UNIVERSITY OF CALIFORNIA
Santa Barbara

Robust Techniques for Hiding Data in Images and Video

A Dissertation submitted in partial satisfaction
of the requirements for the degree of

Doctor of Philosophy

in

Electrical and Computer Engineering

by

Jong Jin Chae

Committee in charge:
Professor B. S. Manjunath, Chairperson
Professor Sanjit K. Mitra
Professor Yuan-Fang Wang
Professor Shiv Chandrasekaran
Doctor Hyun Doo Shin

June 2000

UMI Number: 9997425

UMI®

UMI Microform 9997425

Bel! & Howell Information and Learning Company
300 North Zeeb Road
P.O. Box 1346
Ann Arbor, MI 48106-1346

The dissertation of Jong Jin Chae is approved:

Committee Chairperson

June 1999

March, 2000

To my beloved mother, Ok Ryun Chang,
to my wife, Sook Kyung Han, and to my daugters Hyuna and Junga.

Acknowledgments

I am very grateful and respectful to Processor B. S. Manjunath, chairman of my thesis committee, for his excellent guidance, encouragement and complete support during the course of this research. I am also very thankful to Professor Sanjit. K. Mitra for his research comments and suggestions of my research. I would like to thank Dr. Hyun Doo Shin for discussing me of part of my thesis. I would also like to thank the other members of my doctoral committee: professor Shiv Chandrasekaran and Professor Yuan-Fang Wang for many valuable suggestions and assistance during my study at UCSB.

I thank the current members of the Vision Research and Image Processing Laboratories: Shawn Newsam, Xinding Sun, Barish Sumengen, Peng Wu, Jelena Tesic, Yining Deng, Debargha Merkerjee, Michael Moore, Rajeev Gandhi, Serkan Hatipoglu, Luca Lucchese, and former members: Jay Winkeler, Wei Ma, Are Hjoerungens, Rolf Schoyen, Wenxia, David Garza, Nam Ik Cho, for their friendship and help. They have all made my life and work at UCSB an joyable experience with perfect english tutoring. My research efforts have benefited immensely from stimulating technical discussions, particularity with Debargha Merkerjee, Shawn Newsam, and Michael Moore.

I am very grateful and respectful to Professor Rea -Hong Park for his strong encouragement to study at UCSB and for the many research discussions. I would also like to thank the members of my company for helping and supporting me to study at UCSB.

Finally, I would like to thank my mother, Ok Ryun Chang. When I reflect upon the past, I realize how she provided me with encouragement and support to make me what I am today. And I give my best words to my wife, for her endless and tremendous love. Without their patience, encouragement, and support, this thesis would not have been possible.

Vita

May 2, 1959	Born, Seoul, Korea
Jun. 1981 - November 1983	Military Service, Korea
Winter 1988	Bachelor of Science Department of Electronics Engineering Sogang University, Seoul, Korea
Winter 1991	Master of Science Department of Electronics Engineering Sogang University, Seoul, Korea
Spring 1988 - Winter 1991	Teaching/Research Assistant Department of Electronics Engineering Sogang University, Seoul, Korea
Spring 1988 - 1995	Researcher/ Senior Researcher Institute for Defense Information System Seoul, Korea
Summer 1996 - 1999	Research Assistant Department of Electrical and Computer Engineering University of California, Santa Barbara
July 1999 - present	Project Manager Institute for Defense Information System Seoul, Korea

Publications

- J. J. Chae and B. S. Manjunath, “Data Hiding in Video,” *Proceeding of IEEE, International Conference of Image Processing '99*, Vol. 1, pp. 311-315, Kobe, Japan, October, 1999.

- J. J. Chae and B. S. Manjunath, “A Technique for Image Data Hiding and Reconstruction without Host Image,” *Proceeding of SPIE EI '99, Security and Watermarking of Multimedia Contents*, Vol. 3657, pp. 386-396, San Jose, California, January, 1999.

- J. J. Chae and B. S. Manjunath, “A Robust Embedded Data from Wavelet Coefficients,” *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database VI*, Vol. 3312, pp. 308-317, San Jose, February, 1998.

- J. J Chae, D. Mukherjee and B. S. Manjunath, “A Robust Data Hiding Technique using Multidimensional Lattices,” *Proceedings of the IEEE Forum on Research and Technology Advances in Digital Libraries,* pp. 319-326, Santa Barbara, April 1998.

- J. J. Chae, D. Mukherjee and B. S. Manjunath, “Color Embedding using Lattice Structures,” *Proceeding of IEEE International Conference of the Image Processing,* Vol. 1, pp. 460-464, Chicago, Illinois, October, 1998.

- D. Mukherjee, J. J. Chae and S. K. Mitra, "A Source and Channel Coding Approach to Data Hiding with Application to Hiding Speech in Video," *Proceeding of IEEE International Conference of Image Processing*, Vol. 1, pp. 348-352, Chicago, October, 1998.

- D. Mukherjee, J. J. Chae, B. S. Manjunath and S. K. Mitra, "A Source and Channel Coding Framework for Vector based Data Hiding in Video," *IEEE Transactions on Circuits and Systems for Video Technology*, (accepted for publication) 2000.

- J. J. Chae, S. B. Chae and R. -H Park, "Effective contour coding technique using the 2x2 block," *International Symposium on Information Theory and its Applications,* Vol. 1, pp. 543-546, 1990.

- J. J. Chae and R. -H. Park, "Hierarchical Fingerprint Recognition using the Ridge-line," *Journal of Korea Information Science Society*, Vol. 18, no. 5, pp. 524-533, 1991.

- J. J. Chae, S. B. Chae and R. -H. Park, "Effective contour coding technique using the 2x2 bolck," *Summer Conference, the Korean Institute of Telematics and Electronics,* Vol. 13, no. 1, pp. 433-435, 1990.

- S. B. Chae, J. J. Chae and R. -H. Park, "An Improved texture coding in the segmentation-based image coding," *Summer Conference, the Korean Institute if Telematics and Elctronics,* Vol. 13, no. 1, pp. 442-445, 1990.

Abstract

Robust Techniques for Data Hiding in Images and Video

by

Jong Jin Chae

In this thesis we present a study and development of robust techniques for hiding data in images and video. In recent years, the internet and the world wide web have revolutionalized the way in which digital data is distributed. The widespread and easy access to multimedia content has motivated development of technologies for digital steganography or data hiding. Much of the recent work in data hiding is about copyright protection and authentication of multimedia data. Such digital watermarking typically require very few bits, and the objectives include robustness to attacks on the data. On the other hand, the primary goal of the research presented here is to develop techniques for *hiding* large amounts of multimedia data such as text, images, audio, and video, in images and video. In developing these techniques, emphasis is on robustness to signal compression as it is one of the mostly frequently performed signal processing operation on the data.

The thesis presents new techniques for such data embedding using well known transforms such as the Discrete Cosine Transform (DCT) and the Discrete Wavelet Transform (DWT). We utilize spread spectrum embedding and lattice encoding to hide signature data that is as much as 25% of the host data size. The signature data can be gray scale or color images or video sequences or audio data. The host signals explored include images and video. We demonstrate robustness to JPEG and MPEG compression that include both lossless and lossy hidden data recovery. We present methods that do not need the original host data for signature signal recovery.

The work presented here significantly advances the state of the art in multimedia data hiding, and has the potential of creating a whole new domain of applications including embedded control of multimedia data, smart multimedia objects, and video/audio quality control.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

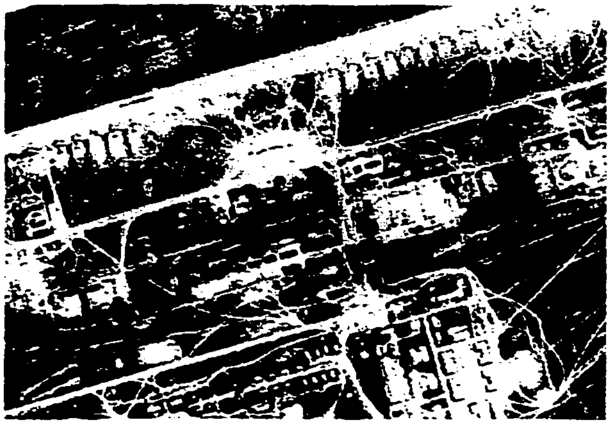
The internet and the world wide web have revolutionized the way in which digital data is distributed. The widespread and easy access to multimedia content has motivated development of technologies for digital steganography or data hiding, with emphasis on access control, authentication, and copyright protection. Steganography deals with information hiding, as opposed to encryption. Much of the recent work in data hiding is about copyright protection of multimedia data. This is also referred to as digital watermarking. While access restrictions can be provided using electronic keys, these do not offer protection against further (illegal) distribution of such data.

Digital watermarking for copyright protection typically requires very few bits, on the order of 1% of the host data size. These watermarks could be alpha-numeric characters, or could be multimedia data as well. The primary objective of watermarking is to be able to identify the rightful owners by authenticating the watermarks. As such, it is desirable that the methods for embedding and extracting digital watermarks be resistant to typical signal processing operations on the host, such as compression, and intentional attacks to remove the watermarks. Signal compression is of special interest as it is perhaps the most frequently performed operation on multimedia data. In particular, lossy compression affects the internal representation of the hidden data. There is a clear need for techniques that are robust to lossy compression, and development of such techniques is the focus of this dissertation.

Our main objective is to develop techniques for *hiding* large amounts of multimedia data, such as text, images, audio and video, in images and video. Hence, the requirements

> Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present [Markus Kuhn 1995-07-03].

