 Motivation

The proliferation of digital multimedia (audio, image and video) and the emergence of digital networks are creating a pressing need for copyright enforcement schemes. Owners of multimedia need to be able to protect them against illegal duplication, distribution and other unauthorized usage. Several technologies have been proposed for intellectual property right (IPR) protection [1]. One is encryption. However, conventional encryption and copy protection mechanisms do not fully solve the issue in some applications. Recently, data hiding (watermarking) is proposed as a hopeful method for authentication, fingerprint, security, data mining, and copyright protection, etc. Here we only focus on invisible data hiding in this dissertation. For the information about visible data hiding, refer to [2].

Image data hiding represents a class of processes used to embed data into cover images. The hidden data are usually a string of binary bits (e.g., digital signature), a logo image, identification (ID) number or any information that is useful. Unlike data labeling which usually adds some data to the header of a file/bitstream, the data are embedded into the content, therefore goes wherever the contents goes, and requires no additional storage. The hidden data are invisible to human being but can be recovered by a computer program.

1.2 Applications

The hidden data may have different purposes according to the applications. Some typical applications are presented in the following.

- **Authentication** The hidden data can be used to decide whether and how much the data has been changed from its original form. In some special situation, we may want the hidden data to survive the ordinary signal processing, but will be more sensitive to other changes such as editing of the objects [3].
- **Fingerprinting** The hidden data can be used to identify the buyer of the content. If the content is illegal copied and distributed, the hidden data (fingerprint) can be used to find the illegal distributor.
- **Resolving rightful ownership** The hidden data can be used to identify the rightful owner of the content. Digital watermark can be used to display the copyright information of the content and may be used to find owner's contact file through certain database [4].
- **Covert communications** In this application the message data can be hidden inside cover media and hence transmitted to a receiver in a secret way.

1.3 Contributions of this Dissertation Research

Besides invisibility, robustness is another important requirement for some applications. Both watermark structure and embedding strategy affect the robustness.

At the early stage, only normal signal processing procedures and noise corruption were considered as attack to hidden data. Afterwards malicious attacks such as collusion and ambiguity attack were included. Recently, geometric attack is the focus of attention. In particular, StirMark has been developed as a benchmark for robustness test purpose

[5]. As a result, great efforts [6] have been devoted to enhancing the robustness of marking algorithms. In most cases, error-correction-code (ECC) is often adopted to improve the robustness, say, in [7]. However, ECC is only suitable to correct random errors. When bursts of errors happen, once the number of errors is so large that the Hamming distance between distinct code words exceeds the maximum allowed value for the used code, ECC will fail to correct the error. When cropping and/or jitter attack, two testing functions in the well-known benchmark test software "StirMark", take place, bursts of errors do occur in still images. Frame loss and 3D error cluster are typical bursts of errors occurred in video sequences. Transmission error is also a source of bursts of errors. Using ECC alone to correct these bursts of errors will not be efficient. A new interleaving technique called successive packing interleaving technique was proposed in [8]. It shows that after de-interleaving, the error burst will be spread evenly in the whole array. Hence it inspires us to apply the interleaving technique to enhance the robustness of data hiding. That is, we should correct bursts of errors by incorporating interleaving in error-correction-codes. In this dissertation, we first introduces the successive packing interleaving technique as well as ECC code to the data hiding algorithm [9] and presents a series of experiments, which incur 2D error burst, cropping, jitter attack to still image data hiding, and 3D error burst, frame loss to video sequence data hiding. The experimental results show that the robustness of hidden data against bursts of errors is significantly improved by integrating 2D and 3D interleaving with ECC.

In some applications, people do care about the cover media. For this type of data embedding, in addition to perceptual transparency, for some applications such as medical diagnosis and law enforcement, it is desired to invert the marked image back to the original cover image after the hidden data have been retrieved. The marking techniques satisfying this requirement are referred to as lossless, *distortion-free* or *invertible* data hiding techniques. From this point of view, it is observed that most of the current digital watermarking algorithms are not lossless. A novel reversible data hiding algorithm was proposed [10]. This algorithm utilizes the zero or the minimum point of the histogram and slightly modifies the pixel values to embed data. It can embed more data (5k-80k bits into a $512\times512\times 8$ grayscale image) than most of the existing reversible data hiding algorithms. It is shown analytically and experimentally that the PSNR of the marked image generated by this method versus the original image is always above 48 dB, which is much higher than all existing reversible data hiding algorithms. The algorithm has been successfully applied to a wide range of images including medical images.

According to the survey [11], recently many lossless data hiding techniques have been proposed, such as in [10,12-19]. However, most of them are fragile in the sense that the hidden data cannot be recovered after compression or other incidental alteration is applied to the marked image. Thus far, De Vleeschouwer et al.'s method [19] is the only existing robust (or semi-fragile) lossless data hiding technique against high-quality JPEG compression. This technique can be applied to semi-fragile image authentication. In the dissertation, it is first pointed out that this technique has suffered from the annoying salt-pepper noise caused by using modulo 256 addition. Then a novel robust lossless data hiding technique is proposed, which does not generate salt-pepper noise at all. This technique is based on the patchwork idea and spatial domain pixel modification. This technique has been successfully applied to many commonly used images (including medical images, more than 1000 images in CorelDRAW database and JPEG2000 test images), thus demonstrating its generality. The experimental results show that the visual quality of stego-images, the data embedding capacity and the robustness of the proposed lossless data hiding scheme against compression are acceptable for some applications, including authentication. It is expected that this new robust lossless data hiding algorithm can find wide applications in the fields such as medical and law enforcement fields. In addition, it has been successfully applied to authenticate losslessly compressed JPEG2000 images, possibly followed by transcoding.

1.4 Outline of this Dissertation Proposal

Chapter 2 briefly reviews some existing data hiding algorithms and classifies them into a few categories. Then introduction about each category is presented.

Chapter 3 proposes a data hiding technique, which introduces interleaving technique and ECC code to combat the burst of errors.

Chapter 4 provides a novel reversible data hiding technique based on the histogram modification. The whole algorithm and the experimental results are presented.

Chapter 5 provides a novel robust lossless data hiding technique based on the patchwork idea. The whole algorithm and the experimental results are presented.

Chapter 6 provides the conclusions and future work of this dissertation.

CHAPTER 2

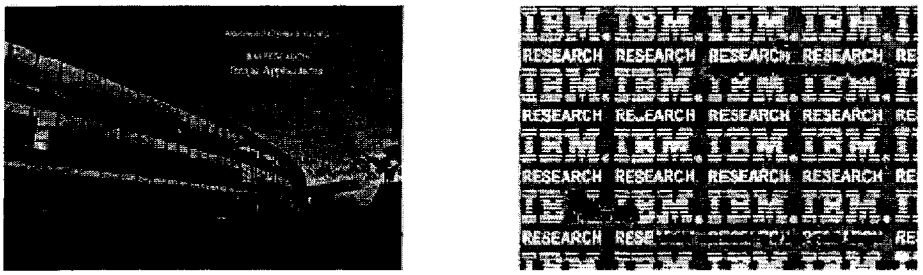
EXISTING IMAGE DATA HIDING TECHNIQUES

In this chapter, first the art of image data hiding development is reviewed. At the early stage, the data hiding techniques are usually in the spatial domain. The data are usually embedded into the perceptually less significant parts of the data such as the least significant bits or the high frequency components [20][21]. Then, the importance of the robustness of the data hiding is gradually recognized. Some data hiding techniques emphasized on the robustness are proposed [22][23][24]. With the development of image data hiding techniques, the data hiding methodology can be classified into different categories, such as in spatial domain, DCT transform, wavelet transform, fractal transform, etc. Recently, many new applications of image data hiding are explored, such as in authentication, fingerprinting, temper detection, stegoanalysis, medical data system, data mining, lossless data hiding, etc. Some typical image data hiding techniques are presented here.

2.1 Spatial Domain Algorithm

There are many image data hiding algorithms based on spatial domain, in which Yeung's technique [25] is a typical one. Her method is actually a variation of the least significant bit (LSB) method. It relies on the slightly modification of the pixel value. This method is intended for use in image verification applications to determine whether the content of an image has been altered, due perhaps, to the act of a malicious party (refer to Figure 2.1). The data hiding process does not introduce visual artifacts and preserves image quality.

The PSNR of all tested images is above 50 dB. As other LSB techniques, the drawback of this method is that it is fragile, without any robustness to the attacks.



(a) Altered image (b) Extracted watermark with detected alteration

Figure 2.1 Spatial domain algorithm.

2.2 DCT Domain Algorithm

Recent years, data hiding algorithms based on DCT transform has become popular since the JPEG compression standard is also based on DCT transform. Among many DCT based image data hiding algorithms, the method of Cox [26] is a typical one, which is based on the DCT transform and spread spectrum. His method got pretty good results in terms of resisting ordinary signal processing attacks. The drawback of his technique is that it only embeds one bit (yes or no) information, without capability to embed useful information. A brief introduction about his algorithm is given here.

First, through discussion, he illustrates that the watermark should *not* be placed in perceptually insignificant regions of the image (or its spectrum) since many common signal and geometric processes affect these components. Then he treats the frequency domain of the image at hand as a communication channel, and correspondingly, the watermark is viewed as a signal that is transmitted through it. Cox then conceived his approach by analogy to *spread spectrum communication.* In spread spectrum communications, one transmits a narrow band signal over a much larger bandwidth such that the signal energy present in any signal frequency is undetectable. Similarly, the watermark is spread over very many frequency bins so that the energy in any one bin is very small and certainly undetectable. Spreading the watermark throughout the spectrum of an image ensures a large measure of security against unintentional or intentional attacks. In his data embedding algorithm, he first uses the DCT transform on the original image to get the coefficients, then chooses the largest 1000 coefficients (except the DC component) to embed the watermark. Finally, he uses the inverse DCT transform to get the marked image. The followings are the data embedding and extraction diagram.

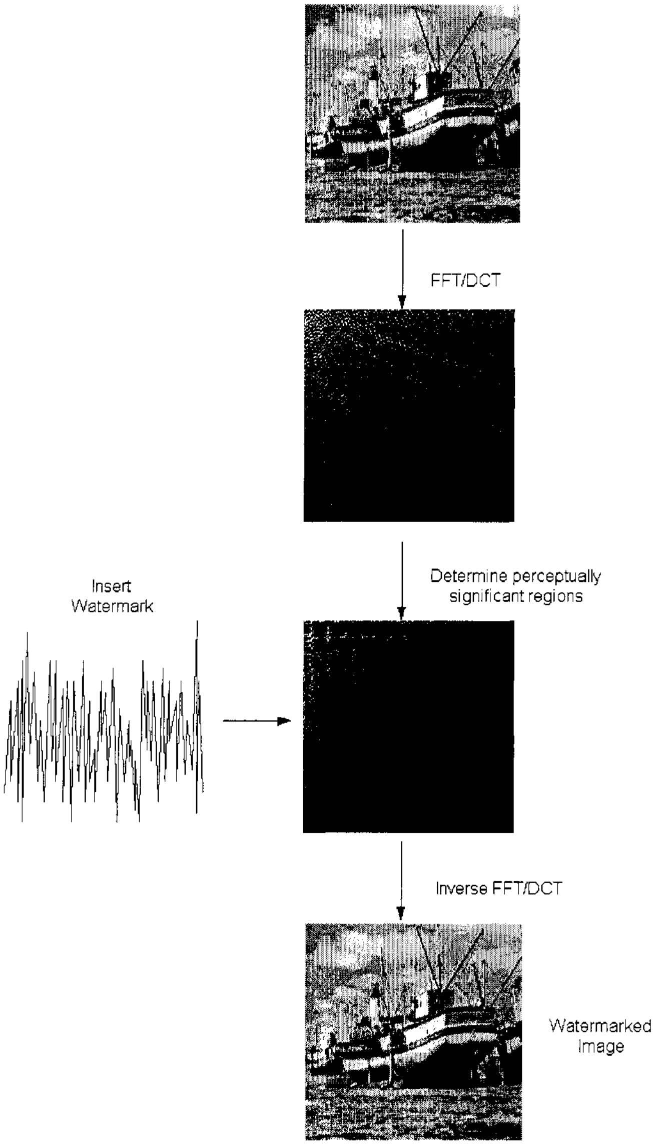


Figure 2.2 Data embedding diagram.

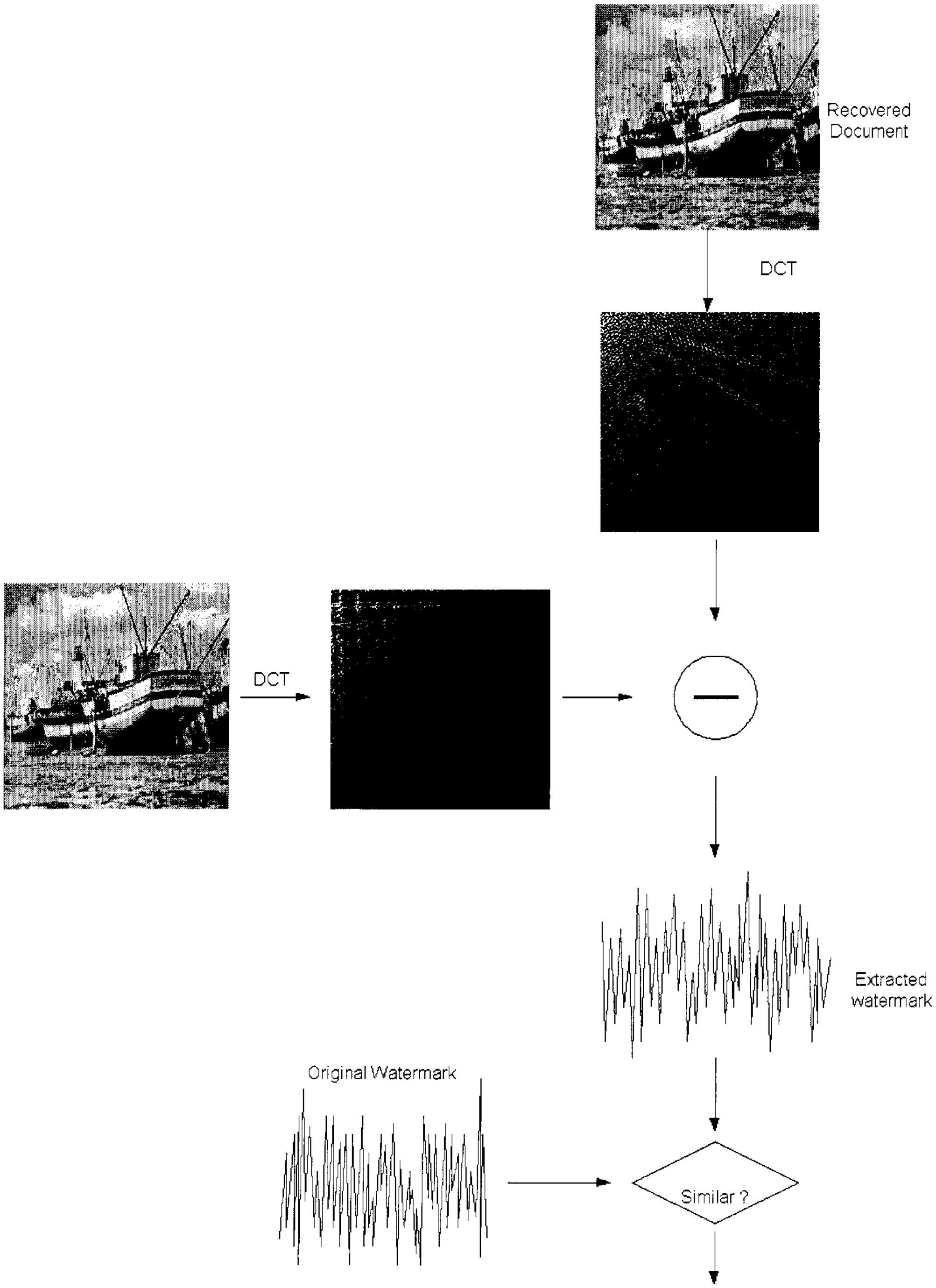


Figure 2.3 Data extraction diagram.

2.3 Wavelet Domain Algorithm

Besides DCT transform, some persons also explore the data hiding method based on the wavelet transform [27][28][29] since JPEG2000 standard is also based on the wavelet transform. Among these techniques, Liu's [29] method is a typical one. She first applies discrete wavelet transform on the original image and gets the coefficients. Then, The wavelet coefficients are changed according to the embedded data. Finally, inverse wavelet transform is applied to get the marked image.

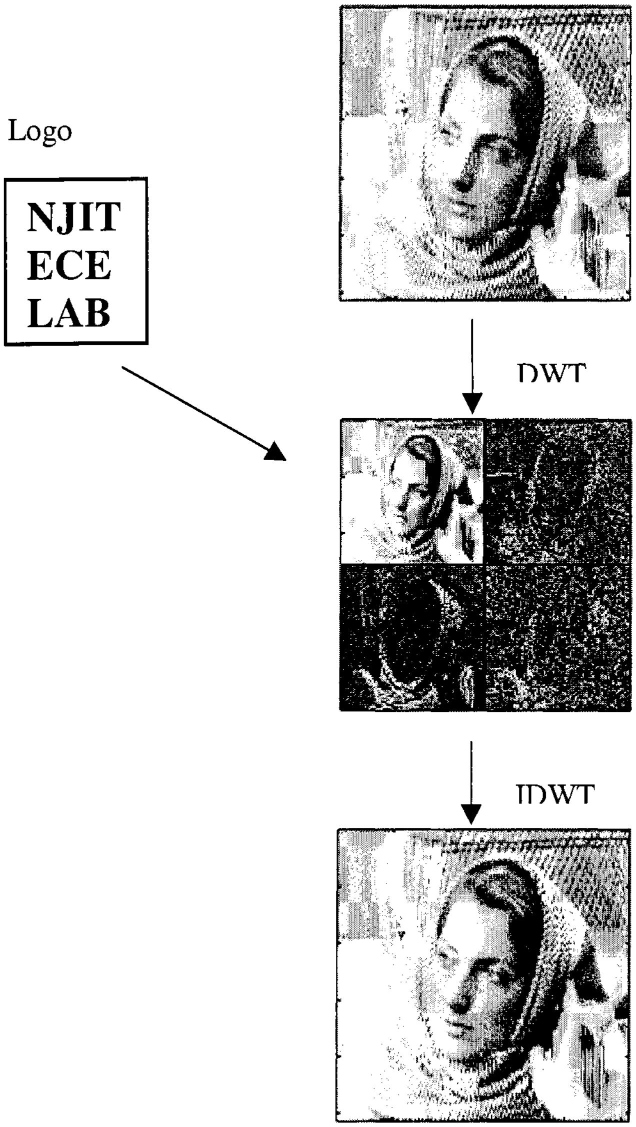


Figure 2.4 Data embedding algorithm.

CHAPTER 3

2-D AND 3-D SUCCESSIVE PACKING INTERLEAVING TECHNIQUES AND THEIR APPLICATIONS TO DATA HIDING

Recently, error-correction-code (ECC) is proposed to improve the robustness of data hiding [7]. However, ECC is only suitable to correct random errors. When bursts of errors happen, once the number of errors is so large that the Hamming distance between distinct code words exceeds the maximum allowed value for the used code, ECC will fail to correct the error. When cropping and/or jitter attack, two testing functions in the well-known benchmark test software "stirmark" [5], take place, bursts of errors do occur in still images. Frame loss and 3-D error cluster are typical bursts of errors occurred in video sequences. Transmission error is also a source of bursts of errors. Using ECC alone to correct these bursts of errors will not be efficient. Surprisingly, it is noted that combating bursts of errors using interleaving technique, which is a common tool used in communication systems, in data hiding has been neither recognized nor addressed before.

A new interleaving technique called successive packing interleaving technique was proposed in [8]. It shows that after de-interleaving, the error burst will be spread evenly in the whole array. Hence, it is inspired to apply the interleaving technique to enhance the robustness of data hiding. That is, bursts of errors should be corrected by incorporating interleaving in error-correction-codes [9]. This dissertation presents a series of experiments, which incur 2D error burst, cropping, jitter attack to still image data hiding, and 3D error burst, frame loss to video sequence data hiding. The experimental results show that the robustness of hidden data against bursts of errors is significantly improved by integrating 2D and 3D interleaving with ECC.

3.1 Background of Successive Packing Interleaving Technique

The philosophy of using interleaving technique to combat bursts of errors is to spread bursts of errors so that the error burst can be converted to random-like noise so that some simple random-error-correction codes can be used to correct the errors [30]. Recently, two-dimensional (2-D) and three-dimensional (3-D) successive packing interleaving techniques have been proposed [8] to correct 2-D and 3-D bursts of errors.

3.1.1 2-D Successive Packing Interleaving Technique

For a detailed introduction and proof about 2-D SP interleaving technique, readers are referred to [8]. Here, we only present the algorithm and main results.

Procedure 1 The 2-D interleaving using the successive packing proceeds as follows. Consider a 2-D array of $2^n \times 2^n$ for 2-D interleaving.

When n = 0, an array of 1×1 is considered for interleaving, the interleaved array is the original array itself. That is,

$$S_1 = [s_0], \tag{3.1}$$

where s_0 represents the element in the array, and S_1 the array. We note that the subscript in the notation S_1 represents the total number of elements in the interleaved array. Hence, when n = 1, i.e., for a 2×2 array, the interleaved array is denoted by S_4; when n = 2, the interleaved array is S_{16}. In general, for a given n, we have the interleaved array denoted by $S_{2^{2n}}$.

The procedure is carried out successively. Given an interleaved array S_i, the interleaved array of S_{4i} can be generated according to

$$S_{4i} = \begin{bmatrix} 4\times S_i + 0 & 4\times S_i + 2 \\ 4\times S_i + 3 & 4\times S_i + 1 \end{bmatrix} \tag{3.2}$$

where the notation of $4\times S_i + k$ with k=0, 1, 2, 3 represents a 2D array that is generated from S_i. This indicates that $4\times S_i + k$ has the same dimensionality as S_i. Furthermore, each element in $4\times S_i + k$ is indexed in such a way that its subscript equals to four times that of the corresponding elements in S_i plus k. It appears that S_{4i} is derived from S_i by *packing* S_i four times. Thus, such interleaving are referred as the successive packing.

According to the above rule, it has

$$S_4 = \begin{bmatrix} 4\times S_1 + 0 & 4\times S_1 + 2 \\ 4\times S_1 + 3 & 4\times S_1 + 1 \end{bmatrix} = \begin{bmatrix} s_0 & s_2 \\ s_3 & s_1 \end{bmatrix}. \tag{3.3}$$

Similarly,

$$S_{16} = \begin{bmatrix} 4\times S_4 + 0 & 4\times S_4 + 2 \\ 4\times S_4 + 3 & 4\times S_4 + 1 \end{bmatrix} = \begin{bmatrix} s_0 & s_8 & s_2 & s_{10} \\ s_{12} & s_4 & s_{14} & s_6 \\ s_3 & s_{11} & s_1 & s_9 \\ s_{15} & s_7 & s_{13} & s_5 \end{bmatrix}. \tag{3.4}$$

Now a theorem is presented in [8] without proof, which guarantees to spread the error burst in the whole 2-D array.

Theorem 1 Consider a 2-D array of $2^n \times 2^n$. Partition it into 2^m blocks for an integer m satisfying $1 \le m \le 2n - 1$. When m is even and $m = 2k$, any burst of $2^k \times 2^k$ in the interleaved array, A, obtained by using the successive packing is spread in the de-interleaved array such that each element of the burst falls into a distinct block of size 2^{2n-2k}. When m is odd and $m = 2k+1$, any burst of $2^k \times 2^{k+1}$ or $2^{k+1} \times 2^k$ in A is spread in the de-interleaved array such that each element of the burst falls into a distinct block of size $2^{2n-2k-1}$.

This theorem ensures that after de-interleaving, the burst error will spread evenly in the whole array. This technique guarantees that the error burst can be corrected with some random-error-correction code, provided the code is available. It also points out that this technique is optimal for a set of bursts with different sizes.

3.1.2 3-D Successive Packing Interleaving Technique

The procedure to implement 3-D successive packing interleaving is as follows.

Procedure 2 Consider a 3-D array of $2^n \times 2^n \times 2^n$ for 3-D interleaving.

When n = 0, i.e., an array of 1×1 is considered for interleaving, the interleaved array is the original array itself. That is,

$$S_1 = [s_0], \tag{3.5}$$

where s_0 represents the element in the array, and S_1 the array. We note that the subscript in the notation S_1 represents the total number of elements in the interleaved array. Hence, when n = 1, i.e., for a $2\times2\times2$ array, the interleaved array is denoted by S_8; when n = 2, the interleaved array is S_{64}. In general, for a given n, we have the interleaved array denoted by $S_{2^{3n}}$.

The procedure is carried out successively. Since both S_i and S_{8i} are 3-dimensional, we express the successive procedure in two (upper and lower) levels. The upper level of S_{8i} is

$$S_{8i} = \begin{bmatrix} 8\times S_i+5 & 8\times S_i+3 \\ 8\times S_i+7 & 8\times S_i+1 \end{bmatrix}. \tag{3.6}$$

The lower level of S_{8i} is

$$S_{8i} = \begin{bmatrix} 8\times S_i+0 & 8\times S_i+6 \\ 8\times S_i+2 & 8\times S_i+4 \end{bmatrix}, \tag{3.7}$$

where the notation of $8\times S_i + k$ with k=0, 1, 2, 3, 4, 5, 6, 7 represents a 3-D array that is generated from S_i. This indicates that $8\times S_i + k$ has the same dimensionality as S_i. Furthermore, each element in $8\times S_i + k$ is indexed in such a way that its subscript equals to eight times that in S_i plus k.

3.2 The Application of 2-D SP Interleaving in Still Image Data Hiding

In the still image data hiding, there are many attacks that can destroy the embedded data. Hence, the 2-D SP interleaving technique is applied to improve the robustness of the hidden data. The results have demonstrated significant improvement of robustness of hidden data.

Now, the simulations are presented. The "lena" image ($256\times 256\times 8$) is used for illustrative purpose. First it is splitted into non-overlapping blocks, each of 8×8 pixels. Then, DCT transform is applied to each block. The largest three AC coefficients are used to embed the information bit. If information bit is "1", the coefficient is added by Δ, where $\Delta = 6$ is empirically used in the simulations. If the information bit is "0", the coefficient is subtracted by Δ. 6 bits are used to represent one symbol. In 2D error burst and cropping attacks, the BCH(31,6) code is used. In jitter attack, the BCH(63,7) code is used. In the former case, 99 symbols are embedded. In the latter case, 48 symbols are embedded. The image is scanned three times. Each scan embeds 1024 bits in the AC coefficient in the same position of each block. These 1024 bits are first interleaved using the above 2D SP interleaving technique, and then embedded into the AC coefficients. In the data extraction, parts damaged by the error burst are replaced with "0"s.

Without 2D interleaving, bits are simply embed block by block, say, from left to right, from top to bottom. In each block, three bits are embedded. Figures 3.1-3.3 show the experimental results, indicating that 2D interleaving can greatly improve the robustness of data hiding.

In Figure 3.1, the x-axis represents the size of the error burst. For example, block size 2 means that the error burst is a square area of 2×2 blocks (each block is 8×8 pixels). That is, the square error burst has a size of 16×16 pixels. As shown in the figure, without interleaving, the character error rate (CER) emerges when the error size is larger than 4. With interleaving, the character error rate is still zero even when the error size is 16. This indicates that even when a quadrant of the marked lena image has been in error, the CER is still zero, implying significant improvement of robustness. Note that when the error size is larger than 22, the CER with interleaving will be larger than that without interleaving. In this case, almost half of the image has been damaged. The CER for both algorithms with and without interleaving has been almost 50%, and is not usable any longer.

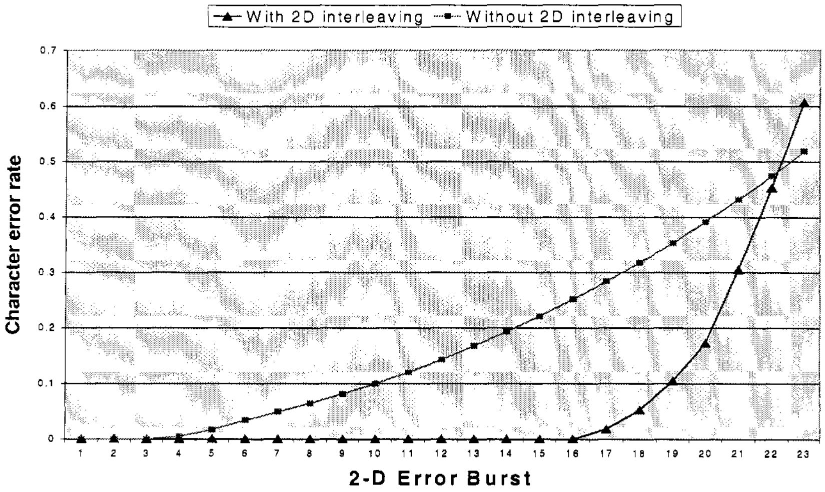


Figure 3.1 Results for block burst error attack.

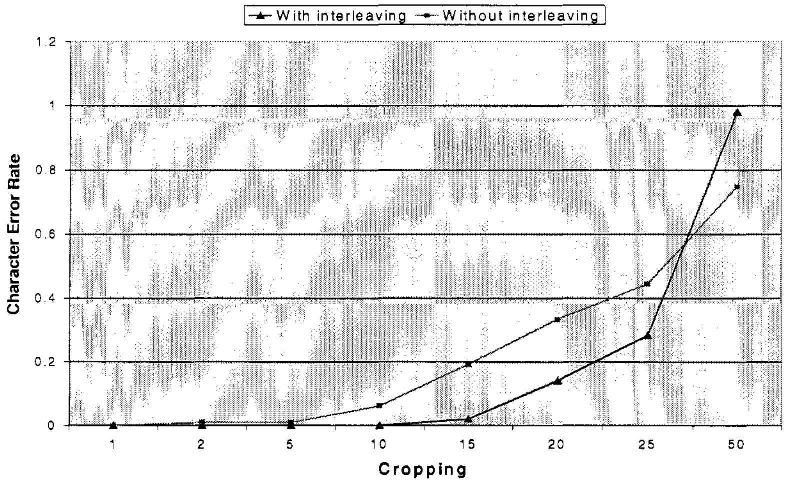


Figure 3.2 Results for "stirmark" cropping attack.

In Figure 3.2, the x-axis represents the size of cropping. For example, a cropping of 10 means that five percentage is cropped from the top, bottom, left and right while keeping the central part intact. From this figure, with interleaving even when the cropping size is 10, the CER is still zero while without interleaving, the CER emerges when the cropping size is only two. The robustness against cropping attack has been improved with interleaving. Note that when the cropping size is close to 50, the CER with interleaving will be higher than that without interleaving, in which case almost all the image has been cropped.