(a) A text message [5]

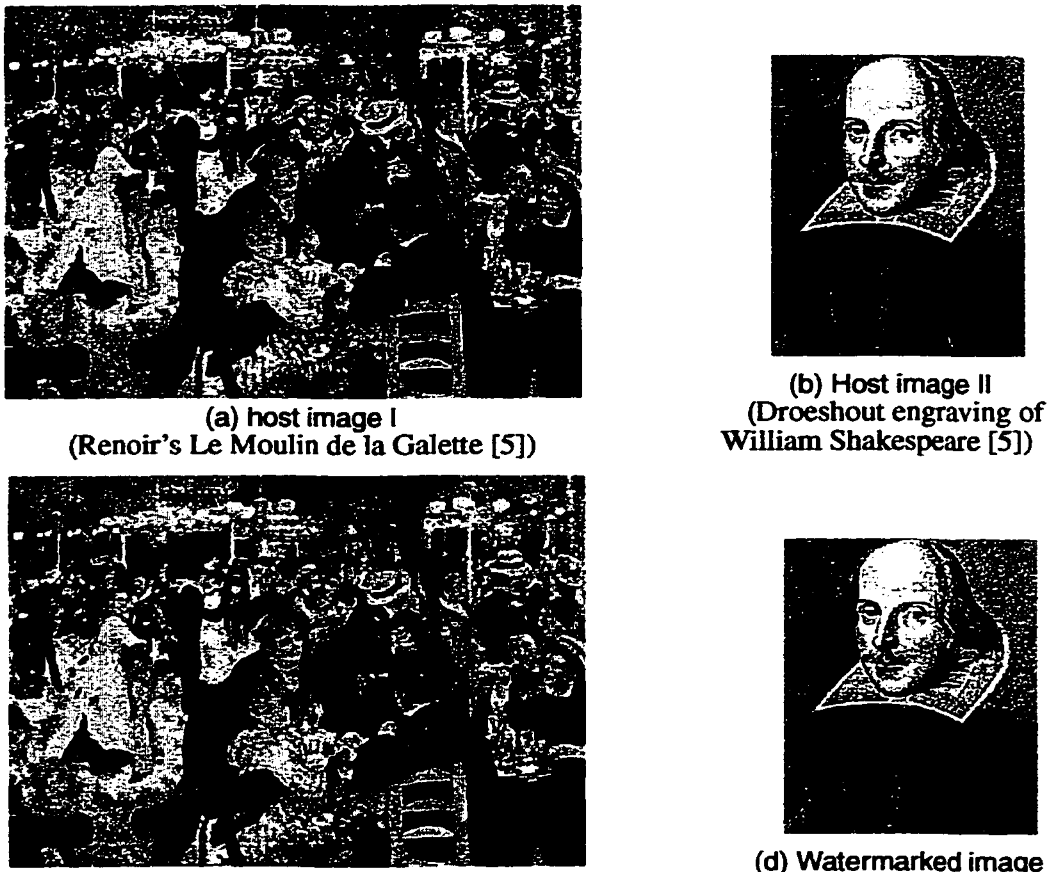


(b) Image example [5]

FIGURE 1-1. Examples of signature data, (a) a text message (1535 bytes), and (b) a satellite image (432x320 pixels, 1 byte per pixel) [5].

are different from typical digital watermarking for data authentication. Enabling applications for such large scale data hiding include embedded control to track the use of a video clip in pay-per-view applications, hidden communications of text (e-mail), voice and visual data, smart images/video that can self-correct under intentional attacks, to mention a few. The capability to hide large amounts of data will also enable robust hiding of digital watermarks by introducing redundancies in the data.

Figure 1-1(a) shows an example of a text message that needs to be sent using a host image. If the hidden data are images or video, as in Figure 1-1(b) which shows a satellite photograph, one can tolerate a moderate amount of loss in reproduction. Since the emphasis



(a) host image I
(Renoir's Le Moulin de la Galette [5])

(b) Host image II
(Droeshout engraving of
William Shakespeare [5])

(c) Watermarked image
(signature: airphoto)

(d) Watermarked image
(signature: text)

FIGURE 1-2. Sample host images (Renoir: 432x320, Shakespeare: 192x240).

is on *hiding*, the original host images should show no obvious distortion when embedded with the messages. Figure 1-2 illustrates some examples of embedding, using the two messages of Figure 1-1 in two host images. As can be seen from the embedded images, it is hard for a casual observer to notice the difference between the original and the embedded hosts.

1.1 Data Hiding vs. Data Encryption

While the data hiding framework is similar to generic encryption, there is one important difference between the two. Although both endeavor to hide a secure signature, the approaches are very different. In encryption or cryptography, the original data is changed to the another format, so that the original data can no longer be determined from the encrypted data [86]. Thus the format of the encrypted data is different from the original data. On the contrary, data hiding strives to conceal and represent the secure data inside a host, in such a manner that the modified host with the secure data inserted remains perceptually indistinguishable from the original host. Here the host data format is unchanged. In contrast to encryption or cryptography, which focuses on rendering messages unintelligible to any unauthorized persons, the heart of data hiding lies in devising methods for concealing messages in a host medium without perceptually altering the host content.

The embedding procedure must be secure in that an unauthorized user must not be able to detect the presence of hidden data, let alone remove or alter it.

1.2 Terminology

Before we continue, we would like to introduce terminology that is used in this thesis. The ***signature or message data*** refers to the secure data that we would like to embed or conceal. This is also referred to as a digital **watermark** in applications related to authentication. In the example of Figure 1-1, the text and the satellite image constitute the signature data. The ***source data*** is used to hide the signature data; we also refer to the source as the ***host***. The procedure of inserting or hiding the signature data in the host is referred to as ***embedding*** or ***watermarking***. Embedding a signature into a host yields the ***watermarked*** or ***embedded data***. The extraction procedure operates on the embedded data and possibly the original host, to yield the ***recovered data***, also referred to as the ***reconstructed data***. If the original host is not available during recovery, the recovered data includes both the recovered signature as well as the recovered host.

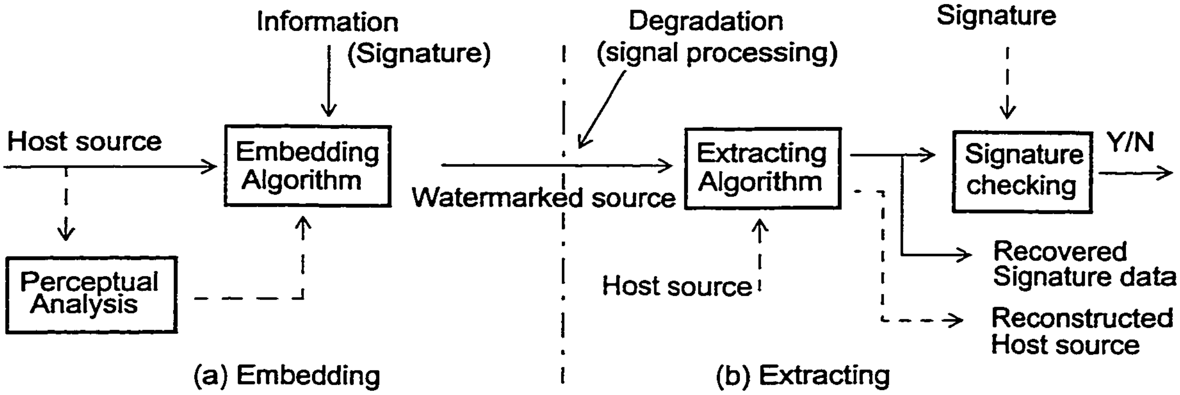


FIGURE 1-3. Schematic of a digital watermarking system.

Figure 1-3 shows a schematic of a typical digital watermarking system. Here we can draw an analogy with a typical digital communications system: **Embedding** is analogous to encoding, and **signature recovery** is similar to decoding [93]. During the embedding process, one can make use of the perceptual characteristics of the human visual system, such that the embedded host data shows very little visible distortions. Similar to noise in a communication channel, the watermarked image might undergo undesirable transformations, such as intentional manipulations to remove or degrade the quality of the watermarking, or typical signal processing operations such as compression that affect the internal representation of the watermark. In most of the recent work on digital watermarking, the original host is assumed to be available during recovery. A more challenging problem, however, is to recover the hidden data without the original host. In this thesis, we propose techniques that can recover the signature data without any knowledge of the original host.

For authentication purposes, the watermarking algorithm must be robust enough to withstand not only degradations brought about by unavoidable signal processing operations, geometric distortions, cropping, A/D conversion, and so on, but also intentional attacks to remove existing watermarks. To ensure this, only a small quantity of hidden data (signature) can be reliably embedded in the host. In the more general data hiding scenario, the quantity of the hidden data could be significantly large and robustness to lossy compression is particularly important.

1.3 Research Objectives

The main requirements for data hiding that are addressed in this dissertation are now summarized.

1. The embedded host should be perceptually indistinguishable from the original host.
2. The embedding techniques should allow for hiding significantly larger amounts of data than that required by the traditional digital watermarking problem.
3. The hidden data must be secured by an encryption key, so that unauthorized retrieval becomes impossible without its knowledge.
4. The algorithms developed should exhibit demonstrable robustness against lossy compression.
5. The methodology should have sufficient flexibility to achieve a wide range of trade-off between the amount of data to be hidden, the level of robustness required, and the amount of perturbation to be tolerated by the host in the process of embedding.
6. The methodology should allow for adaptivity based on perceptual characteristics of the human visual system.
7. The methodology should be sufficiently generic so that both compressible and incompressible sources can be embedded.
8. Authorized retrieval will be considered both with and without the knowledge of the original host.

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

1.4 Main Contributions

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

1.4.1 Functionality

Hiding large amounts of data

Much of the prior work published on watermarking has used signature data which is only a small fraction of the size of the host, with data rates on the order of less than 1%. Typically, signatures include pseudo-random noise sequence or binary images. We propose methods that can embed signature data as much as 25% of the host data size, for applications such as image-in-image hiding or video-in-video hiding. In data hiding, the faithfulness of the embedded data to the original host is important. The proposed algorithms are very resistant to compressions, handling up to 90% JPEG compression. We present several techniques for large signature data hiding in Chapters 3, 4, 5 and 6.

No-host signal recovery

An important issue in watermarking and data hiding is hidden data recovery when the original host is not available during the extraction procedure. The basic idea of the no-host signal recovery algorithm is to split the region of host signal coefficients in a transform domain, such as the discrete cosine transform (DCT). The signature data is then inserted into a specified region of host signal data. A private key can be used to specify the host signal coefficients which are affected by the signature data insertion. We present techniques in Chapters 5 and 6 which enable no-host recovery.

Lossless/lossy recovery

Another equally important issue not well addressed in current literature is lossless recovery of the message or signature data. Many of the existing algorithms can not handle any degradation of the embedded signal (See [5,68]). As mentioned earlier, lossless recovery of hidden data is similar to data encryption. This enables interesting applications including embedded control signals, text messages in multimedia, and smart images that can self-correct. We demonstrate lossless recovery of the message data in Chapter 7.