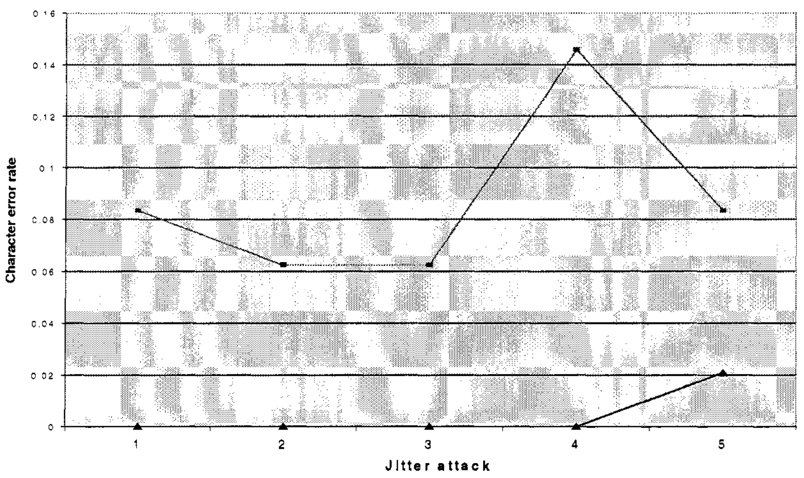


Figure 3.3 Results for "Stirmark" jitter attack.

In Figure 3.3, the x-axis represents the number of columns or rows randomly removed. From this figure, the CER with interleaving is always below that without interleaving.

3.3 The Application of 3-D SP Interleaving to Video Data Hiding

The 3D successive packing interleaving technique is also applied for video data hiding. A video sequence is actually a set of successive frames. Each frame is a 2D image.

There are two types of burst error that can occur to the data embedded in a video sequence. The first type of error is the frame loss. Frame loss often occurs in the video transmission, especially when the video is transmitted through a busy network or noisy channel. The second type of error is the 3-D error burst. Since there is a large correlation among the neighboring frames, a 2-D error burst sometimes leads to the errors approximately in the same location of the proceeding frames. Hence, a 3-D error burst is produced.

For these two types of error, again, simulation is constructed similar to that for 2D data hiding. The simulation results show that video data hiding using 3D SP interleaving significantly improves the robustness as compared with that without using the 3D interleaving.

The simulation set-up is first presented. The tested video has 32 frames. Each frame is a 256×256 gray image. We split each frame into non-overlapping blocks, each of 8×8 pixels. Therefore, $32 \times 32 \times 32$ blocks are within the entire sequence. DCT transform is applied to each block and the information bits are embedded into the three largest AC coefficients. Again, 6 bits are used to represent one symbol and BCH(31,6) code is used. In each scan, it embeds $32 \times 32 \times 32$=32768 bits (equivalent to 1057 symbols) in the AC coefficient having the same position in each block. These 32768 bits are first interleaved using the above 3D SP interleaving technique before being embedded into the AC coefficients. In the data extraction, for destroyed parts, we fill up with "0"s.

Without 3D interleaving, we simply embed bits block by block, say, from left to right, from top to bottom, and from front to rear. In each block, we embed three bits. Figures 3.4-3.5 show the simulation results, demonstrating that 3D interleaving can greatly improve the robustness of data hiding.

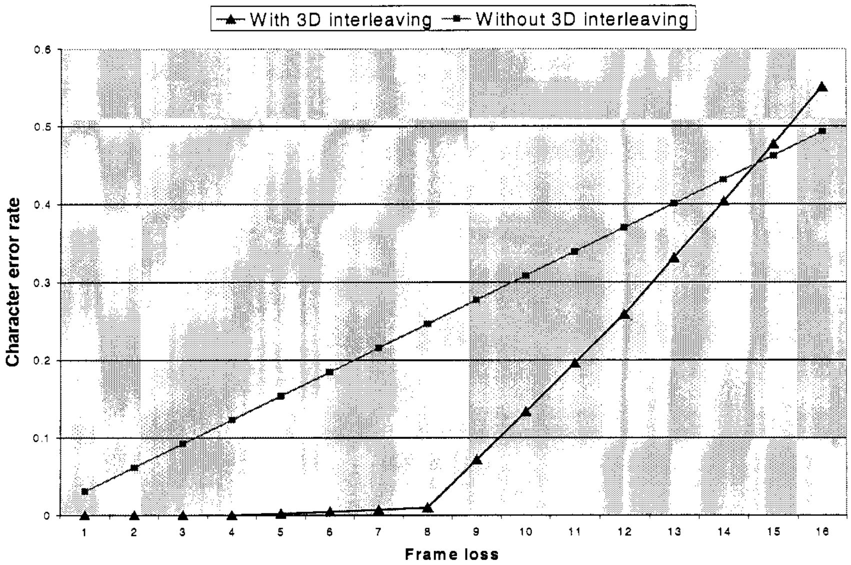


Figure 3.4 Results for frame loss attack.

In Figure 3.4, the x-axis denotes the number of lost frames. From this figure, when eight frames are lost, the character error rate (CER) with interleaving is almost zero while that without interleaving is about 25 percentages. Note that when 15 frames are lost, the error rate with interleaving will be higher than that without interleaving, in which case, almost half of the 32 frames are lost. The CER for both algorithms with and without interleaving has been almost 50%, implying that the hidden data have been severely damaged.

In Figure 3.5, the x-axis is the size of the 3D error burst. For example, block size 2 means that the error burst is a cubic area of $2 \times 2 \times 2$ blocks (each block is of 8×8 pixels). That is, the cubic error burst has a size of $16 \times 16 \times 2$ pixels. With interleaving, the CER is still zero even when the error burst size is 16 (one eighth of the blocks are lost $(16 \times 16 \times 16)/(32 \times 32 \times 32)=1/8$), while without interleaving the CER is more than 12%.

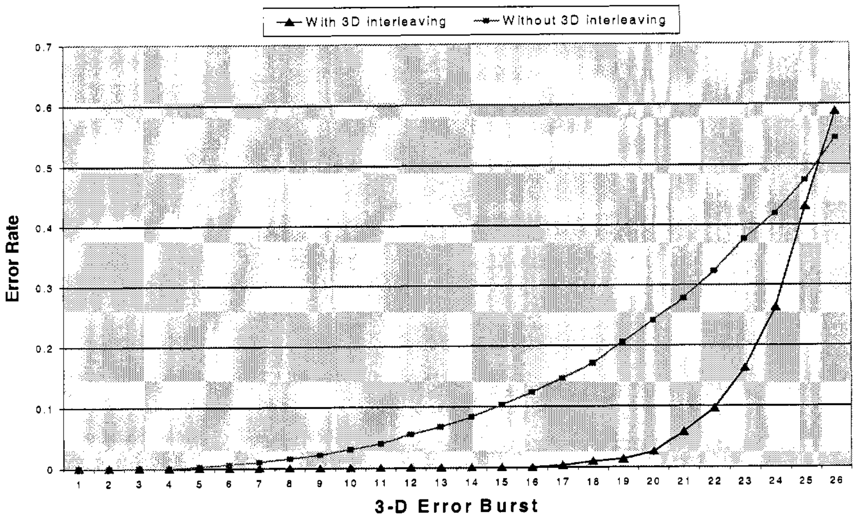


Figure 3.5 Results for 3-D error burst attack.

CHAPTER 4

LOSSLESS IMAGE DATA HIDING

In data hiding, pieces of information represented by the data are hidden in cover media. In some applications, people do care about the cover media. That is, the hidden data and the cover media may be closely related. For this type of data embedding, in addition to perceptual transparency it is desired to invert the marked media back to the original cover media without any distortion after the hidden data have been retrieved in some applications such as medical diagnosis and law enforcement. The marking techniques satisfying this requirement are referred to as *invertible* or *distortionless* data hiding techniques. From this point of view, it is observed that most of the current digital watermarking algorithms are not distortionaless. For instance, with the most popularly utilized spread-spectrum watermarking techniques, either in DCT domain [26] or block 8x8 DCT domain [31,32,33], round-off error and/or truncation error may take place during data embedding. As a result, there is no way to invert the marked media back to the original without distortion. With the popularly used least significant bit-plane (LSB) embedding method, the bits in the LSB are replaced according to the data to be embedded and the bit-replacement is not *memorized*. Consequently, the LSB method is not invertible. With another group of frequently used watermarking techniques, called quantization index modulation (QIM) [34], quantization error makes distotionless data hiding impossible.

Recently, some distortionless marking techniques have been reported in the literature. The first method [12] is carried out in the image spatial domain by using addition modulo 256. Another spatial domain technique was reported in [13], which losslessly compresses bit-planes. The third work of distortionless marking technique [14] is also carried out in the spatial domain, which is a modification of the patchwork algorithm and embeds one bit watermark. These techniques aim at authentication, instead of data embedding. As a result, the amount of hidden data is quite limited. The first distortionless marking technique that is suitable for data embedding was presented in [15]. While it is successful in distortionless data hiding, the amount of hidden data by this technique is still not large enough for, say, some medical applications. Specifically, from what is reported in [15], the estimated capacity ranges from 3k bits to 24k bits for a $512 \times 512 \times 8$ grayscale image. Another problem with the method is that when the embedding strength increases in order to increase payload, the visual quality will drop severely due to annoying artifacts.

4.1 Reversible Data Hiding based on the Histogram Modification

In this dissertation, a new reversible data embedding technique [10] is proposed, which can embed a large amount of data (5k-80k bits for a $512 \times 512 \times 8$ grayscale image) while keeping a very high visual quality for all images (the PSNR of marked images versus original images is guaranteed to be higher than 48 dB). It utilizes the zero or the minimum point of the histogram (defined below) and slightly modifies the pixel value to embed the data. This technique can be applied to virtually all types of images. So far, it has been successfully tested on more than 100 images, including medical images. The computation of the proposed technique is quite simple and the execution time is very short. This technique is suitable to be applied to medical data system.

Algorithm

The "Lena" image is used as an example to illustrate our algorithm. For a given grayscale image, say, the Lena image (512×512×8), its histogram is first generated as shown in Figure 4.1.

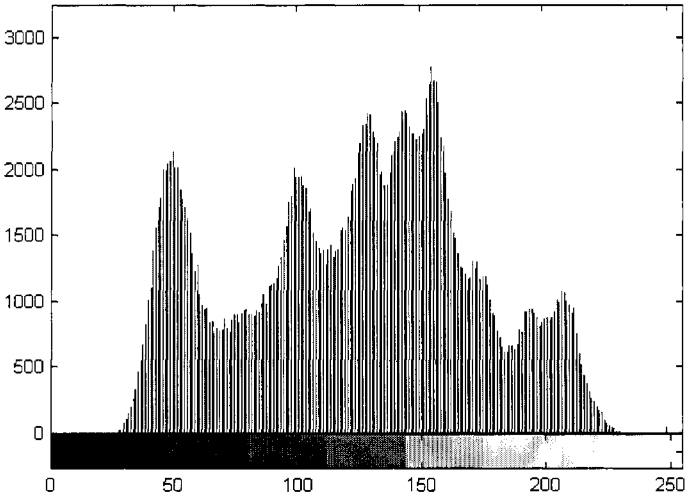


Figure 4.1 Histogram of Lena image.

A. Embedding algorithm:

1. In the histogram, a *zero point* is first found, e.g., 255, i.e., no pixel assumes the gray value of 255. Then, a *peak point* is found, e.g., 154, i.e., a maximum number of pixels assume the gray level of 154. The objective of finding the peak point is to increase the embedding capacity as large as possible.

2. The whole image is scanned in a sequential order, say, row-by-row, from top to bottom, or, column-by-column, from left to right. The gray value of pixels between 155 and 254 is incremented by "1". This step is equivalent to shifting the range of the histogram [155,254] to the right by 1 unit, leaving the gray value 155 empty.

3. The whole image is scanned once again in the same sequential order. Once a pixel with gray value of 154 is encountered, the data sequence to be embedded is checked. If the corresponding to-be-embedded bit in the sequence is binary "1", the pixel value is added by 1. Otherwise, the pixel value remains intact.

In this way, the data embedding process is completed. The capacity of this algorithm equals to the maximum number of pixels obtained in the above-mentioned Step 1.

Figure 4.2 shows the original and the marked Lena image, respectively. Figure 4.3 is the histogram of the marked image.

(a) Original (b) Marked (PSNR=48.2 dB)

Figure 4.2 Lena image.

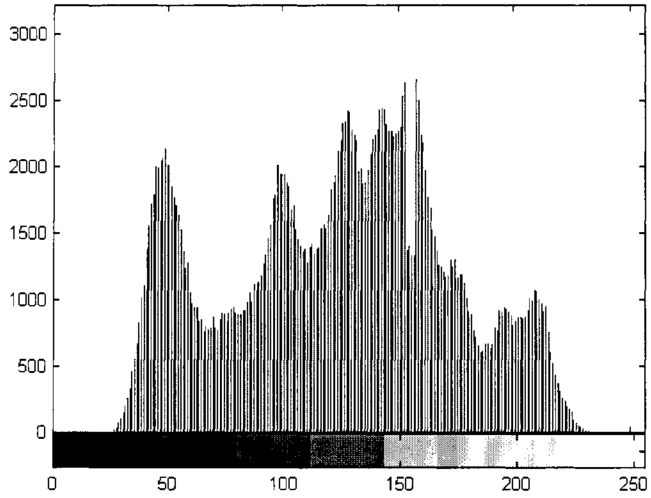


Figure 4.3 Histogram of marked Lena image.

Some comments:

- In very rare cases, the zero point is not able to be found in a histogram. For instance, the histogram shown in Figure 4.4 has no zero point. Then, the *minimum point* is used instead of the zero point. For instance, in the histogram shown in Figure 4.4, the gray value 7 is assumed by only 23 pixels. This number of 23 is the minimum number since any other gray value will be assumed by more than 23 pixels. Then, the gray value and the coordinates of the minimum point are recorded as an overhead part of the embedded data. This book-keeping information will be used later to recover the minimum point after the data retrieval.

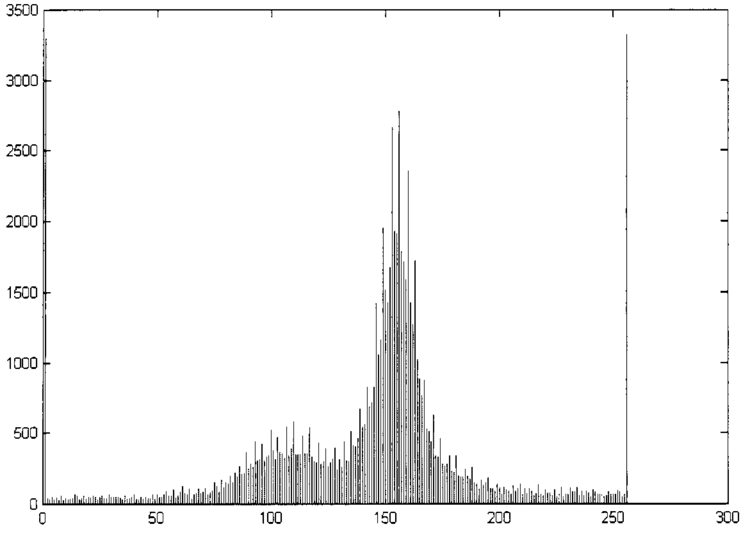


Figure 4.4 Histogram of one medical image. (Note that there are two huge peaks, 164,683 and 20,739, in the histogram, located at gray levels 0 and 255, respectively. They have been scaled down on purpose in order to show more details for the vast majority of other gray levels.)

- If there are multiple pairs of zero points and peak points, obviously, it is possible to further increase the payload by adding complexity to this algorithm. For simplicity, however, in the experiments at most two pairs of zero points and peak points are used in the above embedding algorithm. For instance, in the experiment with the Lena image, two pairs of peak and zero points are used, thus achieving a payload of 5,460 bits.

- If the gray value of zero point is greater than that of the peak point, then adding by "1" is used in Step 2. Otherwise, subtracting by "1" is used in Step 2. That is, the shifting in histogram is "two-way".

- The gray value of the zero point and the peak point will be treated as side information that needs to be transmitted to the receiving side for data retrieval.

B. Data retrieval algorithm:

For simplicity, only the case of one pair of zero point and peak point is considered here.

1. The whole marked image is scanned in the same sequential order as that used in the embedding procedure. Once the gray value of the maximum point is met, e.g., 154, the "0" is retrieved. If the altered gray value of the maximum point is met, e.g., 155, the "1" is retrieved. In this way, the data embedded before can be retrieved.

2. The whole image is scanned in the same manner once again. Once a pixel whose gray value is between the peak point, excluding the peak point (e.g., 154), and the zero point, including the zero point (e.g., 255), is met, the gray value of those pixels will be subtracted by 1. In this way, the original image can be recovered without any distortion.

Note that if the number of the zero points is zero or two, the above data retrieval algorithm can be similarly carried out with only a little alteration.

C. Embedding and Extraction Flow Charts

In summary, the proposed reversible data hiding and extraction algorithms can be illustrated by the flow charts shown in Figures 4.5 and 4.6, respectively.

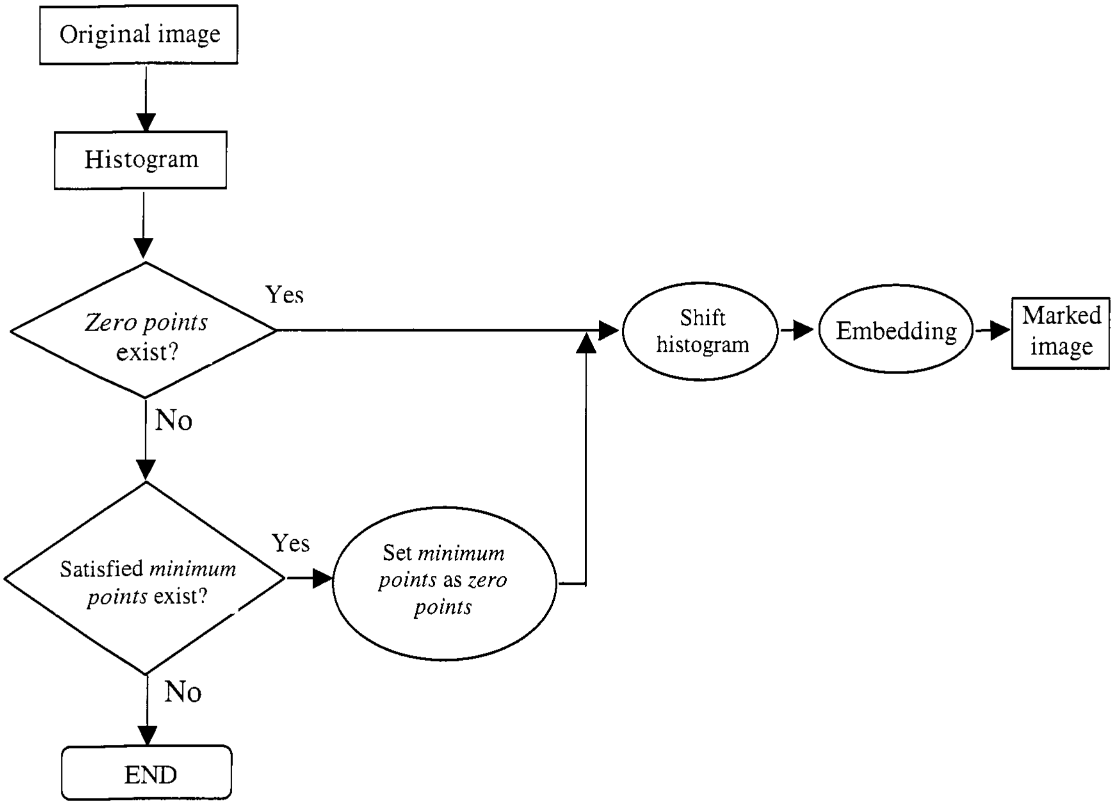


Figure 4.5 Data embedding algorithm.

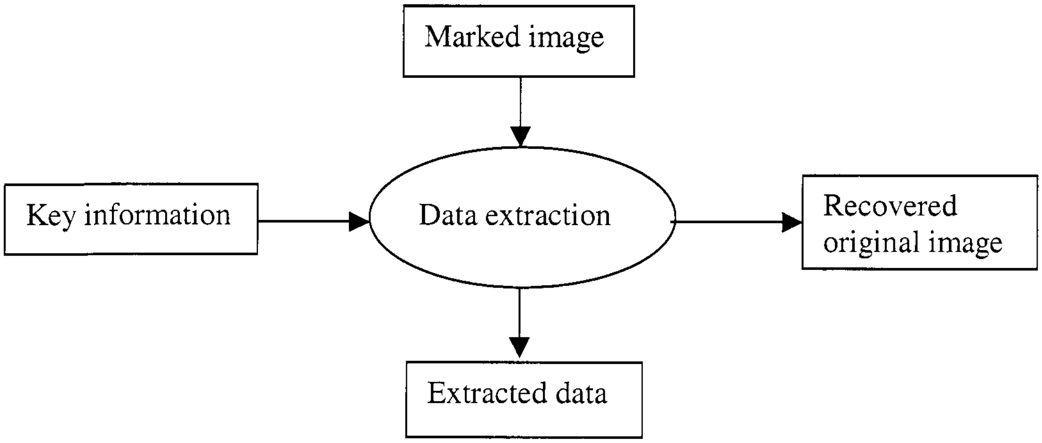


Figure 4.6 Data extracting algorithm.

D. The lower bound of the PSNR of a marked image versus the original image

In the experiments, the PSNR of all marked images is above 48 dB. This can be theoretically proved as follows. It is noted from the embedding algorithm that the pixels whose gray value is between the zero point and the peak point will be added or subtracted by 1. Therefore, in the worst case, all pixels of the image will be added or subtracted by 1, implying that the mean square of errors is at most equal to one, i.e., MSE=1. It is hence easy to see that the PSNR of a marked image versus the original image is bounded by 48.13 dB. That is, $\mathrm{PSNR} = 10 \times \log_{10}(255 \times 255 / MSE) = 48.13\,\mathrm{dB}$. The conclusion that the lower bound of the PSNR of a marked image is 48.13 dB has been validated by numerous experiments as well. To the best knowledge of the authors, this resultant PSNR is much higher than all reversible data hiding techniques reported in the literature.

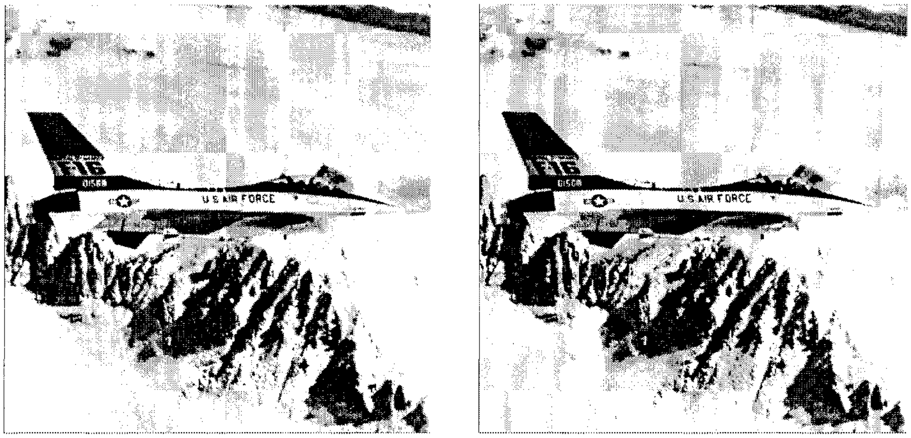
4.2 Some Experimental Results

The proposed reversible data hiding algorithm has been applied to many typical grayscale images and medical images, and has achieved satisfactory results, thus demonstrating its universal capability. Here, the results with some commonly used grayscale images and medical images are presented, in particular, the details of the "Airplane" image are provided. The mark signal in the experiment is a binary logo image as shown in Figure 4.7, equivalent to a binary sequence of 15,903 bits.

Figure 4.7 Watermark (a binary image of 171×93).

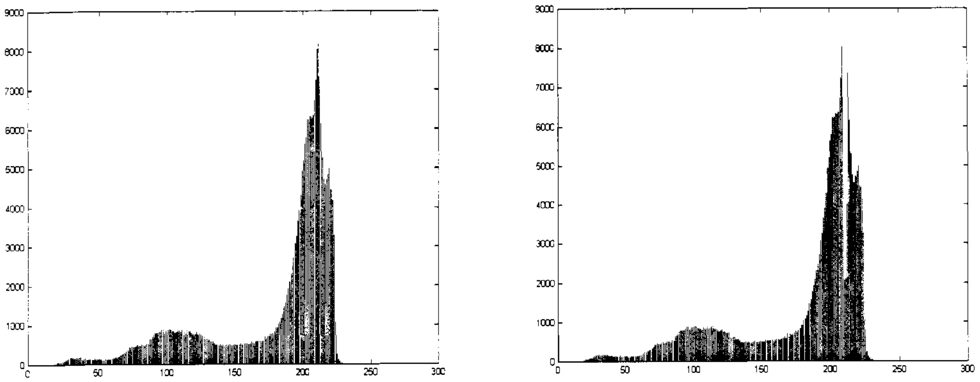
Figure 4.8 shows the original and marked Airplane image (512 × 512 × 8), respectively. Figure 4.9 shows the histogram of the original and marked Airplane image, respectively. The gray values of two zero points are 0 and 255, respectively, and the gray values of two peak points are 210 and 211, respectively. The numbers of pixels associated with these two peak points are 8,016 and 8,155, respectively. Hence, the capacity is 8,016+8,155=16,171 bits.

Figure 4.10 shows other eight marked images. Table 4.1 summarizes the experimental results. Comparison between the existing reversible marking techniques and the proposed technique in terms of capacity and PSNR is presented in Table 4.2.



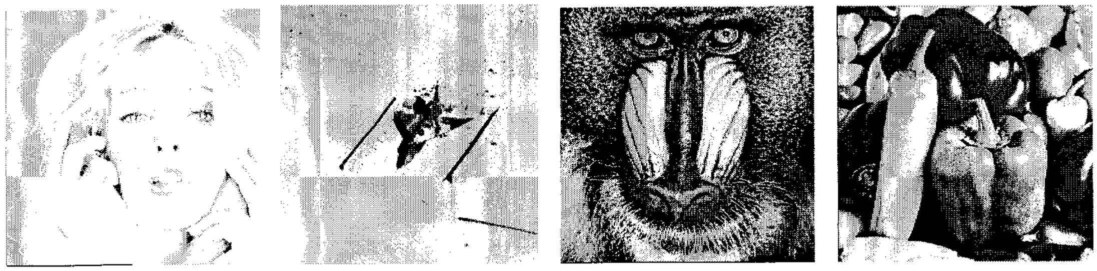
(a) Original (b) Marked (PSNR=48.2 dB)

Figure 4.8 Airplane image.

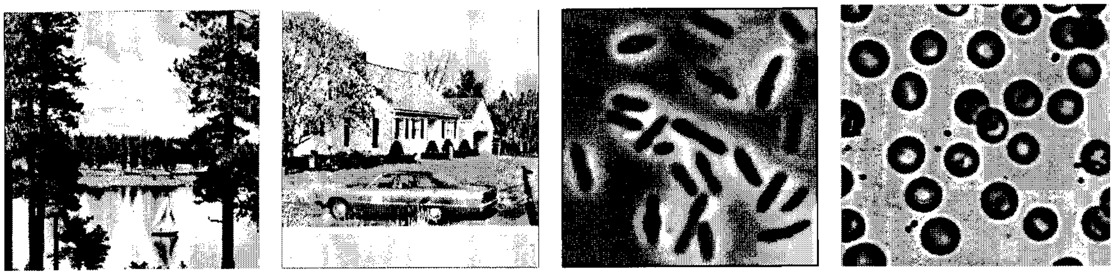


(a) the original Airplane image (b) the marked Airplane image

Figure 4.9 Histogram.



(a) Tiffany (b) Jet (c) Baboon (d) Pepper



(a) Sailboat (b) House (c) Bacteria (d) Blood

Figure 4.10 Eight marked images.

Table 4.1 Experimental Results

Images (512x512x8)	PSNR of marked image (dB)	Capacity (bits)
Lena	48.2	5,460
Airplane	48.3	16,171
Tiffany	48.2	8,782
Jet	48.7	59,979
Baboon	48.2	5,421
Pepper	48.2	5,449
Sailboat	48.2	7,301
House	48.3	14,310
Bacteria	48.2	13,579
Blood	48.2	79,460

Table 4.2 Comparison Between Three Reversible Marking Methods [14,15,16] and Proposed Method

Methods	The amount of data embedded in a 512×512×8 image	PSNR of marked image (dB)
Macq's	<2,046 bits	Not mentioned
Goljan's*	3k-24k bits	35
Xuan's	15k-94k bits	24-36
Ours	5k-80k bits	>48

CHAPTER 5

ROBUST LOSSLESS IMAGE DATA HIDING

5.1 Introduction

According to the survey [11], recently many lossless data hiding techniques have been proposed, such as in [12-19]. However, most of them are fragile in the sense that the hidden data cannot be recovered after compression or other incidental alteration is applied to the marked image. Thus far, De Vleeschouwer et al.'s method [19] is the only existing robust (or semi-fragile) lossless data hiding technique against high-quality JPEG compression. This technique can be applied to semi-fragile image authentication. That is, on the one hand, if the marked image does not change at all, the hidden data can be extracted out correctly, and the original image can be recovered losslessly, and hence it is authentic. On the other hand, if the marked image goes through compression to some extent, the hidden data can still be correctly extracted for semi-fragile authentication if the hidden data represent the compressed version of the image. Semi-fragile authentication may be more practical than fragile authentication for many applications since it allows some incidental modification, say, image compression, through which the perceived content of the image has not been changed.

The main idea of their algorithm is based on the patchwork theory [24]. Below is an introduction to the algorithm.