1.4.2 Robust Embedding Methods

In enabling the above functionality, we have developed new and robust image and video data embedding methods.

An extended spread spectrum technique

We propose an extended spread spectrum technique which introduces redundancy and spreads the signature information in the wavelet domain. This enables robust recovery of the signature even under lossy compression. The proposed scheme's focus is on hiding the signature mostly in the low frequency bands of a wavelet decomposition, and stable reconstruction can be obtained even when the images are highly degraded. An implementation of this method is described in Chapter 3.

Use of Lattice Codes for Error Resilience

We suggest a methodology for embedding data which is coded using multidimensional lattice structures. The use of lattice codes enable resistance to lossy compression. The scheme adopts a vector-based approach to hidden data injection, where the lattice consists of all integer n-tuples with some constraints. Embedding the data amounts to a perturbation of vectors in a high dimensional space. When the embedded data is degraded by compression, it is equivalent to adding noise to the already perturbed coefficients. The true perturbations that represent the signal data are then estimated from the degraded embedded host. We present data embedding methods using lattice codes in Chapters 4, 5, 6 and 7.

Adaptive Embedding

A simple adaptive embedding procedure using texture masking and a user-define quantization matrix of signature data, is suggested. Since the signature data is spread based on the texture properties of the host image blocks, this results in a high quality embedded signal. A texture masking technique considers the properties of the human visual system for better watermarked image quality. The underlying assumption is that the human visual system is less sensitive to changes in highly textured regions as opposed to changes in low frequencies. The signature data is first quantized and lattice codes are assigned to these quantized signature coefficients. These coded coefficients are then embedded in the host signal. We present this method of texture masking with user-defined signature quantization in Chapter 5.

Vector Embedding

A novel method for data hiding in the transform domain is presented in Chapter 6. Many data hiding methods based on the wavelet transform insert data by the merging of wavelet subband coefficients. In contrast, in vector embedding, a new signal for embedding is first constructed using the host and signature data. This signal captures the non-redundancies in the signature in comparison with the selected host data, and as such requires less bandwidth to hide. Consequently, this procedure results in a better signal embedding and reconstruction for a given bandwidth. The technical details of vector embedding is presented in Chapter 6 and some applications are discussed in Chapter 7.

1.5 Organization of the Thesis

This dissertation is organized into eight chapters. Chapter 2 discusses the general requirements on data hiding and digital watermarking. An overview of related research and a brief survey of currently available commercial and public domain software is provided.

Chapter 3 describes an approach to embedding gray scale images using a discrete wavelet transform. The signature wavelet coefficients are distributed in the corresponding subbands of the host. The proposed scheme enables signature images to be as much as 25% of the host image data, and hence could be used both in digital watermarking as well as in image/data hiding. The proposed scheme provides a simple control parameter that can be tailored to either hiding or watermarking purposes, and is robust to operations such as JPEG compression. Preliminary results from this chapter has appeared in [22].

In Chapter 4, we propose a robust data embedding scheme which uses noise resilient channel codes based on a multidimensional lattice structure. A trade-off between the quantity of hidden data and the quality of the watermarked image is achieved by varying the number of quantization levels for the signature and a scale factor for data embedding. Experimental results show that the watermarked image is transparent to embedding large amounts of hidden data, and the quality of the extracted signature is high even when the watermarked image is subjected to up to 75% Wavelet compression [39] and 85% JPEG lossy compression. A private key-based scheme can be used to make unauthorized retrieval practically impossible, even with the knowledge of the reconstruction algorithm. This research has been published in [23,24,83].

Chapter 5 presents a new technique for embedding image data that can be recovered in the absence of the original host image. The data to be embedded, referred to as the signature data, is inserted into the host image in the DCT domain. The signature DCT coefficients are encoded using a lattice coding scheme before embedding. Each block of host DCT coefficients is first checked for its texture content and the signatures codes are appropriately inserted depending on a local texture measure. Experimental results indicate that high quality embedding is possible, with no visible distortions. Signature images can be recovered even when the embedded data is subject to significant lossy JPEG compression. The content of this chapter has appeared in [25,26].

In Chapter 6, a different and novel method for data embedding is presented. There are two main steps in the embedding procedure. The first step creates a new signal, referred to as the β-signal, which is generated based on the host and signature data. Then, the β-signal is embedded into the host coefficients following the method detailed in Chapter 5. This vector embedding technique adds information about the signature signal that cannot be estimated directly from the host, and as such adds robustness to the overall scheme. Lossless recovery of the signature signal is possible as well.

In Chapter 7, we demonstrate lossless signature recovery and video-in-video embedding using the methods described in the previous chapters. We also describe methods for embedding images and video into video which can be recovered following MPEG compression. Chapter 8 concludes with some discussions and future research directions.

Chapter 2

Data Hiding and

Digital Watermarking

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

2.1 Main Issues

Data Authentication

There is an extensive literature on hiding signature data for ownership and authentication [12,14,16,34,31,40,51,58,75,87,90,96,98,103,118,123,125,126]. Primary concerns are protection of intellectual property rights, and checking and tracking content manipulation. Robustness of the watermarking process to signal processing operations performed on the watermarked data is one of the main issues.

Quality of the embedded data

Another important issue in data hiding is the perceptual quality of the embedded data when the signature data is embedded into the host data. It is desirable to minimize the perceived visual distortion of the embedded data, particularly in the case of embedding large amounts of data into the host.

The quality of the reconstructed data depends on various factors. There is an obvious trade-off between the quality and quantity of the signature data that can be inserted into the host without causing significant perceptual degradation. Our proposed algorithms can embed up to 25% of the host data size and can recover the embedded data even under significant JPEG compression.

No-host recovery

If one has access to the original host data, the recovery procedure simply determines, in some sense, the difference between the received and the original data. However, no-host recovery is substantially more challenging. In no-host recovery, the original host data is not available to the receiver. For this reason, the methods that can support no-host recovery have a wider spectrum of applications than just data authentication.

2.2 General Requirements

Several constraints affect the embedding process: the quantity of data to be hidden, the robustness from attacks on the embedded data, and the degree to which the data must be immune to interception or removal by unauthorized users. Obviously, different applications will have different requirements [31,91,113]. In [31], Cox outlines a set of requirements for digital watermarking. Since these are generally applicable to data hiding as well, we give a brief overview of these requirements.

2.2.1 Unobtrusiveness

Digital watermarks should be perceptually invisible, but readable by a computer algorithm. In many applications, such as copyright and usage tracking, embedding metadata or additional information, the algorithms must embed data without affecting the perceptual quality of the underlying host signal. In some applications, though, perceptually detectable watermarks have been used.

2.2.2 Robustness

Digital information is readily manipulated and modified using computers. Operations that damage the embedded host signal may also damage the message data. Furthermore, unauthorized users may attempt to modify the host signal to thwart detection of the embedded data. The following are some of the typical operations:

Common signal processing operations : The watermark should be recovered even if common signal processing operations have been applied to the embedded data. This may include low/high pass filtering, dithering, compression, and A/D and D/A conversion. In this thesis, we focus mainly on robustness to JPEG and MPEG compressions.

Common geometric distortions: The watermark should be extracted following any geometric distortions including rotating, translating, cropping, and scaling of the data. Usually, pixel based embedding techniques are not robust against these kinds of disturbances.

Subterfuge attacks (Collusion and forgery) : The watermark should be protected against collusion by multiple individuals who each posses a watermarked copy of the data. That is, the watermark should be robust to combining copies of the same data set in order to destroy the watermarks.

2.2.3 Universality

The same digital watermarking algorithm should apply to all other media under consideration. Most digital watermarking methods satisfy this requirement. However, data hiding methods which aim at embedding significant amount of data typically make use of the specific properties of the medium.

2.2.4 Unambiguous

The watermark should unambiguously identify the owner. Furthermore, the accuracy of the owner identification should degrade gracefully in the event of an attack.

Much of the recent research on digital watermarking address issues related to copyright authentication and protection. The data used to represent a digital watermark is a very small fraction of the host source. Typical signature data embedded into the host include pseudo-random 1-bit numbers, trade-mark symbols, and binary images [31,53,58,59,87,91]. Multiple watermarks is another important issue in case of data authentication [34,35,36]. Issues of interest in multiple watermarking include identifying the original owner(s) and the progressive degradation of the original watermarks. In this thesis, we will only be concerned with a single data embedding step and will not consider multiple embedding into a given host data.

2.3 Visible vs. Invisible Watermarking

Watermarking of image data could be visible, as in a background transparent signature [85,98], or could be perceptually invisible [34,35,36,53-57,60,87,90,102,118,123,125,126, 127]. A visible watermark acts like a deterrent but may not be acceptable to users in some contexts. In order to be effective, an invisible watermark should be secure, reliable, and resistant to common signal processing operations and intentional attacks. Therefore, most digitally watermarked images are obtained by invisibly hiding signature information into the host image. The signature information is recovered by an appropriate decoding process. The challenge is to simultaneously ensure that the watermarked image be perceptually indistinguishable from the original, and that the signature information be recoverable even when the watermarked image has been compressed or transformed by standard image processing operations.

2.4 Spatial vs. Frequency Domain Embedding

Watermarking methods can be classified into two broad categories. The first class of techniques is based on embedding data in the spatial domain [16,102,115]. Spatial domain methods usually modify the least significant bits of the host image, and are easily affected by signal processing operations. The second class of techniques are based on data injection in a transform domain. The discrete cosine transform [31,60,90,102,118,123] and the discrete wavelet transform [87,125,126,127] are two of the frequently used transformations.

2.4.1 Spatial Domain Methods

Bender *et al.* [16] propose a spatial domain approach to data hiding at varying bit rates. The low-bit encoding emphasizes resistance to unauthorized data removal. Their statistical approach, which is referred to as the *patch work*, is based on a pseudo-random statistical process which embeds one bit per pixel data in a host image. At high bit-rates, such methods tend not to be immune to image modifications. The most common form of high bit-rate encoding is simply replacing the least significant bit (LSB) of the host data with the signature data.