1. First, each bit of the hidden data is associated with a group of pixels, e.g., a block in an image. Each group is randomly divided into two sets of pixels with an equal size, named zones A and B. The histogram of each zone is mapped to a circle (Figure 5.1) where positions on the circle are indexed by the corresponding grayscale values, and the weight of a position is the number of pixels assuming the grayscale *value.*

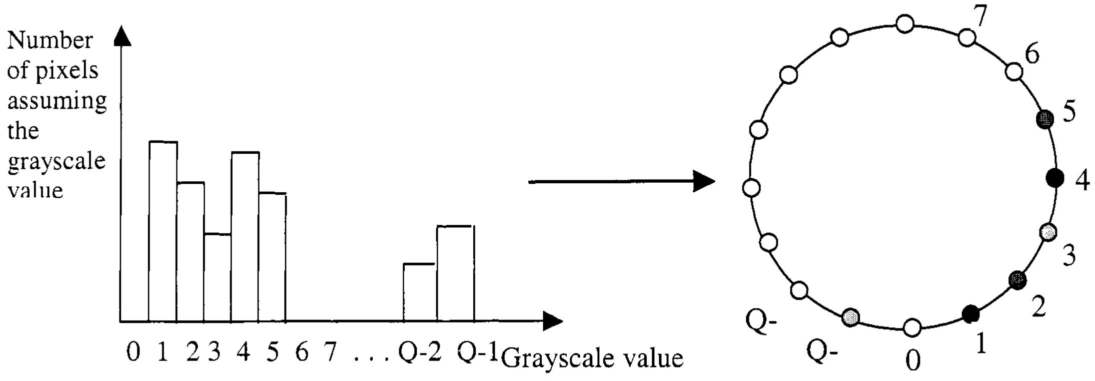
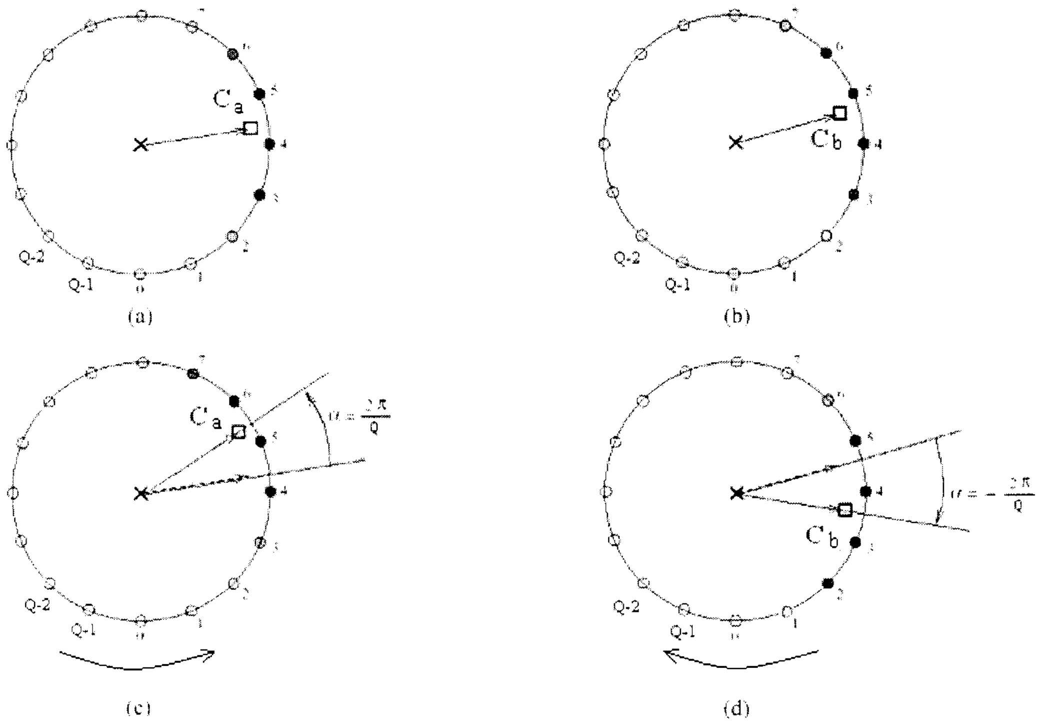

Number of pixels assuming the grayscale value
0 1 2 3 4 5 6 7 . . . Q-2 Q-1
Grayscale value
7
6
5
4
3
2
1
0
Q-
Q-

Figure 5.1 Histogram mapping onto a circle.

2. In Figure 5.2, vectors C_a and C_b point from the center of the circle to the *mass* center of zones A and B, respectively. Since zones A and B are two pseudo-randomly divided sets of equal size within the same block, it is highly probable that C_a and C_b are similar to each other. Slight rotation of these two vectors in opposite directions allows embedding a bit of information in the block. Specifically, C_a is rotated, say, clockwise and C_b is anti-clockwise to embed a bit '0', while C_a is anti-clockwise rotated and C_b is clockwise to embed a bit '1'. As to the pixel grayscale values, rotations of the vectors correspond to grayscale shifts.



(a) (b) (c) (d)

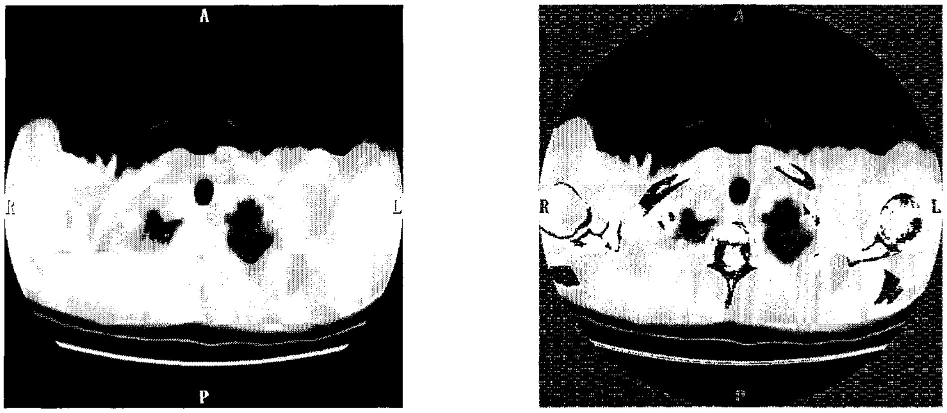
Figure 5.2 Data embedding diagram.

3. The data extraction process is actually the inverse process of the data embedding. The extraction process first partitions the image into blocks and zones A and B as the same as that in the embedding process. Histograms of each zone are mapped to the corresponding circles. For both zones, the center of mass is computed. Let V be the difference of the orientation angles between the vectors C_a and C_b. The sign of V provides the information of rotation direction during the embedding process and hence enables bit retrieval. After data extraction, the C_a and C_b can be rotated back to the original position, thus achieving the reversibility.

Apparently, the angle difference between the vectors of C_a and C_b depends on all of the pixel grayscale values in zones A and B, respectively. From the patchwork theory, it can be seen that this method is possibly robust against high-quality JPEG compression. The experimental results verify this claim.

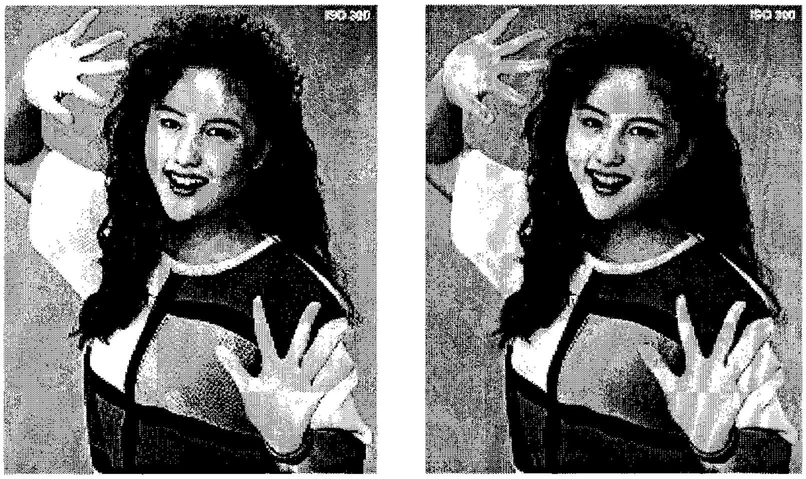
However, extensive investigation has revealed a sever problem suffered by this technique.

1. In data embedding process, one may encounter the overflow/underflow problem, which means that after data embedding, the grayscale values of some pixels in the marked image may exceed the upper bound (255 for a gray level image having eight-bit per pixel) and/or the lower bound (0 for eight-bit gray images). This situation will necessitate the use of truncation, hence violating the principle of lossless data hiding. Therefore, avoiding overflow/underflow problem is a key issue in lossless data hiding. From Figure 5.2, it is noted that modulo 256 addition is used to handle overflow/underflow problem achieving losslessness in [19]. Therefore, this algorithm generates the salt-and-pepper noise. That is, in doing modulo 256 addition, a very bright pixel with a large grayscale value close to 255 will be possibly changed to a very dark pixel with a small grayscale value close to 0, and vice versa. One example is shown in Figure 5.3 when De Vleeschouwer et al.'s algorithm is applied to a medical image, Mpic1. Obviously, severe salt-pepper noise has been observed. The noise is so dense that the name "*salt-pepper*" become *improper*. Medical images often contain many rather dark and bright pixels, hence suffering from severe salt-pepper noise. Not only for medical images, the salt-pepper noise may be severe for daily-life images as well. Figure 5.4 presents an example of severe salt-pepper noise case with a color image, Woman (a JPEG2000 test image). There, the algorithm is applied to the Red component of the image. The salt-pepper noise manifests itself as severe color distortion. Specifically, the color of a half of her hair area has changed from black to red, while the color of most of her right-hand palm area has changed from flesh color to green. Note that authors of [19] also proposed an optional method in the same paper to overcome the salt-pepper noise. However, as stated in their paper, this new method is a fragile, instead of semi-fragile lossless data hiding method.



(a) Original (b) Marked

Figure 5.3 A medical image, Mpic1.



(a) Original (b) Marked

Figure 5.4 A JPEG2000 test image, Woman.

2. Another drawback is that the marked image does not have high enough PSNR with respect to the original image. Table 5.1 and Table 5.2 summarize the performance of [19] applied to medical images and JPEG2000 test images. It is observed in Table 5.1 that, when 476 information bits are embedded in eight 512×512 medical images, the PSNR is below 30 dB. There are five marked images suffer from severe salt-pepper noise with PSNR below 10 dB. In Table 2, when 805 or 1410 bits are embedded in eight 1536×1920 JPEG2000 color test images, the PSNR is below 25 dB. Five marked images suffer from severe salt-pepper noise resulting in PSNR below 20 dB. Note that in Tables 1 and 2, *robustness (bpp)* means the surviving bit rate in the unit of bpp (bits per pixel) above or equal to which the hidden data can be retrieved with no error.

Table 5.1 Test Results for Eight Medical Images

Images (512x512)	PSNR of marked image (dB)	Data embedding capacity (bits)	Robustness (bpp)
Mpic1	9.28	476	1.0
Mpic2	4.73	476	2.0
Mpic3	26.38	476	0.8
Mpic4	26.49	476	0.6
Mpic5	26.49	476	0.6
Mpic6	5.60	476	1.6
Mpic7	9.64	476	0.8
Mpic8	5.93	476	2.8

Table 5.2 Test Results for Eight JPEG2000 Test Images

Images (1536x1920)	PSNR of marked image (dB)	Data embedding capacity (bits)	Robustness (bpp)
N1A (Woman)	17.73	1410	0.8
N2A	17.73	1410	2.2
N3A	23.73	1410	0.6
N4A	19.67	1410	1.2
N5A	17.28	1410	1.2
N6A	23.99	805	0.6
N7A	20.66	1410	1.4
N8A	14.32	805	1.4

Therefore, from the above experimental results, our observation is that all of the lossless data hiding algorithms based on modulo 256 addition (including [12,19]) are not acceptable for many practical usages. Thus, a new robust lossless data hiding technology that do not use modulo 256 addition and hence can avoid the above mentioned drawbacks is called for.

The rest of the paper is organized as follows. The proposed algorithm is described in Section 5.2. Experimental results are presented in Section 5.3, and conclusions are drawn in Sections 5.4.

5.2 A Novel Robust Lossless Image Data Hiding Algorithm

In order to be robust against JPEG/JPEG2000 compression, a robust parameter should be selected to embed data. In this proposed algorithm, the following statistic quantity is selected as the parameter.

5.2.1 Main Idea

For a given 8×8 image block, it is splitted into two subsets A and B as shown in Figure 5.5, i.e., subset A consists of all pixels marked by '+', denoted by a_i, the subset B '-', b_i. Each subset has 32 pixels.

+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+
+	-	+	-	+	-	+	-
-	+	-	+	-	+	-	+

Figure 5.5 Difference pair pattern.

For each block, the difference value α is calculated, which is defined as the arithmetic average of differences grayscale values of pixel pairs within the block. A pair may be chosen as two horizontally neighboring pixels (one is marked as '+', another '-', and each pixel is used only once). Below is the formula to calculate α. In this example, n is equal to 32.

$$\alpha = \frac{1}{n}\sum_{i=1}^{n}(a_i - b_i)$$

Since the pixel grayscale values in a local block are highly correlated and have spatial redundancy, the difference value α is expected to be very close to zero. The experimental results have supported this observation. The distribution of the difference value α of blocks of 'Boat' image is shown in Figure 5.6. Note that most values of α are very close to zero (or the mean value of this distribution is zero). The α distributions of other images also follow this pattern. Another point we should mention is that we have tried many other strategies to split subsets A and B. Among them, splitting strategy shown in Figure 5.5 seems to have the least variance of α. Hence we select this splitting strategy for achieving the least visual distortion of marked images versus the original image.

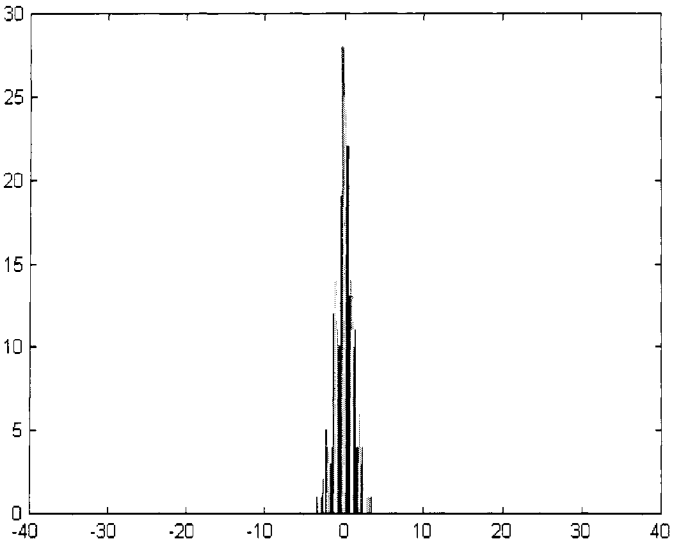


Figure 5.6 The distribution of difference value α.

Since the difference value α is based on the statistics of all pixels in the block, this value α has certain robustness against JPEG/JPEG2000 compression and other small incidental alteration. This difference value α is selected as the robust quantity for data embedding.

Note that the block size is not necessary to be 8×8. Since, as shown below, each block is used to embed one bit, the block size will affect embedding capacity. The robustness of embedded bits, on the other hand, will be stronger if the block size is larger. Therefore, a compromise between the data embedding capacity and robustness of hidden data needs to be made according to specific applications.

5.2.2 Bit Embedding Strategy

A cover image is divided into non-overlapping blocks. Then one bit is embedded in each block. The main idea for bit embedding is that the difference value α is kept within a specified threshold K and –K (usually K is less than 5 in our numerous experiments) to embed bit '0' and the difference value α is shifted beyond the threshold K or –K to embed bit '1'. As analyzed in Section 1, although modulo 256 addition can avoid the overflow/underflow problem, we have decided not to use it since it will lead to annoying salt-pepper noise. In order to overcome this overflow/underflow problem, the blocks are classified into four categories and use different bit embedding schemes for each category. Theory and experimental results show that this technique successfully solves the overflow/underflow problem and avoids the salt-pepper noise at the same time. Below are the detailed bit embedding steps. In the algorithm, the shift quantity β is usually twice of the specified threshold K. Note that shifting α towards the right-hand side means adding a fixed shift quantity, β, to each pixel grayscale value marked by '+' in subset A, and shifting α towards the left-hand side means subtracting a fixed shift number, β, from each pixel grayscale value marked by '+' in subset A. In the whole bit embedding process, the pixel grayscale values marked by '-' in subset B are kept intact, reducing the distortion caused by bit embedding. Since error bits may be introduced in data embedding, error correction coding (ECC) is applied to correct them.

Category 1: The pixel grayscale values of a block under consideration are far enough away from two bounds of histogram (0 and 255 for an 8-bit gray level image). That is, the distance $d = \min(d_l , 255 - d_r)$ satisfies $d \geq \beta$ (where β is the shift quantity), as shown in Figure 5.7.

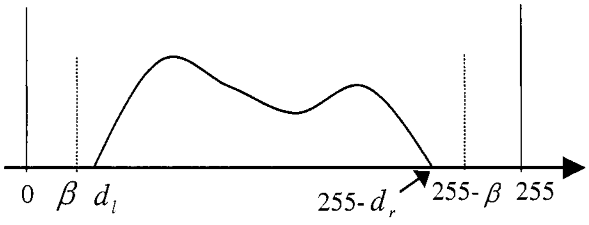


Figure 5.7 Block histogram for category 1.

In this category, the following two cases are further considered according to the value of α.

Case 1. The difference value α is located between the thresholds K and -K.

1. If the to be embedded bit is '1', we shift the difference value α by a quantity β towards the right-hand side or left-hand side depending on if α is positive or negative. Refer to Figure 5.8.
2. If to be embedded bit is '0', the pixel value of that block is intact.

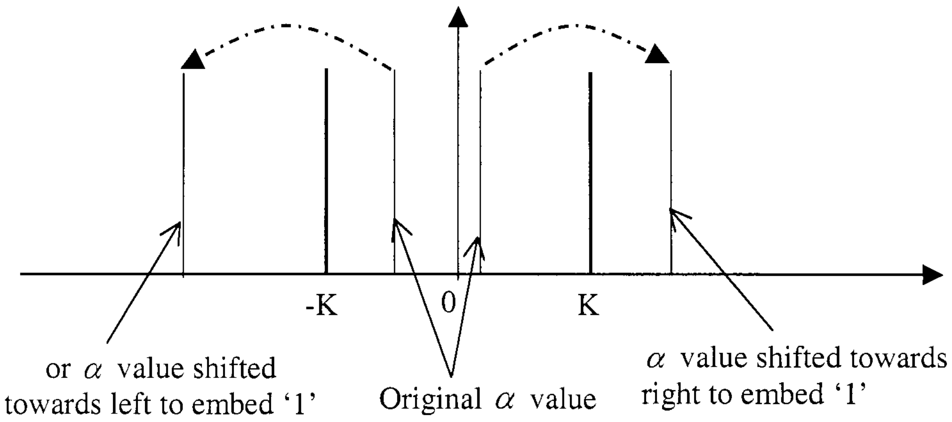

-K
0
K
or α value shifted towards left to embed '1'
Original α value
α value shifted towards right to embed '1'

Figure 5.8 Embedding a bit '1'.

Case 2. The absolute value of α value exceeds the threshold K.

In order to keep the lossless data hiding principle, no matter whether the to-be-embedded bit is '0' or '1', it always embeds bit '1' by shifting the difference value α further away from 0 by a quantity β, as shown in Figure 5.9. In this way, it may introduce an error bit, which will be corrected by using ECC.

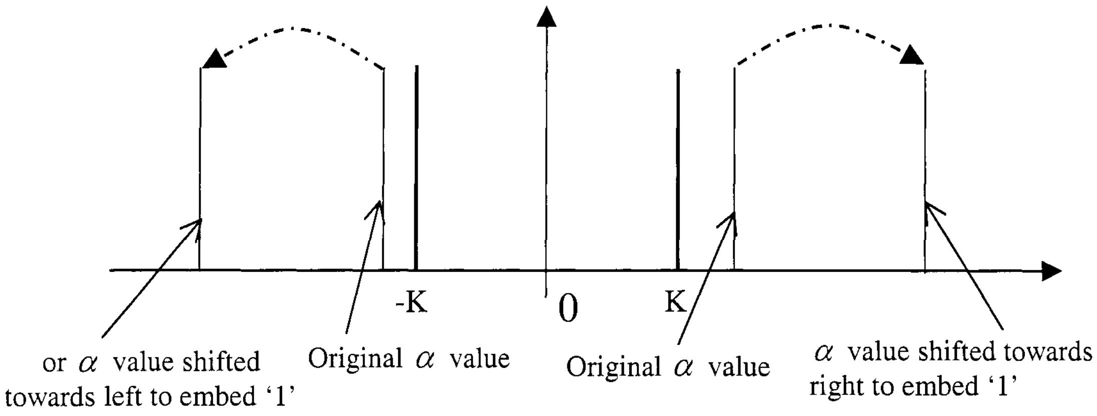

-K
0
K
or α value shifted towards left to embed '1'
Original α value
Original α value
α value shifted towards right to embed '1'

Figure 5.9 Embedding a bit '1'.

Category 2: Some pixel grayscale values of the block under consideration are very close to the lower bound of the histogram, 0, while no pixel grayscale values are close to the upper bound of the histogram, as shown in Figure 5.10.

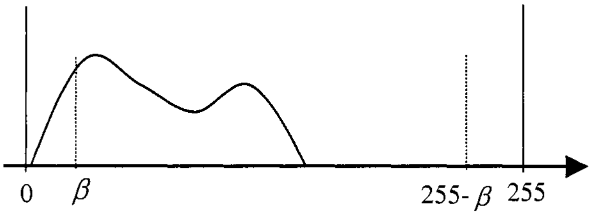


Figure 5.10 Block histogram for category 2.

In this category, three different cases are further considered according to the value α.

Case 1. The value α is located between the thresholds K and -K.

1. If the to-be-embedded bit is '1', we always shift the difference value α by a quantity β towards the right-hand side beyond the threshold K. Refer to Figure 5.11.
2. If the to-be-embedded bit is '0', the pixel value of that block is intact.

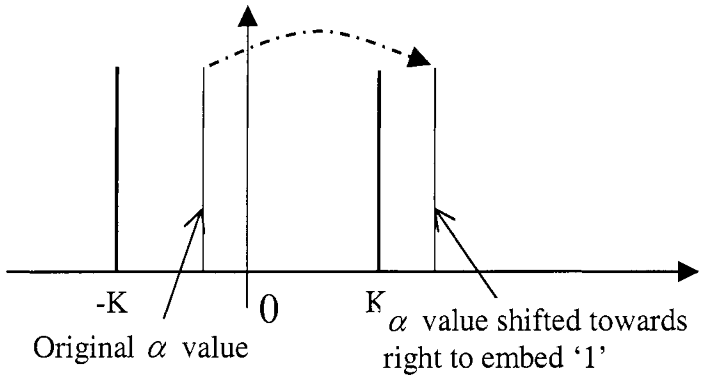


Figure 5.11 Embedding a bit '1'.

Case 2. The value α is located on the right-hand side beyond the threshold K.

No matter whether the to-be-embedded bit is '0' or '1', it always embeds bit '1' by shifting the difference value α by a quantity β further leaving away from the zero point, as shown in Figure 5.12. In this way, it may introduce an error bit, which will be corrected by using ECC.

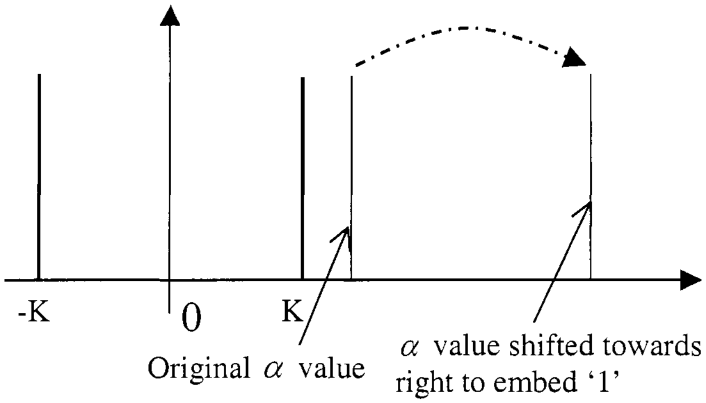


Figure 5.12 Embedding a bit '1'.

Case 3. The value α is located on the left-hand side beyond the threshold -K.

If this case is met, the threshold K or the block size will increase until no value α is located on the left-hand side beyond the threshold -K. That means case 3 becomes case 1. The error bits introduced will be corrected by ECC code. In this way, side information about the coordinates of these blocks can be avoided.

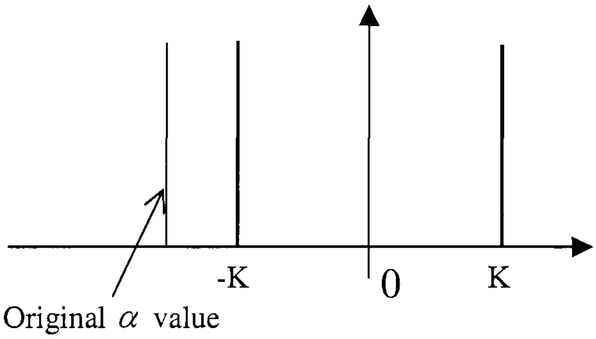


Figure 5.13 Increase threshold or block size to embed bit '0'.

Category 3: Some pixel grayscale values of the block under consideration are very close to the upper bound of the histogram, while no pixel grayscale values are close to the lower bound of the histogram, as shown in Figure 5.14.

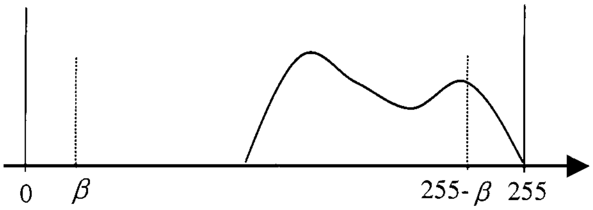


Figure 5.14 Block histogram for category 3.

Category 3 is similar to category 2 except that the distribution of grayscale values of the block is close to the upper bound instead of the lower bound of the histogram. Hence, data embedding algorithm of category 3 is similar to that of category 2 except shifting difference value α by a quantity β to the left-hand side instead of to the right-hand side.

Category 4: Some pixel grayscale values of the block under consideration are close to the upper bounds, while some pixel grayscale values are close to the lower bounds of the histogram, as shown in Figure 5.15.

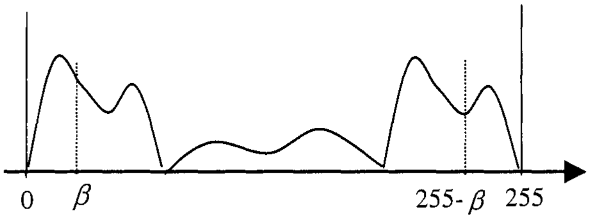


Figure 5.15 Block histogram for category 4.

In this category, two different cases are further considered according to the α value.

Case 1. The value α is located between the thresholds K and -K.

No matter whether the to-be-embedded bit is '0' or '1', it always embeds bit '0' by keeping the difference value α intact, as shown in Figure 16. In this way, it may introduce an error bit, which is to be corrected by using ECC.

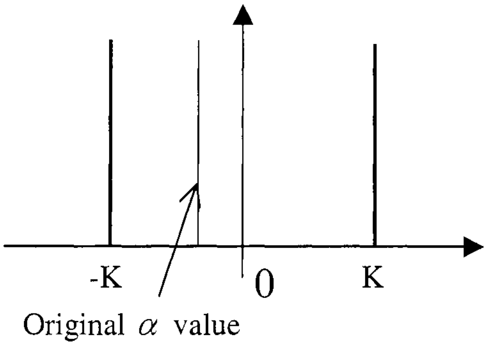


Figure 5.16 Embedding a bit '0'.

Case 2. The absolute value α is beyond the threshold K.

No matter whether the to-be-embedded bit is '0' or '1', nothing happens to this block, as shown in Figure 5.17. The constantly bit '0' or bit '1' is to be embedded in this block to avoid recording the side information of the block coordinate. The error bits introduced in this way can be corrected by ECC.

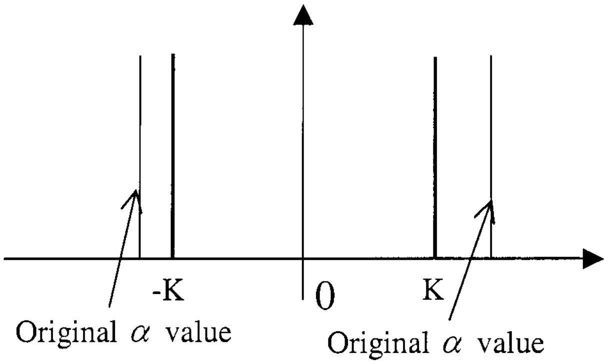


Figure 5.17 The whole block is intact.

The above-mentioned four categories cover all situations that a block may encounter. The detailed description of bit embedding apparently demonstrates that the modified pixel grayscale value is still in the range of [0,255] and hence, no overflow/underflow problem will take place. It is noted that after data embedding some error bits will be generated. They will be handled as follows.

5.2.3 Error Correction Code

In the above-mentioned bit embedding process, it may introduce some error bits. In order to recover the information bits correctly, error correction code (ECC) is utilized to correct error bits at the price of sacrificing some data embedding capacity. In the algorithm, a few BCH codes [35] are included for selection. They are BCH (15,11,1), BCH (15,7,2), BCH (15,5,3), BCH (31,6,7) and BCH (63,7,15). The reason to have these BCH codes for selection is to facilitate trade-off between the coding ratio of ECC (hence the payload) and the error correction capability of ECC (therefore, robustness of lossless data hiding). For example, BCH (63,7,15) code is the most powerful code among these several codes in terms of error correction capability. It can correct 15 random error bits within a codeword of 63 bits with the price of more redundant bits are introduced, resulting in a smallest payload among these codes.

5.2.4 Chaotic Mixing

In some special images, error bits may concentrated in some small areas in the image, which leads to too many error bits in one codeword, thus causing error in data extraction. In this case, even the powerful code BCH (63,7,15) cannot correct the error bits. To combat this type of bursts of errors, which may fail the algorithm, chaotic mixing [36] is introduced on the watermark matrix (a square matrix formed by the to-be-embedded bits) to spread the error bursts evenly in the whole watermark matrix. Let $\boldsymbol{r} = (x,y)^T$ be the location of an element in the watermark matrix, and $\boldsymbol{r}'=(x',y')^T$ be the new location of the element in the mixed watermark matrix, we have $\boldsymbol{r}' = \boldsymbol{A}\ \boldsymbol{r}$, where $\boldsymbol{A} = \begin{bmatrix} 2 & 1 \\ 1 & 1 \end{bmatrix}$ in the algorithm. Note that chaotic mixing [36] is only applicable to square matrixes. But most test images are rectangular. Hence, a mapping from rectangular to square is needed. An example of chaotic mixing is shown in Figure 5.18.