The algorithm proposed by Van Schyndel *et al.* [115] is based on least significant bit manipulation. In this method, the signature data is added to a pseudo-random sequence, making it more difficult to decode, and thus offering inherent security.

Another technique, developed by O'Ruanaidh *et al.* [102], uses a block-mean approach with bi-directional coding. The mean of each block is increased to encode a '1' or decremented to encode a '0'. The number of bits which may be encoded equals the number of blocks. In general, such spatial domain approaches are not robust to simple image processing operations as the embedded information is simply stored in a particular location of the host image data.

2.4.2 Frequency Domain Methods

Most of the recent research on watermarking emphasize the transform domain approach. In general, a digital watermarking in the transform domain is more robust (to signal modifications) than spatial domain methods. The basic idea in these transform domain embedding methods is to perturb the host transform coefficients using the signature information. Since these changes are no longer localized in the spatial domain, the resulting embedded data is more robust to operations such as signal compression. In general, the low frequency components of a signal (such as an image) contain most of the host signal energy. Changes to the host signal's low frequency components may distort the signal or make such changes otherwise quite perceivable. The high frequency components, on the other hand, could be easily removed through signal processing operations.

2.4.3 Discrete Cosine Transform (DCT)

The discrete cosine transform is widely used in signal compression [17,60,90,118,123]. The primary reason being that the DCT is quite effective in signal energy compaction. For a large class of images the energy compaction using DCT is almost as good as using the optimal Karhunen-Loeve transformation [49,67,123]. The current ISO standards JPEG [124] and MPEG [2,3,84,64,65,66] utilize the DCT.

Cox *et al.* [31] propose the spread spectrum coding method for digital watermarking using the DCT transform coefficients for data embedding. The basic idea is to spread small amounts of data across the entire frequency spectrum of the host data. This is discussed in more detail in Section 3.1. Many of the current watermarking techniques are variations of this spread spectrum coding method [27,53,83,90, 95,109].

While the original spread-spectrum method proposed in [31] is based on the DCT of the whole image, block DCT-based embedding methods appear to be quite popular [17,60,61,90,91,118,123,129]. These methods, similar to JPEG coding, typically use 8x8 blocks of pixels. Koch *et al.* [129] propose an embedding algorithm based on the block DCT. A pseudorandom subset of blocks are chosen, and a triplet of midrange frequencies are slightly altered to encode a binary sequence. Such a scheme provides reasonable results on average, although a more image dependent scheme could provide better quality as well as robustness. Huang *et al.* [61] propose an adaptive image watermarking scheme based on visual masking under the DCT domain. After dividing the host into three different block type categories, they embed random noise into the three low frequency components of each block corresponding to the given block types. A similar image-adaptive watermarking scheme using visual models is proposed by Podilchuk *et al.* [91,92]. DCT based video watermarking using perceptual information is proposed by Swanson *et al.* [109-114]. One significant difference in Swanson's approach is that the visual model results in a frequency weighting dependent only on the basis function, and does not adapt to local image characteristics.

Extending these to video, Hartung *et al.* [53-57] propose digital watermarking of raw and compressed video. They used spread spectrum in the DCT domain. The signature is a pseudo-random noise sequence that could be inserted either into an uncoded video stream or into an MPEG bit stream.

2.4.4 Discrete Wavelet Transform (DWT)

During the last decade, multiresolution representation using discrete wavelet transforms has emerged as a strong alternative to the DCT [11,19,62,76,77,100,105,106,117,122]. A given signal is decomposed into successive approximations at different scales using a class of self-similar filters. Efficient implementations of such decompositions now exist and the new JPEG-2000 standard will replace the DCT with the DWT.

Several watermarking algorithms using the DWT have been proposed recently [22,87,125,126,127]. Ohnishi *et al.* [87] propose an embedding algorithm in which binary signature data is inserted in the wavelet transform domain. They use three wavelet coefficients in the high frequency bands and calculate the absolute value of the difference between the maximum and minimum coefficients from the three bands. This absolute value is used for embedding a signature into a host using the Haar wavelet transformation. However, experimental results indicate that this method is not very robust to signal compression.

Xia *et al.* [126] propose a multiresolution watermarking method using a wavelet transform based encoding that is robust to wavelet transform compression and digital halftoning. Since their target application is copyright protection, they used pseudo-random sequences as signature data. In [127], they extend this work to both lossless and lossy compression. After selecting three coefficients of the low-low band in the wavelet decomposed image, the selected coefficients are ordered by magnitude. The dynamic range of these coefficient values is then partitioned into M intervals. The method supports multiple-bit embedding schemes in a large number of intervals.

Wang *et al.* [125] propose WaveMark, a wavelet-based multiresolution digital watermarking system for color images. The algorithm uses an error-corrective coding scheme to provide robust watermarking of digital images. The method does not require the original image for authentication. The wavelet transform of each color band is partitioned into 10x10 blocks, and the inner 8x8 sub-block of each block is used to hide a 64 bit watermark code. The embedding mechanism alters the lower bits of the block borders to code '0' in order to assist the decoding process following the wavelet transform of a 10x10 block.

2.5 Visual Masking

As mentioned earlier, it is often desirable to have the minimum amount of distortion while embedding the signature information [21,25,70,81,90,110,118]. In the case of images, visual masking techniques have been introduced that are adaptive to the local image properties. Swanson *et al.* [110] propose visual masking techniques based on models of the human visual system. They use both spatial and frequency masking models to embed the signature data. For spatial masking, they use a low bit rate image coding model that determines a tolerable error level for each pixel value. For frequency masking, they compute a contrast threshold for each frequency as a function of the frequency, the masking frequency, and the masking contrast. These masking thresholds are used to predict if the changes are perceivable, and thus adaptively embed the signature information.

Similar visual masking techniques have also been explored by other researchers. Piva *et al.* [90] propose a visual masking procedure in the spatial domain to achieve data watermarking. Watermarking is performed by exploiting the masking characteristics of the human visual system, to ensure watermark invisibility. Tao *et al.*[118] propose an algorithm which assigns a noise sensitivity label to each spatial region and embedded data with different labels according to the block DCT. Their six noise sensitivity indices exploit various masking effects of the HVS. Podilchuk *et al.* [92] propose two watermarking techniques based on utilizing visual models which have been developed in the context of image compression.

2.6 Embedding Multimedia Data

While much of the initial work on watermarking was on embedding pseudo-random noise sequences as signature data, some of the recent work address embedding multimedia data, such as image, video, and audio into video sequences [54-56,80,83,95,96,109-114]. Swanson *et. al.* [109] present a scheme for hiding compressed video streams into video frames. They are able to embed up to 2400 bits in a 240x320 video frame using 2 bits per 8x8 block. The data is embedded by linear projection of a pseudo-random sequence. Then, the projected data is quantized and perturbed. However, it is difficult to retrieve the hidden data from the lossy (compressed) watermarked image.

Mukherjee et. al. [83] present an interesting approach to hiding audio in compressed video. Their algorithm, based on multidimensional lattices, works in the DWT domain, typically hiding 8 Khz speech data into QCIF video sequences. The host video is wavelet transformed frame by frame, and vectors of coefficients are perturbed using lattice channel codes to represent hidden vector quantized speech. The embedded video is subjected to H.263 compression before retrieving the hidden speech from it. The retrieved speech is intelligible even with significant compression of the embedded video.

2.7 Signal Recovery without Original Host

The key to any data hiding method is the ability to recover a high quality rendering of the embedded watermark. More specifically, the embedded data may be considered as information transmitted on a communication channel and corrupted by strong interference and channel defects. Bit-wise or noise dependent methods read the watermark without requiring the original host. However, these methods are vulnerable to small changes in the embedded data and thus yield relatively weak watermarks. Many of the prior works in digital watermarking assume that the host source is available [17,34,22,31,53,115].

Other researchers have proposed methods that do not require the original host for hidden data recovery [22-25,53,83,109]. For example, Swanson *et. al* [109] use a perturbation coefficient in the data hiding channel which can be easily extracted using channel coding schemes.

Each 8x8 block of host DCT coefficients is projected onto a pseudo-random direction, quantized, and then perturbed by the signature data. The extraction procedure does not need the original data. The authors demonstrate data hiding for video-in-video. Their scheme can hide about 300 bytes per frame of a 240 x 320 video sequence.

2.8 Other Related Work

Some recent papers have also considered digital watermarking in color images [46,69]. Kutter [69] proposes an amplitude modulation scheme wherein signature bits are embedded by modifying pixel values in the blue channel. The blue channel is chosen as the human visual system is less sensitive to blue than other primary colors. Additionally, changes in regions of high frequency content and high luminance are less perceptible, and thus are favorable locations for data embedding. Robustness is achieved by embedding the signature several times at many different locations in the image. Fleet *et al.* [46] propose an embedding scheme using the S-CIELAB, a well-known standard for measuring color reproduction errors. They embed amplitude-modulated sinusoidal signals into the yellow-blue color band of an opponent-color representation scheme.

Another area where data hiding is useful is in the authentication of printed documents. Inexpensive computers and easy access to high quality printers has resulted in an increase in the forgery of printed documents, including currency. Text document protection is also an important issue [21,70,81]. For copyright protection of electronic text documents, line space encoding, word space encoding, and noise placement encoding are being considered. A detailed review of data hiding for text and printed documents is beyond the scope of this thesis.

2.8.1 Commercial Software

Recently, there have been many commercial software packages for copyright authentication [1,4-9,13,20], some of which could also be used for multimedia data hiding. Johnson *et al.* [41,68] provide a comparative evaluation of several different commercial software. Most of these methods employ variations of least-significant bit encoding for data embedding. This may introduce significant changes in the case of high quality (24-bit) color images. S-Tools [20], a steganography tool, reduces the number of colors while maintaining the image quality, so that the bit changes do not drastically change color values. The resulting image is in a non-standard file format. Jpeg-Jsteg [1] creates a so-called JPEG stego-image using a message to be hidden and a lossless embedded image. The JPEG compression coding scheme is modified for 1-bit steganography in the output files, which are composed of lossy and non-lossy sections. The software combines the message and embedded images using the JPEG algorithm to create lossy JPEG stego-images.

Another shareware program is StegoDos [6]. This program uses the least-significant bit method to hide messages, and is less successful than the other tools mentioned above. The extracted message contains much more data than the original message since the extracting procedure does not identify the end-of-file character for the embedded message. The White Noise Strom [13] uses spread spectrum technology in combination with LSB encoding to hide the signature data.