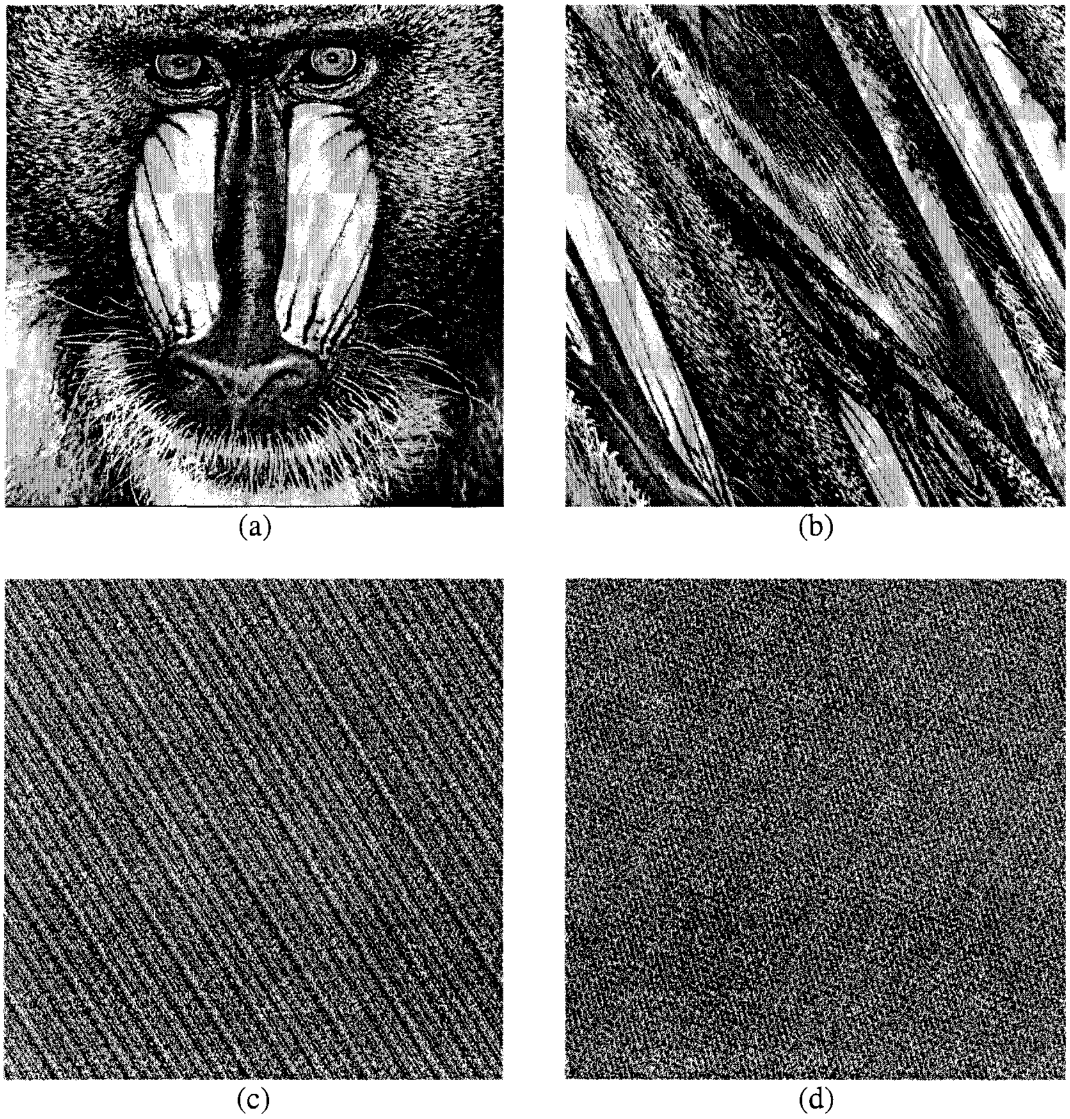

(a) (b) (c) (d)

Figure 5.18. Chaotic mixing of Baboon image by A^n, a) original image, b) n=1, c) n=5, d) n=10, where *n* is the numbers the chaotic mixing has applied.

5.2.5 Data Embedding Diagram

With above mentioned techniques, the data embedding diagram are constructed as shown in Figure 5.19.

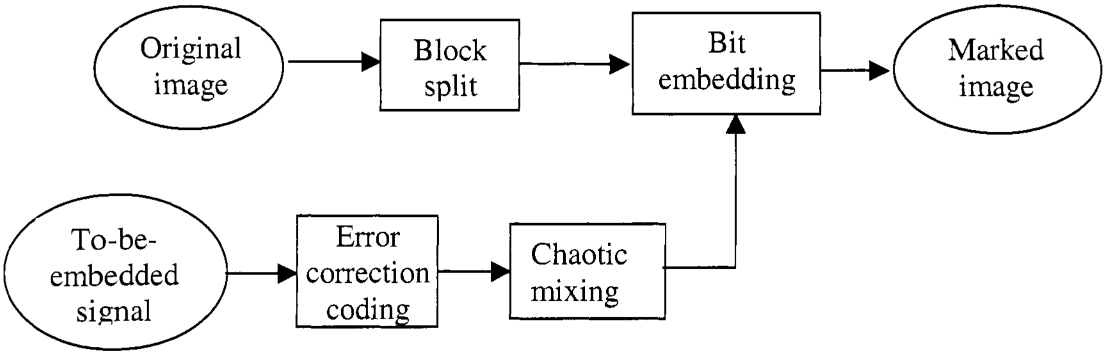


Figure 5.19 Block diagram of data embedding.

5.2.6 Data Extraction

Data extraction is actually the reverse process of data embedding and is much simpler than data embedding. For a given marked image, it is first splitted into non-overlapping blocks and then the difference value α for each block is calculated in the same way as that used in data embedding. The main steps are described below.

1. If the absolute difference value α is larger than the threshold K, we then check if the block belongs to case 2 of category 4. If it belongs to that case, the block is intact and the default bit '0' or bit'1' is extracted. Otherwise, bit '1' is extracted and the difference value α is shifted back towards the zero point by adding or subtracting a quantity β, depending on the difference value α is negative or positive. In this way, difference value α is back to its original value, which means each pixel grayscale value in subset A is back to its original value, as shown in Figure 5.20.

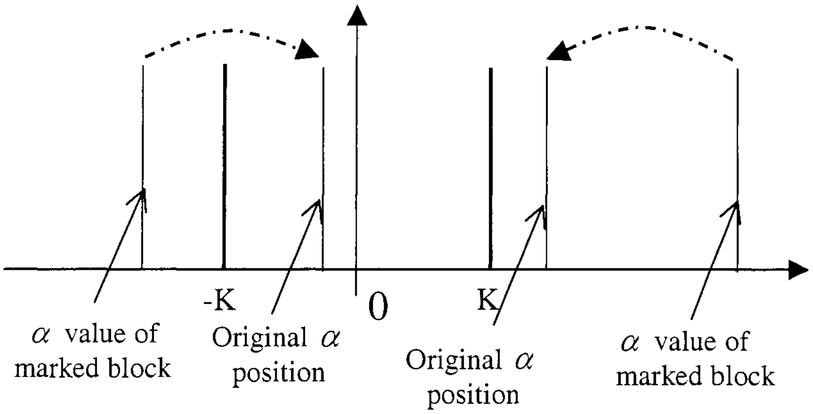

-K
0
K
α value of marked block
Original α position
Original α position
α value of marked block

Figure 5.20 Extracting bit '1'.

2. If the absolute value of the difference value α is less than the threshold K, then bit '0' is extracted and nothing to do on the pixel grayscale value of that block.

3. After data extraction, inverse chaotic mixing and ECC decoding are applied, respectively, so as to obtain the original information bits correctly.

In this way, the original information bits can be extracted and the original image is recovered without any distortion. The whole data extraction diagram is depicted in Figure 5.21.

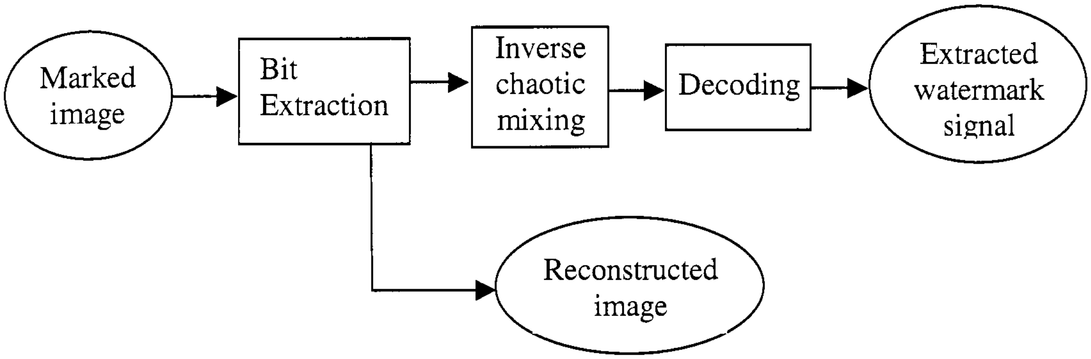
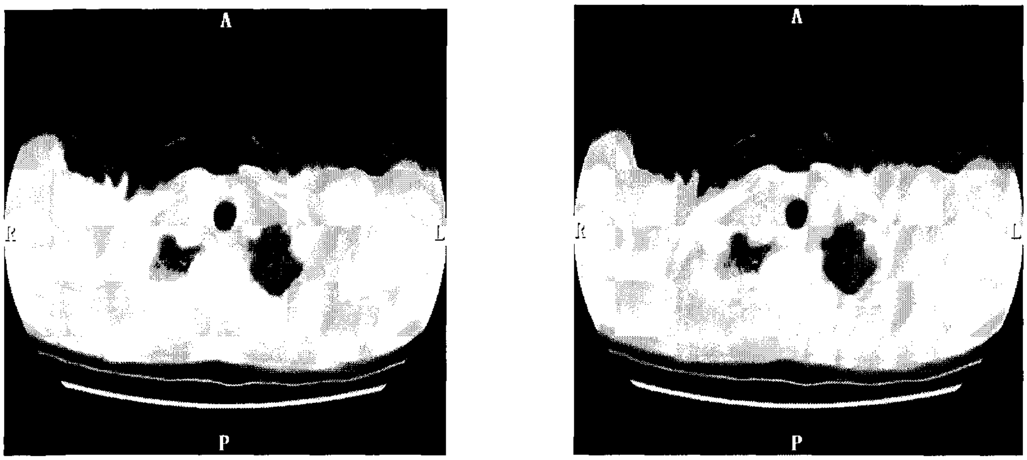


Figure 5.21 Block diagram of data extraction.

5.3 Experimental Results

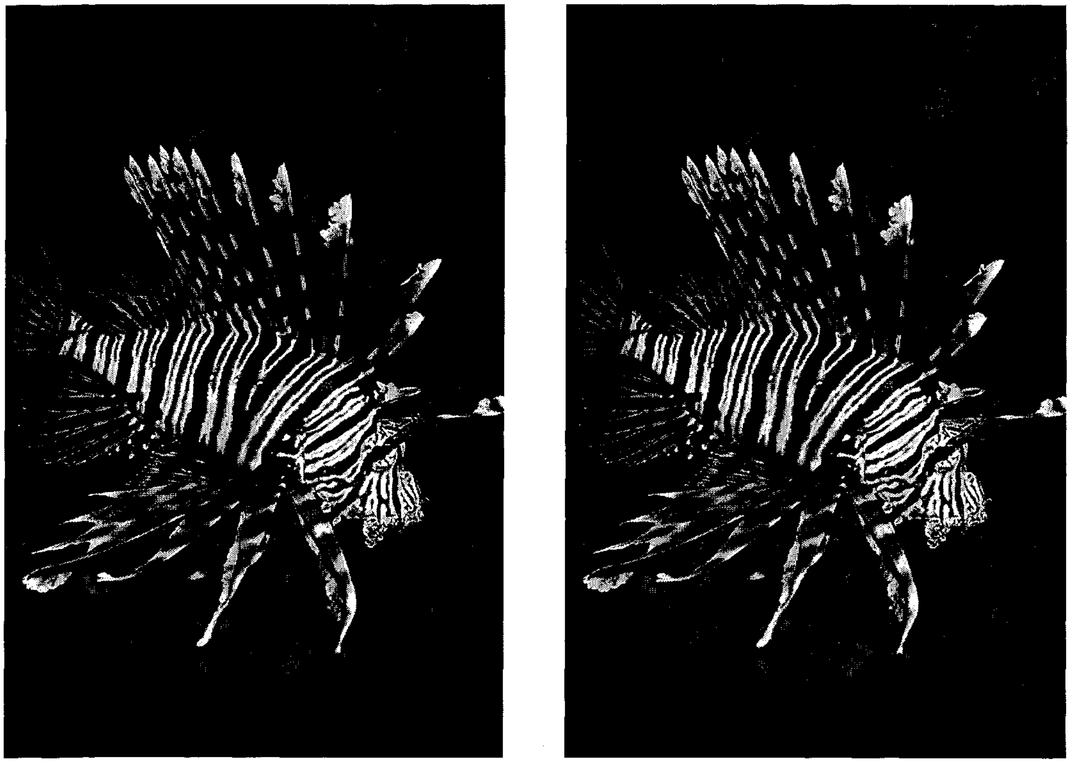
The proposed algorithm has been successfully applied to some commonly used grayscale images such as 'Lena', 'Baboon', etc., eight medical images, more than 1000 images in CorelDraw image database, and eight JPEG2000 color test images. For color images, only one color plane is applied by the algorithm. Note that salt-pepper noise is not generated at all since we do not use modulo 256 addition in our algorithm. The embedding capacity can range from 512 to 1024 bits for authentication purpose and it is adjustable for other applications. The PSNR is above 39 dB on an average basis. It is noted that the data embedding capacity and the PSNR of the marked image versus the original image can be adjusted according to the requirements and they are usually conflicting each other in the sense that if the embedding capacity is improved, the PSNR will drop and vice versa. The tested images can resist the JPEG2000 compression attack with the surviving bit rate ranging from 2.0 bpp to 0.2 bpp. It means that the hidden data can be retrieved without error when image compression is applied to marked images with the resultant bit rate in the unit of bpp (bits per pixel) equal to or above this range.

The followings are some test examples. Note that no visible artifacts exist, indicating a significant performance improvement has been achieved as compared with [19]. It needs to mention that there is no color distortion at all in the marked Woman image as shown in Figure 24(b).



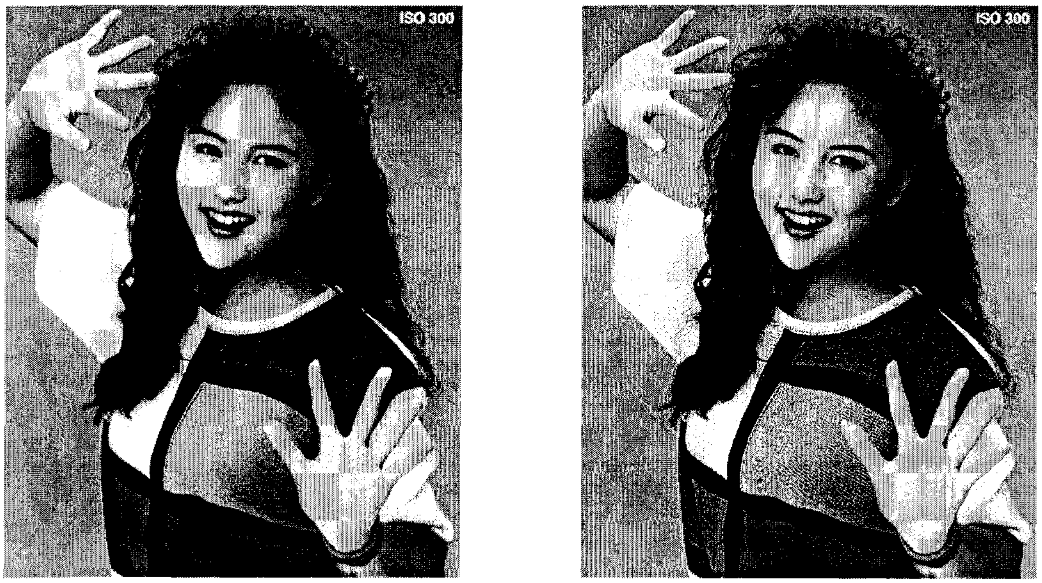
(a) Original (b) Marked

Figure 5.22 A medical image, Mpic1.



(a) Original (b) Marked

Figure 5.23 A CoralDraw image.



(a) Original (b) Marked

Figure 5.24 A JPEG2000 test image, Woman.

Tables 5.3, 5.4, 5.5 and 5.6 summarize test results for commonly used images, 1000 images in CorelDraw database, eight medical images and eight JPEG2000 test images, respectively. Note that the embedding capacity reported in Tables 5.5 and 5.6 are very close to that in Tables 5.1 and 5.2.