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

2.9 Summary

Digital watermarking and data hiding has been a very active research area. A majority of the previous research work is related to digital watermarking for copyright authentication. Methods for embedding data both in the spatial and in the frequency domain have been explored. However, most of these existing algorithms do not support large amounts of data hiding. In the following chapters, we introduce several new methods which enable large quantities of data hiding in images and in video. The proposed algorithms are robust to image/video compression, and one can recover the hidden data without requiring the original host.

Chapter 3

Image Hiding in the Wavelet Transform Domain

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

We consider here the problem of hiding images in images. We specifically address robustness to data compression. Lossy compression techniques, such as JPEG, typically affect the high frequency components. This is also true with most perceptual coding techniques based on the human visual system. For these reasons, a digital signature should be placed in perceptually salient regions of the host data. For techniques based on frequency domain modifications, this implies embedding the signature in mostly low frequency components. Inserting signatures in the low frequency components creates problems if one is interested in invisible watermarks. This is particularly true in data hiding applications where the data to be hidden could be a significant percentage of the original data.

We present a data hiding method that allows large scale image to image embedding that is robust to various compression techniques and low-pass filtering. The data is embedded in the wavelet transform domain. We demonstrate that distributing the signature image information in the wavelet transform domain is robust to JPEG [124] and wavelet lossy compression [39]. In the wavelet transform, high-detail image components are projected onto shorter basis functions with high resolution, while the low-detail components are projected onto longer basis functions and lower resolution, thus establishing a trade-off between time and frequency. Typically, for robustness, the signature data is embedded into the low frequency bands in the wavelet transform domain. The proposed scheme focuses on hiding the signature information mostly within the low-frequency DWT bands, and stable reconstruction of the signature image can be obtained even when the images are transformed, quantized (as in JPEG), or otherwise modified by enhancement or low pass filtering operations [42,67].

Another important feature of the proposed method is that it enables large amount of data hiding with little perceptual distortion to the host image. The method can embed a signature that is as much as 25% of the host image size. A possible application of this large data embedding is in secure hidden communications. Note that the requirements for data hiding are quite different from those for watermarking. In data hiding, the faithfulness of the embedded data to the original host is more important than the consequences of unauthorized manipulations of the embedded data, which are not of concern.

In recovering the signature image, it is assumed that the original host image is available. For digital watermarking applications, which typically require embedding a small amount of signature data, further robustness to image processing can be achieved by introducing redundancies in the hidden data. In Chapter 6 we propose another embedding method that does not require the knowledge of the original host in signal reconstruction. In this method, using the given host and the signature data, a new signal for embedding is computed. This computation is based on combining the coefficients from different vector spaces in the wavelet transform domain, and hence the embedding method is referred to as the *vector embedding*, in contrast to the scalar embedding presented in this chapter.

The next section reviews the early work of Cox and his group [31] on spread spectrum based embedding methods. These methods were initially developed for the purpose of authentication and watermarking. Section 3.2 discuss the review and the terminology of wavelet transformation. A wavelet transform based embedding and recovery procedures

(MDWT) are discussed in Section 3.3 and Section 3.4. Experimental results are presented in Section 3.5, and we conclude with discussions in Section 3.6.

3.1 Spread Spectrum Embedding

In the spread spectrum technique for digital communications [93], one transmits a narrow band signal over a much larger bandwidth such that the signal energy present in any single frequency is undetectable. The main idea behind the spread spectrum technique is to spread the signature data over a larger frequency range. Spreading the watermark throughout the spectrum of an image ensures security against unintentional or intentional attacks, and the location of the watermark is not obvious.

Spread spectrum signals used for the transmission of digital information are distinguished by the characteristic that their bandwidth W is much greater than the information rate R (in bits/second) [93]. That is, the bandwidth expansion factor $B_e = W/R$ for a spread spectrum signal is much greater than unity. The large redundancy inherent in spread spectrum signals is required to overcome the severe levels of interference that are encountered in the transmission of digital information over some radio and satellite channels. A second important element employed in spread spectrum communication is selecting the spectrum in pseudo-random manner, which makes the signals appear similar to random noise. Thus, only the intended receivers can decode the signal.

3.1.1 Cox's Work [31]

The spread spectrum method is used for hiding a signal by transmitting it at low power and making it difficult for an unintended receiver to detect the hidden signal in the presence of background noise. Cox *et al.* [31] were among the first to propose a technique for digital watermarking using spread spectrum, and embedded the data in the DCT domain. In their embedding technique the signal energy is spread over many frequency components so that the energy in any one component is small and likely to be undetectable.

Consider a host signal coefficient $f(x)$. A signal coefficients $s(x)$ is then added to $f(x)$ to result in the embedded coefficient

$$f'(x) = f(x)(1 + \alpha \cdot s(x)) \tag{3.1}$$

where $f(x)$ and $s(x)$ are the *x-th* DCT coefficients of the host and signature, respectively. Typically, in watermarking applications, $s(x)$ is a pseudo-random noise sequence where each coefficient is chosen independently according to *N(0,1)*. *N(0,1)* denotes a normal distribution with zero mean and variance 1. The scale factor α determines how densely the signature information is embedded into the host image. In selecting α, there is a clear trade-off between the quality of the watermarked image and the robustness of the watermark to changes in the embedded host. Typically, α is in the range 0.05-1.0 [31].

The spread spectrum technique is quite robust to simple image processing operations, and one can easily detect the similarity between the recovered and original signature image for copyright authentication. Much of the recent work on digital watermarking for data authentication [27,43,55,75,83,90,96,101] is influenced by Cox's initial work.

The recovered watermark is authenticated using standard statistical correlation. Let $s(x)$ be the original watermark and let $s^*(x)$ be the recovered watermark. Then the similarity of $s(x)$ to $s^*(x)$ is defined as [31]

$$S = \frac{\sum_{x} s^*(x)s(x)}{\sum_{x}(s^*(x))^2} \tag{3.2}$$

To match $s^*(x)$ and $s(x)$, one determines the similarity measure S with a threshold value. This is a classical decision-estimation problem which attempts to minimize both the rate of false negatives and false positives.

3.1.2 Other Related Work

Several recent papers [27,31,55,83,90,95,96] further investigate spread spectrum method for authentication purposes. For example, Piva *et al.* [90] present a DCT-based watermarking algorithm using spread spectrum embedding. In their scheme, signature data is embedded in selected host coefficients. The watermark is a M-length pseudo-random sequence $X = \{x_1, x_2, \ldots, x_M\}$. Consider an $N \times N$ block of DCT coefficients from the host image. The $N \times N$ DCT coefficients are reordered into a zig-zag scan, and the first $M + L$ coefficients are selected to generate a vector

$$T = \{t_1, t_2, \ldots, t_L, t_{L+1}, t_{L+2}, \ldots, t_{L+M}\}.$$

The watermarked vector is then computed as

$$T' = \{t_1, t_2, \ldots, t_L, t'_{L+1}, t'_{L+2}, \ldots, t'_{L+M}\}$$

where $t'_{L+i} = t_{L+i} + \alpha \cdot |t_{L+i}| \cdot x_i$, $i = 1, \ldots, M$. In their algorithm, the random noise signature sequence x_i is inserted into the host DCT coefficients.

Hartung *et al.* [55] propose an embedding algorithm using the direct-sequence spread spectrum scheme. They consider binary signature sequence, then spread this discrete signal by a large factor c_r, called the chip-rate, to obtain the spread sequence. Consider the *j*-th watermark coefficient a_j. Then the spread sequence b_i is obtained by $b_i = a_j$, where $j \cdot c_r < i < (j + 1) \cdot c_r$. Let $\{h_i\}$ denote the host signal coefficients. Then the watermarked coefficients are obtained by $\hat{h}_i = h_i + \alpha \cdot b_i \cdot p_i$, where p_i is a pseudo-random noise sequence used for encryption.

Qiao *et al.* [95] propose a watermarking scheme which is similar to that of Hartung [83]. Essentially, their method expands the watermark *n* times. The expanded signature bits are inserted into the *n*-th value of the bit stream of the modified MPEG coded bits, which are then encrypted by the data encryption standard (DES) with KEY [86]. However, the bits are inserted into the MPEG bit stream directly. Some of the more recent work on the use of digital communications technology for data embedding can be found in Chen *et al.* [27], Mukerjee *et al.* [83] and Swanson *et al.* [109].

3.2 Discrete Wavelet Transform: Terminology

A brief overview of the wavelet transform and the terminology used, is now given. We use the notation and terminology from Mallat [76,77], Ogden [117], and Vetterli [122]. Let *Z* denote the set of integers and let *R* denote the real numbers. The square-integrable, one-dimensional function $f(x)$ includes a vector space of measurable $L^2(R)$. For $f(x) \in L^2(R)$ and $g(x) \in L^2(R)$, the inner product of $f(x)$ and $g(x)$ is written as

$$\langle g(x), f(x)\rangle = \int_{-\infty}^{\infty} g(u)f(u)du$$

and in two-dimensions, $f(x, y) \in L^2(R^2)$ and $g(x, y) \in L^2(R^2)$, the inner product of $f(x, y)$ and $g(x, y)$ is written as,

$$\langle g(x, y), f(x, y)\rangle = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} f(x, y)g(x, y)dxdy.$$

3.2.1 Wavelet Transform

Let A_{2^j} be the operator which approximates a signal at a resolution 2^j, $j \in Z$. If $A_{2^j}f(x)$ is the approximation of some function $f(x)$ at the resolution 2^j, then $A_{2^j}f(x)$ is not modified if we approximate it again at 2^j. Thus, A_{2^j} is a projection operator on a vector space $V_{2^j} \subset L^2(R)$. The vector space V_{2^j} can be interpreted as a set of all possible approximations at the resolution 2^j of functions in $L^2(R)$.

Among all the approximated functions at 2^j, $A_{2^j}f(x)$ is the function which is most similar to $f(x)$.

$$\forall g(x) \in V_{2^j}, \quad \|g(x) - f(x)\| \geq \|A_{2^j}f(x) - f(x)\|.$$

Hence, the operator $A_{2^j}f(x)$ is an orthogonal projection on the vector space V_{2^j}.