Table 5.3 Test Results for Commonly Used 512×512×8 Grayscale Images

	Lena	Baboon	Boat
PSNR (dB)	40.2	38.7	40.5
Capacity (bits)	792	585	560
Robustness (bpp)	0.8	1.6	1.0

Table 5.4 Test Results for 1000 Images in CorelDraw Database

Images (512×768)	PSNR of marked image (dB)			Data embedding capacity (bits)	Robustness (bpp)		
	Max	Min	Avg	714	Max	Min	Avg
	45.2	37.4	40.2		2.0	0.2	1.21

Table 5.5 Test Results for Eight Medical Images

Images (512×512)	PSNR of marked image (dB)	Data embedding capacity (bits)	Robustness (bpp)
Mpic1	40.4	768	0.8
Mpic2	40.8	560	0.8
Mpic3	40.3	792	0.6
Mpic4	40.3	792	1
Mpic5	40.3	792	0.8
Mpic6	40.7	560	0.8
Mpic7	40.4	768	0.4
Mpic8	40.6	560	0.8

Table 5.6 Test Results for Eight JPEG2000 Color Test Images

Images (1536×1920)	PSNR of marked image (dB)	Data embedding capacity (bits)	Robustness (bpp)
N1A(Woman)	45.1	1398	0.8
N2A	43.1	1398	1.6
N3A	45.1	1398	1
N4A	45.2	1398	1
N5A	45.5	1200	1
N6A	45.0	1267	0.4
N7A	40.6	1398	1.2
N8A	41.5	798	1.4

5.4 Authentication Application

5.4.1 Content Signing

In this application, the embedded data is a digital signature produced from the content feature. We first extract the content feature from the image, then use one way hash and private/public key encryption to get the digital signature. The length is 1024 bits or 512 bits. The process is shown in Figure 5.25.

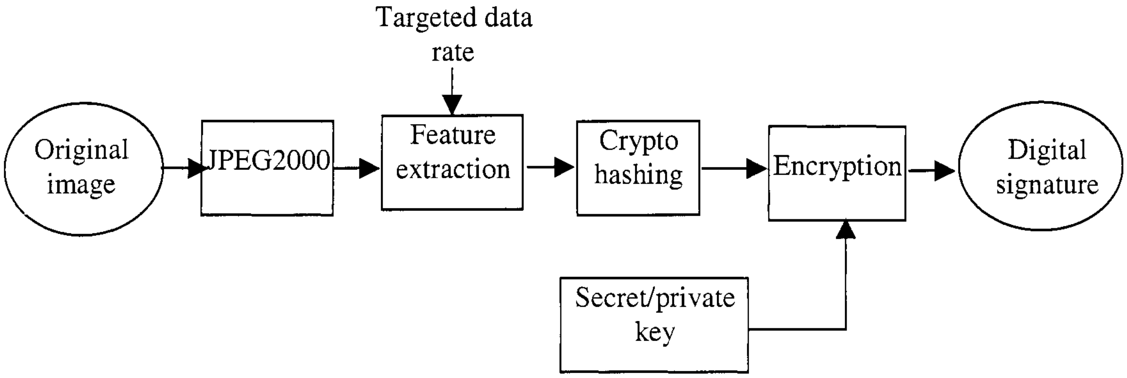


Figure 5.25 Block diagram of content signing.

5.4.2 Signature Embedding

The digital signature is embedded into the original image according to the lossless data hiding algorithm in section 5.2 to get the watermarked image.

5.4.3 Authentication

The whole authentication process is shown in Figure 5.26. The extracted mark and the reconstructed image are gotten from the watermarked image according to the extraction technique in section 5.2.

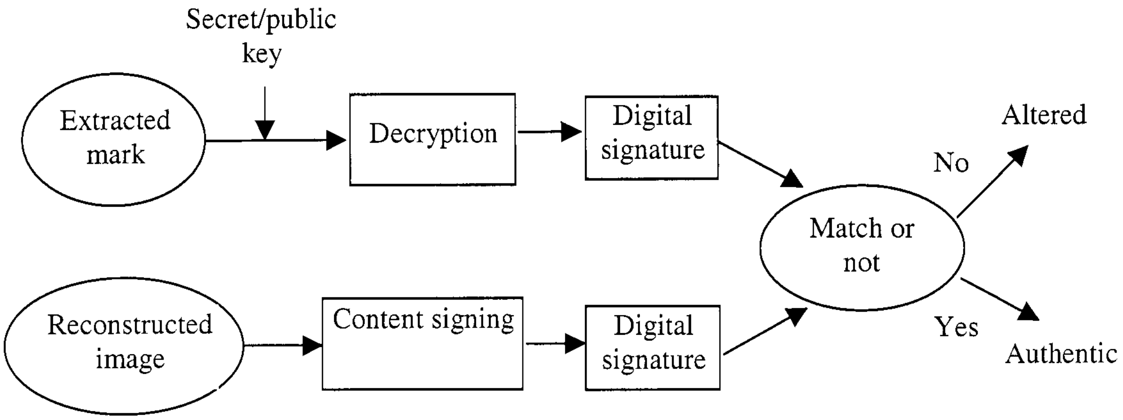


Figure 5.26 Block diagram of authentication.

Note:

Localization: the authentication process can also be used to check which local part of the image has been changed if the image has been altered. If the local extracted bit does not match the produced signature bit, it shows that block has been changed.

5.5 Conclusion

A novel robust lossless image data hiding technique is proposed, which employs a statistical quantity, which is robust to image compression and small incidental alteration, for data embedding, thus successfully avoiding annoying salt-and-pepper noise. This technique has the following advantages: 1) no salt-and-pepper noise; 2) applicable to virtually all the images (including commonly used images, medical images, more than 1000 images in CorelDRAW database, and JPEG2000 test images); 3) average PSNR of marked images is above 39 dB; 4) robust to JPEG/JPEG2000 compression to a certain extent; 5) data embedding capacity ranges from 512 bits to 1024 bits (often sufficient for authentication purpose), and the embedding capacity can be adjusted according to the requirement.

This proposed scheme [37] has been utilized in a unified authentication framework for JPEG2000 images [38], in which the proposed technique is used to authenticate losslessly compressed JPEG2000 images, possibly followed by transcoding.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Robustness and lossless of the image data hiding are investigated in this dissertation. Since most image data hiding algorithms are lossy and only have robustness to random errors, this dissertation presents an algorithm based on the interleaving technique to combat burst errors and a lossless image data hiding technique. Both of them obtain pretty good results. Besides that, this dissertation proposes a novel robust and lossless image data hiding technique, which totally avoids the pepper-and-salt noise existed in other algorithms. So far, this technique is the most front development in this field.

Chapter 1 first gives the detailed introduction about image data hiding (or watermarking) and provides some background knowledge. Then it points out some probable applications of the image data hiding techniques. Finally, it presents a brief introduction of the whole dissertation.

Chapter 2 classifies the existing algorithms into different big categories (such as spatial domain, DCT domain, wavelet domain, etc.) and presents some typical examples in each category. From these examples, readers are able to obtain some basic knowledge about what kind of popular techniques used in the image data hiding fields.

Chapter 3 concentrates on the robustness of image data hiding against burst errors. It proposes to apply interleaving technique and error correction code to combat the burst errors. From the experimental results in 2D and 3D scenarios, it shows the significant improvement of robustness against burst errors.

Chapter 4 presents a novel lossless image data hiding algorithm. The most important advantage of this algorithm is that this algorithm guarantees that the PSNR of the marked image is always above 48 dB. The experimental results show that it is better than most existing lossless data hiding techniqyes.

In Chapter 5, a novel robust lossless image data hiding technique is proposed. Most existing lossless image data hiding techniques are fragile. The only existed robust lossless image data hiding algorithm has a fatal drawback, severe pepper-and-salt noise. The proposed novel technique overcomes the above drawbacks and obtains pretty good experimental results. This technique is the nearest development in this field.

The future work of this dissertation will focus on the further robustness improvement of the technique proposed in Chapter 5 and explore the new research area--stegnography, a technique used to detect if any information are embedded in the image. The robustness of the technique in Chapter 5 is not very high enough. In the very low bit rate compression, such as 0.1 bpp, the proposed technique will fail. Hence, it is a promising field for further research. Another promising research area is the stegnography. With the popular of image data hiding (or watermarking), more and more people will use marked image as a covert communication media for crime purpose. Hence, for any given images, it needs to develop a set of algorithms to detect if any secret information inside and what the information is. The research direction may be in the feature detection, statistics and pattern recognition.

REFERENCES

1. B. Macq and J. Quisquater. "*Cryptology for digital TV broadcasting*," Proc. of the IEEE. Vol. 83. No. 6, pp. 944-957, June 1995.

2. G. Braudaway, K. Magerlein. And F. Mintzer. "*Protecting publicly available images with a visual image watermark*," *Proc. SPIE: Optical Security and Counterfeit Deterrence Techniques*, vol. 2659. pp. 126-133, 1996.

3. M. Schneider and S. Chang, *"A robust content based digital signature for image authentication,"* in Proc. IEEE Int. Conf. On Image processing, 1996.

4. Digimarc Corporation. (Oct. 2004). Available WWW: http://www.digimarc.com/

5. F. A. P. Petitcolas, R. J. Anderson and M. G. Kuhn, "Attacks on copyright marking systems," *Second Workshop on Information Hiding*, Portland, OR, USA, April 1998.

6. Zhicheng Ni, Yun Q. Shi, Nirwan Ansari, "Stirmark Attack Resistant Fractal Transform-based Information Hiding," *The Seventh International Conference on Distributed Multimedia Systems*, Tamkang University, Taipei, Taiwan, September 26-28, 2001.

7. J. Huang, G. Elmasry and Y. Q. Shi, "Power constrained multiple signaling in digital image watermarking," *Proceedings of 1998 IEEE Workshop on Multimedia Signal Processing*, pp. 388-393, Los Angeles, CA, USA, December 1998.

8. Y. Q. Shi and X. M. Zhang, "A New Two-dimensional Interleaving Technique Using Successive Packing", *IEEE Transactions on Circuits and Systems I: Fundamental Theory and Applications*, vol. 49, no. 6, pp. 779–789, June 2002.

9. Yun Q. Shi, Zhicheng Ni, Nirwan Ansari and Jiwu Huang, "2-D and 3-D Successive Packing Interleaving Techniques and their Applications to Image and Video Data Hiding," *IEEE International Symposium on Circuits and Systems,* Bangkok, Thailand, May, 2003.

10. Zhicheng Ni, Yun Q. Shi, Nirwan Ansari and Wei Su, "Reversible Data Hiding," *IEEE International Symposium on Circuits and Systems,* Bangkok, Thailand, May, 2003.

11. Y. Q. Shi, Z. Ni, D. Zou, C. Liang and G. Xuan, "Lossless data hiding: Fundamentals, algorithms and applications," *Proceedings of IEEE International Symposium on Circuits and Systems*, Vancouver, Canada, May 2004.

12. C. W. Honsinger, P. Jones, M. Rabbani, and J. C. Stoffel, "Lossless recovery of an original image containing embedded data," US Patent: 6,278,791, 2001.

13. J. Fridrich, M. Goljan and R. Du, "Invertible authentication," *Proc. SPIE Photonics West, Security and Watermarking of Multimedia Contents III*, vol. 397, pp. 197-208, San Jose, California, January 2001.

14. B. Macq and F. Deweyand, "Trusted headers for medical images," *DFG VIII-D II Watermarking Workshop*, Erlangen, Germany, Oct. 1999.

15. M. Goljan, J. Fridrich, and R. Du, "Distortion-free data embedding," *Proceedings of 4th Information Hiding Workshop*, pp. 27-41, Pittsburgh, PA, April 2001.

16. G. Xuan, J. Zhu, J. Chen, Y. Q. Shi, Z. Ni, W. Su "Distortionless data hiding based on integer wavelet transform," *Proceedings of IEEE International Workshop on Multimedia Signal Processing*, St. Thomas, US Virgin islands, December 2002. *IEE Electronics Letters*, vol. 38, no. 25, pp. 1646-1648, Dec. 2002

17. M. Celik, G. Sharma, A.M. Tekalp, E. Saber, "Reversible data hiding," in *Proceedings of the International Conference on Image Processing 2002*, pp. 157-160, Rochester, NY, September 2002.

18. J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890-896, August 2003.

19. C. De Vleeschouwer, J. F. Delaigle and B. Macq, "Circular interpretation of bijective transformations in lossless watermarking for media asset management," *IEEE Tran. Multimedia,* vol. 5, pp. 97-105, March 2003.

20. L. F. Turner, "*Digital data security system*," Patent IPN WO 89/08915, 1989.

21. R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne, "*A digital watermark*," in Proc. IEEE Int. Conf. Image Processing, vol. 2, pp. 86-90, 1994.

22. E. Koch and J. Zhao, "*Towards robust and hidden image copyright labeling*," in Proc. IEEE Workshop on Nonlinear Signal and Image Processing, June 1995.

23. J. R. Smith and B. O. Comiskey, "*Modulation and information hiding in images*," in Proc. First Int. Workshop on Information Hiding, R. Anderson, ed., vol. 1174 of Lecture Notes in Computer Science, pp. 207-226, Springer-Verlag, 1996.

24. W. Bender, D. Gruhl, N. Morimoto, and A. Lu, "*Techniques for data hiding*," IBM Systems Journal 35(3/4), pp. 313-336, 1996.

25. M. M. Yeung, and F. C. Mintzer, "*Invisible watermarking for image verification*," In Journal of Electronic Imaging, vol. 7(3), pp. 578-591, July 1998.

26. J. Cox, J. Kilian, T. Leighton, and T. Shamoon, "*Secure Spread Spectrum Watermarking for Multimedia*," in IEEE Trans. on Image Processing, vol. 6. 12, pp. 1673-1687, Dec. 1997.

27. Charles G. Boncelet, Xiang Gen Xia, and Gonzalo. R. Arce, "*A multisolution watermark for digital images*," in International Conference on Image Processing (ICIP 97), Santa Barbara, CA, July 1997. IEEE Signal Processing Society.

28. Chiou-Ting Hsu, and Ja-Ling Wu, "*Multiresolution watermarking for digital images*," in IEEE Trans. on Circuits and Systems-II: Analog and digital Signal Processing, vol. 45, no. 8, pp. 1097-1101, Aug. 1998.

29. Hongmei Liu, Jiufen Liu, Jiwu Huang, Huang, Daren Huang, Yun Q. Shi, "A Robust DWT-Based Blind Data Hiding Algorithm," in *IEEE International Symposiumon Circuits and Systems,* May 2002.

30. S. B. Wicker, *Error Control System for Digital Communication and Storage*, Englewood Cliffs, NJ: Prentice-Hall, Inc, 1995.

31. C. I. Podilchuk and W. Zeng, "Image-adaptive watermarking using visual models," *IEEE J. Select. Areas Commun.*, Vol. 16, No. 4, pp. 525-539, 1998.

32. J. Huang and Y. Q. Shi, "An adaptive image watermarking scheme based on visual masking," *Electronics Letters*, 1998, 34 (8), pp. 748-750.

33. J. Huang, Y. Q. Shi and Y. Shi, "Embedding image watermarks in DC component," *IEEE Transactions on Circuits and Systems: Video Technology*, vol. 10, no. 6, pp. 974-979, September 2000.

34. B. Chen, G. W. Wornell, "Quantization index modulation: a class of provably good methods for digital watermarking and information embedding," *IEEE Transaction on Information Theory*, vol. 47, no. 4, pp. 1423-1443, May 2001.

35. J. G. Proakis, *Digital communication,* 4th edition, McGraw-Hill 2000.

36. G. Voyatzis and I. Pitas, "Chaotic mixing of digital images and applications to watermarking," *Proceedings of European Conference of Multimedia Applications, Services Techniques (ECMAST'96*), 2, pp. 687-695, May 1996.

37. Z. Ni, Y. Q. Shi, N. Ansari, W. Su, Q. Sun, and X. Lin, "Robust lossless image data hiding," *2004 IEEE International Conference on Multimedia and Expo,* Taipei, Taiwan, June 2004.

38. Z. Zhang, G. Qiu, Q. Sun, X. Lin, Z. Ni and Y. Q. Shi, "A unified authentication framework for JPEG2000 images," WG1N2946, JPEG Strasbourg meeting, July 2003; WG1N3107, JPEG Hawaii meeting, December 2003; WG1N3297, CD version 1.0, April 2004 after JPEG Madrid meeting.