The approximation of signal at a 2^{j+1} contains all the necessary information to compute the same signal at a smaller resolution 2^j. Further, the approximation $A_{2^j}f(x)$ of a signal $f(x)$ can be characterized by 2^j samples per unit length.

We noted that the approximation operator A_{2^j} is an orthogonal projection on the vector space V_{2^j}. Let $(V_{2^j})_{j \in Z}$ be a multiresolution approximation of $L^2(R)$. Then, there exists a unique function $\phi(x) \in L^2(R)$, called a scaling function, such that if we set $\phi_{2^j}(x) = 2^j \phi(2^j x)$ for $j \in Z$, then $\left(\sqrt{2^{-j}}\phi_{2^j}(x - 2^{-j}n)\right)_{n \in Z}$ is an orthonormal basis of V_{2^j}. A discrete approximation of $f(x)$ can be expressed as,

$$A^d_{2^j}f = \left((f(u) \bullet \phi_{2^j}(-u))(2^{-j}n)\right)_{n \in Z}$$

where ' $\bullet$ ' is the convolution operator. Since $\phi(x)$ is a low-pass filter, this $A^d_{2^j}f$ can be interpreted as a low-pass filtering of $f(x)$ followed by a uniform sampling at the rate 2^j. The scaling function $\phi(x)$ forms a very particular low-pass filter since the family of functions $\left(\sqrt{2^{-j}}\phi_{2^j}(x - 2^{-j}n)\right)_{n \in Z}$ is an orthonormal family.

The difference information between the approximation of a function $f(x)$ at the resolution 2^{j+1} and 2^j is called the detail signal at the resolution 2^j. Let O_{2^j} be the orthogonal complement given by

$$O_{2^j} + V_{2^j} = V_{2^{j+1}} \tag{3.3}$$

The detail signal belongs to the vector space O_{2^j}. To compute the orthogonal projection of a function $f(x)$ on O_{2^j}, we need to find an orthogonal basis of O_{2^j}. The basis can be built by scaling and translating a function $\psi(x)$.

Let $(V_{2^j})_{j \in Z}$ be a multiresolution approximation of $L^2(R)$ and $\phi(x)$ be the corresponding scaling function. The function $\left(\sqrt{2^{-j-1}}\phi_{2^{j+1}}(x - 2^{-j-1}k)\right)_{k \in Z}$ is an orthonormal basis of $V_{2^{j+1}}$. The function $\phi_{2^j}(x-2^{-j}n)$ is a member of V_{2^j} which is included in $V_{2^{j+1}}$.

When computing the inner products of $f(x)$, we have

$$\langle f(u), \phi_{2^j}(u-2^{-j}n)\rangle = \sum_{k=-\infty}^{\infty} \langle \phi_{2^{-1}}(u), \phi(u-(k-2n))\rangle \cdot \langle f(u), \phi_{2^{j+1}}(u-2^{-j-1}k)\rangle .$$

Let H be the discrete filter whose impulse response is given by $h(n) \in \langle \phi_{2^{-1}}(u), \phi(u-n)\rangle$, $\forall n \in Z$. Furthermore, $\tilde{H}$ is the mirror filter with impulse response $\tilde{h}(n) = h(-n)$. Let $H(\omega)$ be the Fourier series defined by

$$H(\omega) = \sum_{n=-\infty}^{\infty} h(n)e^{-in\omega} \tag{3.4}$$

which satisfies two conditions, $|H(0)| = 1$ and $|H(\omega)|^2 + |H(\omega+\pi)|^2 = 1$. Hence, the Fourier transform of a scaling function is defined by

$$\hat{\phi}(\omega) = \prod_{p=1}^{\infty} H(2^{-p}\omega) \tag{3.5}$$

The impulse response of the filter *G* is related to the impulse response of the filter *H* by

$$g(n) = (-1)^{1-n}h(1-n),$$

where *G* is quadrature mirror filter of *H*, and is a high-pass filter.

The Fourier transform of the orthogonal wavelet $\psi(x)$ is given by $\hat{\psi}(x) = G(\omega/2)\hat{\phi}(\omega/2)$ with $G(\omega) = e^{-i\omega}\overline{H(\omega+\pi)}$. Let $\psi_{2^j}(x) = 2^j\psi(2^jx)$ denote the dilation of $\psi(x)$ by 2^j. Then $\left(\sqrt{2^{-j}}\psi_{2^j}(x-2^{-j}n)\right)_{n \in Z}$ is an orthonormal basis of O_{2^j} and $\left(\sqrt{2^{-j}}\psi_{2^j}(x-2^{-j}n)\right)_{(n,j) \in Z}$ is an orthogonal basis of $L^2(R)$.

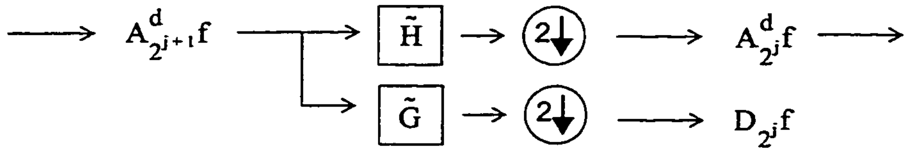


FIGURE 3-1. General tree-structured wavelet filter bank.

Let $P_{O_{2^j}}$ be the orthogonal projection on the vector space O_{2^j}. $P_{O_{2^j}}f(x)$ yields to the detail signal of $f(x)$ at the 2^j. It is characterized by the set of inner products

$$D_{2^j}f = \left(\langle f(u), \psi_{2^j}(u-2^{-j}n)\rangle\right)_{n \in Z}$$

where $D_{2^j}f$ is called the discrete detail signal at 2^j. It contains the difference of information between $A^d_{2^{j-1}}f$ and $A^d_{2^j}f$.

For any $J > 0$, the original discrete signal $A^d_1 f$ measured at a resolution of 1 is represented by

$$\left(A^d_{2^J}f, (D_{2^j}f)_{-J \le j \le -1}\right).$$

This set of discrete signals is called an orthogonal wavelet representation, and consists of the reference signal at a coarse resolution $A^d_{2^{j-1}}f$ and the detail signals at resolution 2^j for $-J \le j \le -1$ (see Figure 3-1). This may be interpreted as a decomposition of the original signal using an orthonormal wavelet basis or as a decomposition of the signal into a set of independent frequency channels. We can compute the detail signal $D_{2^j}f$ by convolving $A^d_{2^j}f$ with the filter $\tilde{G}$ and retaining every other sample of the output. The orthogonal wavelet representation of a discrete signal $A^d_1 f$ can therefore be computed by successively decomposing $A^d_{2^{j+1}}f$ into $A^d_{2^j}f$ and $D_{2^j}f$.

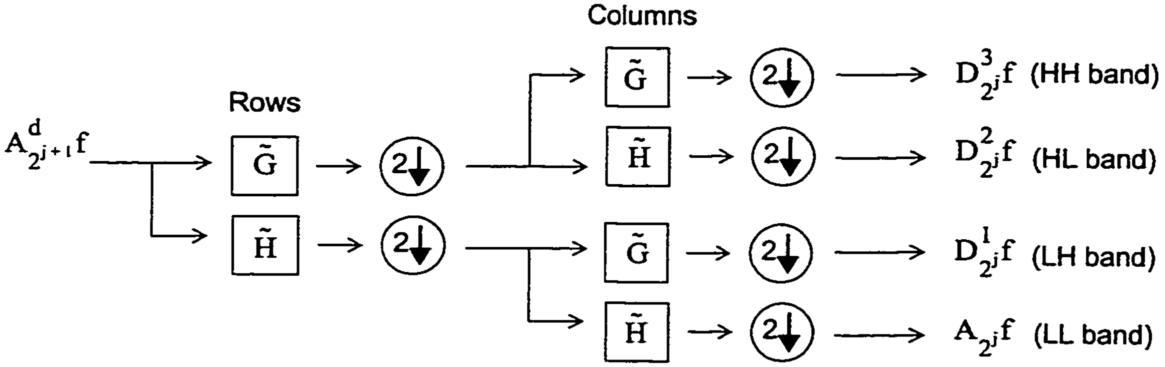


FIGURE 3-2. A schematic of the 2-D wavelet decomposition. Four subbands are obtained: the Low-low (LL), Low-high (LH), High-low (HL), and High-high (HH) pass filters.

3.2.2 2-D Orthogonal Wavelet Transform

In the two-dimensional case, a multiresolution approximation of $L^2(R^2)$ is a sequence of subspaces of $L^2(R^2)$. Let $(V_{2^j})_{j \in Z}$ be a multiresolution approximation of $L^2(R^2)$. One can readily show that the scaling function $\Phi(x, y)$ can be written as $\Phi(x, y) = \phi(x)\phi(y)$, where $\phi(x)$ is the one-dimensional scaling function of the multiresolution approximation $\left(V^1_{2^j}\right)_{j \in Z}$.

The detail signal at resolution 2^j is equal to the orthogonal projection of the signal on the orthogonal complement of V_{2^j} in $V_{2^{j+1}}$ where O_{2^j} is the orthogonal complement. One can build an orthogonal basis of O_{2^j} by scaling and translating three wavelets function, $\Phi^1(x, y)$, $\Phi^2(x, y)$, and $\Phi^3(x, y)$, where $\Phi^1(x, y) = \phi(x)\psi(y)$, $\Phi^2(x, y) = \psi(x)\phi(y)$, and $\Phi^3(x, y) = \psi(x)\psi(y)$.

Figure 3-2 shows a schematic of the two-dimensional wavelet decomposition. The 2-D wavelet transform can be implemented as a sequence of 1-D filtering operation. The resulting decomposition includes the low-low (LL) band corresponding to $A^d_{2^j}f$, the low-high (LH) band $D^1_{2^j}f$, the high-low (HL) band $D^2_{2^j}f$, and the high-high (HH) band $D^3_{2^j}f$.

3.2.3 Haar Wavelet Transform

The Haar wavelet having been developed in 1910 by Haar [117,122]. The Haar function is defined by

$$\psi(x) = \begin{cases} 1, 0 \le x < 1/2 \\ -1, 1/2 \le x < 1 \\ 0, \text{ otherwise} \end{cases} \tag{3.6}$$

which is also called the mother wavelet. For computing a wavelet, we may define a function $H(\omega)$ from (3.4), and compute the corresponding scaling function $\phi(x)$ from (3.5). The scaling function $\phi(x)$ is given by

$$\phi(x) = \begin{cases} 1, 0 \le x < 1 \\ 0, \text{otherwise} \end{cases} \tag{3.7}$$

Let

$$\phi_{j,k}(x) = \sqrt{2^j}\phi(2^j x - k)$$

where $k \in Z$ in each vector space V_{2^j}.

Note that $\phi \in V_0$ and $\phi \in V_1$. Since $\{\phi_{1,k}, \ k \in Z\}$ is a basis for V_1, we can write ϕ in terms of $\phi_{1,k}$ which is defined as

$$\phi(x) = \frac{1}{\sqrt{2}}\phi_{1,0}(x) + \frac{1}{\sqrt{2}}\phi_{1,1}(x), \tag{3.8}$$

and the Haar wavelet is defined by

$$\psi(x) = \frac{1}{\sqrt{2}}\phi_{1,0}(x) - \frac{1}{\sqrt{2}}\phi_{1,1}(x). \tag{3.9}$$

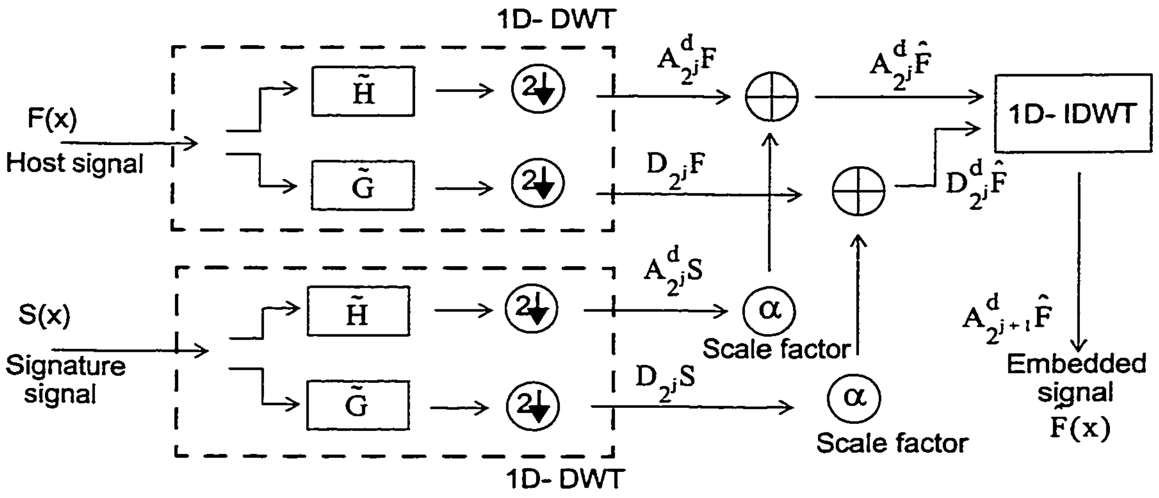


FIGURE 3-3. A schematic of embedding in the DWT domain.

3.3 A Scheme for Merging Wavelet Coefficients

Consider now a host signal $F(x)$ and a signature signal $S(x)$, each of dimension *N*. If *F* and *S* are two-dimensional signals, such as images, one could construct one-dimensional signals by row scanning the images. A simple scheme for merging the two images is illustrated in Figure 3-3. In this scheme, both the host and signature signals are first wavelet decomposed by one level. The host wavelet coefficients (the approximation $A^d_{2^j}F$ and detail $D^d_{2^j}F$ signals) are then added to the corresponding scaled signature coefficients (the approximation $A^d_{2^j}S$ and detail $D^d_{2^j}S$ signals), to obtain the combined signal coefficients. They are represented by

$$A^d_{2^j}\hat{F}(x) = A^d_{2^j}F(x) + \alpha \cdot A^d_{2^j}S(x) \qquad (3.10)$$

$$D_{2^j}\hat{F}(x) = D_{2^j}F(x) + \alpha \cdot D_{2^j}S(x). \qquad (3.11)$$

The combined coefficients are then inverse transformed to obtain the embedded signal $\hat{F}(x)$. From (3.10) and (3.11) the approximation signals of the host and signature data, when they are merged, are in the same vector space V_{2^j} at the resolution 2^j. Similarly, the detail signals of the host and signature data are also located in the same vector spaces. Since the coefficients are added in their respective sub-bands, we refer to this embedding method as the *scalar* embedding in the DWT domain. This method is distinguished from the *vector embedding* method, which will be described later in Chapter 6.

This scheme, while conceptually simple to understand, is not practical. In order to obtain significant robustness to lossy compression while maintaining little perceptual distortion in the embedded host, we need to consider signature data that is only a small fraction of the host data.

3.4 Embedding using DWT

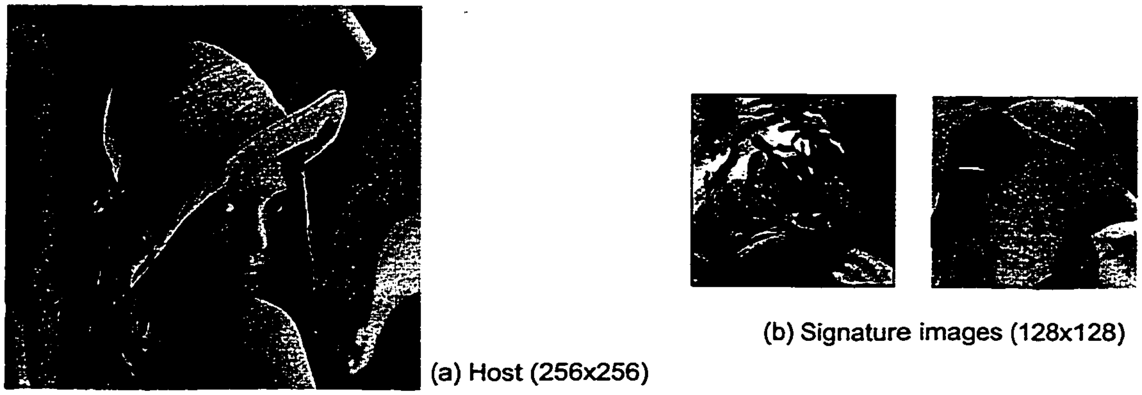
3.4.1 Embedding Procedure: MDWT

An extended spread-spectrum scheme for data embedding in the wavelet transform domain is now presented. The basic idea is to spread the signature data in the host wavelet coefficients. The dominant signature image coefficients are spread over multiple host signal coefficients to ensure robustness to degradation.

3.4.1.1 Merging the coefficients

In the following discussion, it is assumed that the signature image is one quarter the size of the host image, and both images are gray scale with one byte per pixel. A two-dimensional Discrete Haar Wavelet transform is used ((3.8),(3.9)). This results in four sub-bands as shown in Figure 3-2. An example of host and signature images is shown in Figure 3-4. The schematic of the proposed approach is shown in Figure 3-5. It is assumed that the host image is available for signature image recovery. The basic steps in embedding the signature coefficients into the host image coefficients are as follows:

1. Decompose by one level the host and signature images using the DHWT. This results in four bands, LL, LH, HL, and the HH bands (Figure 3-5 (a)).



(a) Host (256x256)

(b) Signature images (128x128)

FIGURE 3-4. (a) A host image and (b) two signature images.

2. Each signature image coefficient is expanded into a 2x2 block as follows.
   - **a.** Each coefficient value is linearly scaled to a 24 bit representation (Figure 3-5 (b)).
   - **b.** Let *A, B, C* represent, respectively, the most significant byte, the middle byte, and the least significant byte in a 24 bit representation (Figure 3-5 (b)).
   - **c.** Three new 24-bit numbers, A', B', C', are generated with their most significant bytes set to *A, B,* and *C,* respectively, and with their two least significant bytes set to zero (Figure 3-5 (c)).
   - **d.** Then, a 2x2 expanded block is formed as shown in Figure 3-5(d).
3. The host image coefficients are also linearly scaled within each band to a 24 bit representation. The minimum and maximum values in each band will be used in the inverse transformation below.
4. The scaled host image coefficients are now added to the expanded signature transformation to form a new fused transform coefficients. Let $f(m, n)$ be the $(m, n)^{th}$ wavelet coefficient of the host image, and let $s(m, n)$ be the $(m, n)^{th}$ signature coefficient after forming the expanded blocks as described in Step 2. Note that after expansion each of the bands in the signature wavelet transformation is of the same dimension as the host image bands. The fused $(m, n)^{th}$ coefficient is then computed as:

$$w(m, n) = \alpha \cdot h(m, n) + s(m, n) \tag{3.12}$$

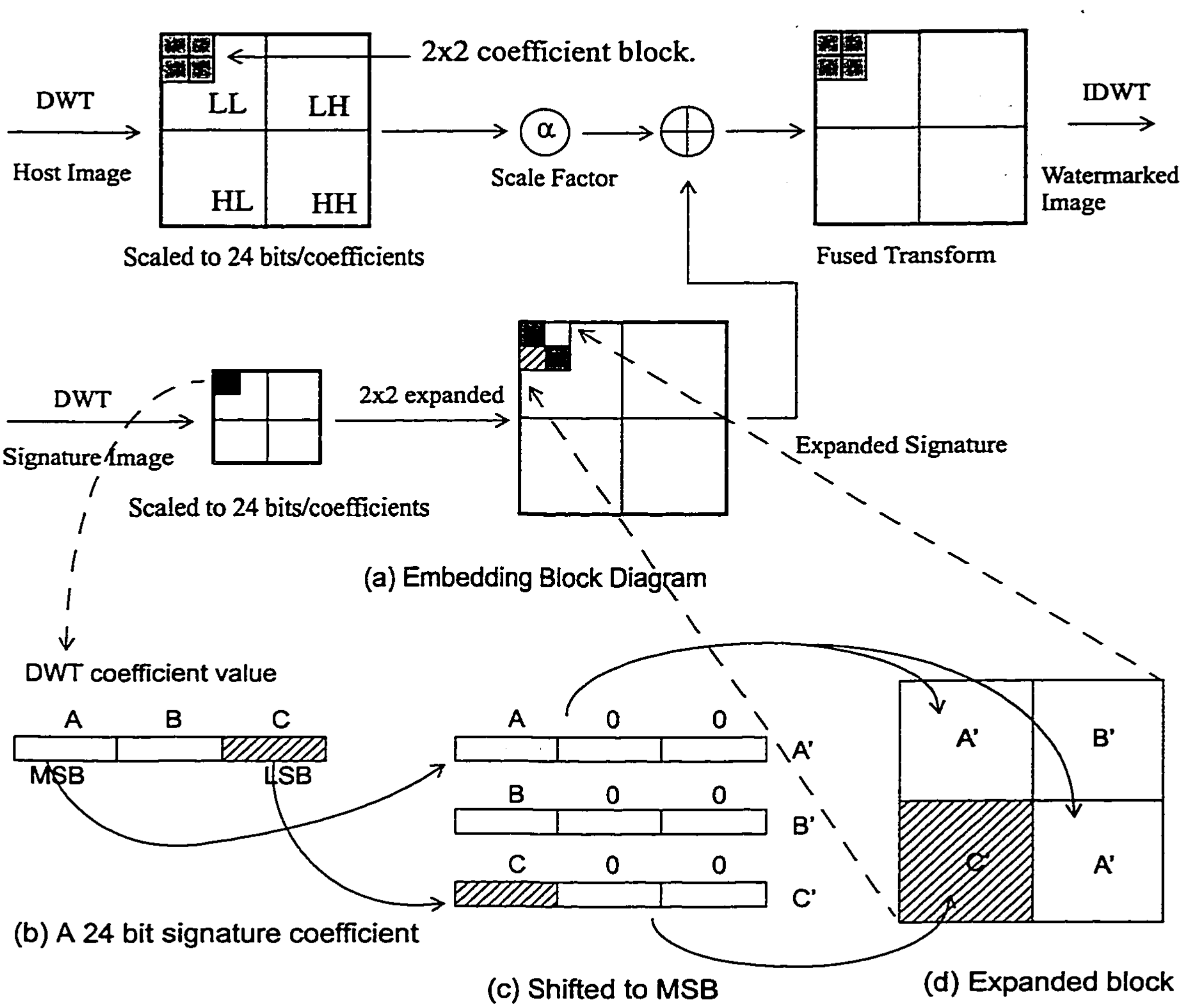
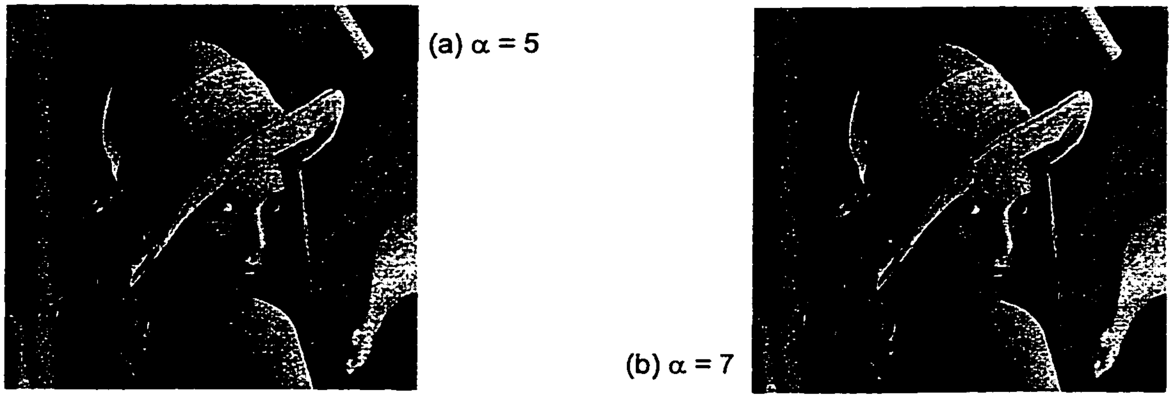


FIGURE 3-5. (a) A schematic of the data embedding approach. (b)-(d) Expanding a single signature coefficient to a 2x2 block of coefficients for embedding in the host image.

where the scale factor α determines the relative percentage of the host and signature image components in the new image. The scale factor α may vary between different subbands. In our implementation, the scale factor for the LH and HL subbands are one-half of the scale factor of LL subband embedding, and that of the HH subband embedding is one-quarter that of the LL subband.

5. The fused transform coefficients in each band are scaled back to the levels of the host



(a) $\alpha = 5$

(b) $\alpha = 7$

FIGURE 3-6. Embedded Lena image at two different scale factors, using the tiger image (see Figure 3-4(b)) as the signature.

image transform coefficients using the minimum and maximum coefficient values in Step 3.

6. An inverse transformation is now computed to give the watermarked image.

Examples of watermarked images for two different values of α alpha are shown in Figure 3-6. Note that smaller values of α result in more perceivable distortions in the embedded image.

3.4.2 Signature Recovery

3.4.2.1 Extracting Procedure

Figure 3-7 shows a schematic of the extraction procedure, which is essentially an inverse sequence of operations in the embedding method. The signature coefficients are separated from the watermarked DWT values by subtracting out the original host coefficients. It is assumed that the original host image data is available.

3.4.2.2 Normalized Similarity Measure

The embedded signature data can be used for authentication. In this case the perceptual quality of the reconstructed image is not the main issue. The similarity measure defined in (3.2) for binary data can be extended to include gray scale images as well. Since the signature

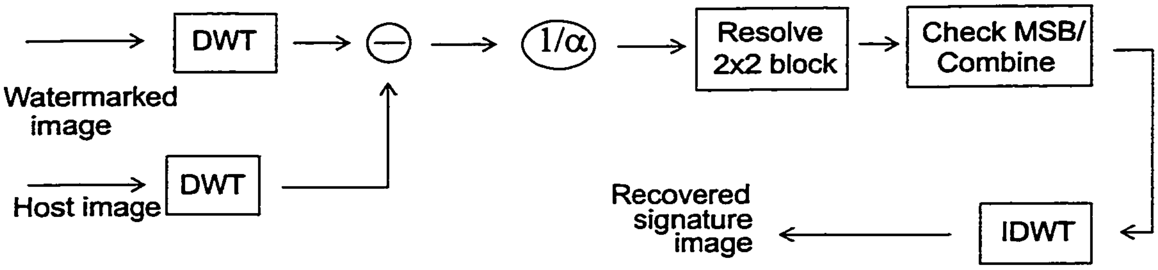


FIGURE 3-7. A schematic of the extracting procedure

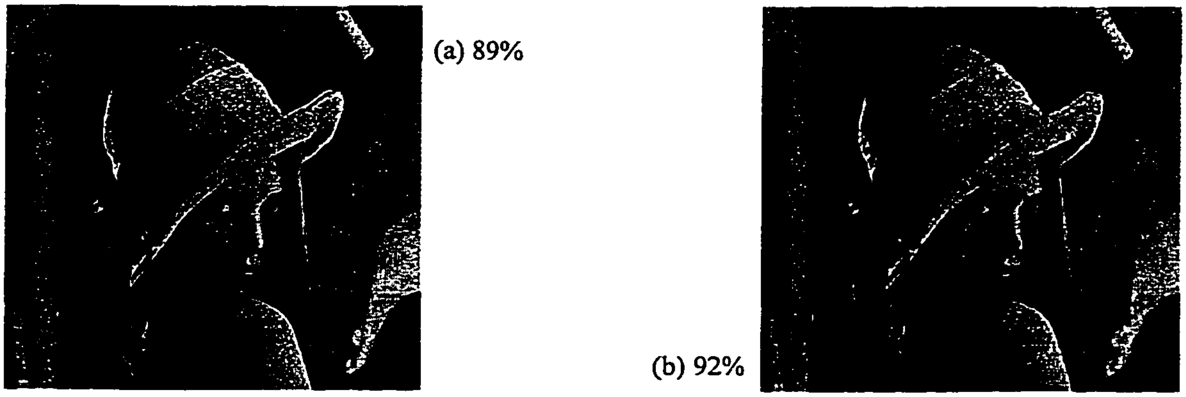
image is assumed to be known in checking for authenticity, we consider cross-correlation between the original signature $s(m, n)$ and the received signature $s^*(m, n)$ from (3.12). The normalized similarity is defined as follows

$$S = \frac{\sum_{m,n} s^*(m,n)s(m,n)}{\sum_{m,n}(s^*(m,n))^2 \sum_{m,n}(s(m,n))^2} \qquad (3.13)$$

3.5 Experimental Results

We present here results of embedding 128x128 gray-scale (one byte per pixel) signature images (tiger and hat-girl) in a 256x256 Lena image. Figure 3-4 shows the host and signature images. Figure 3-6 shows the embedded Lena images using different scale factors. Note that the higher the scale factor, the better the quality of the embedded image (i.e., less distortion due to embedding). Even if the signature image has much texture information like the tiger picture, the embedded image cannot be visually distinguished from the original host image. Figure 3-8 shows the embedded Lena image at various levels of JPEG compression for a scale factor of $\alpha = 7$.

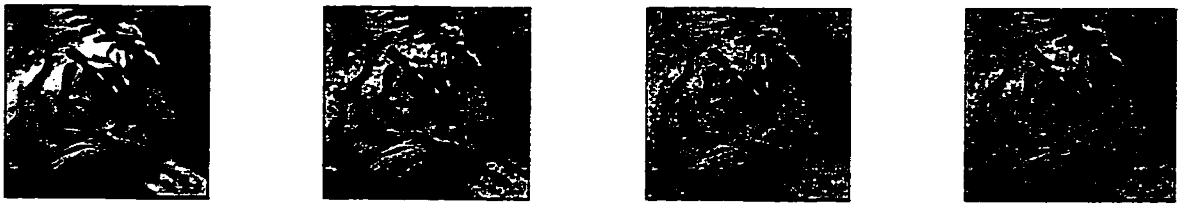
For data hiding purposes it is reasonable to choose a larger scale factor in (3.12) as we are not too concerned with degradation due to image processing operations. In this case, it is more important to ensure that the quality of the watermarked image is as close to the original as possible, with very little visual distortion. Almost perfect reconstruction is possible when



(a) 89%

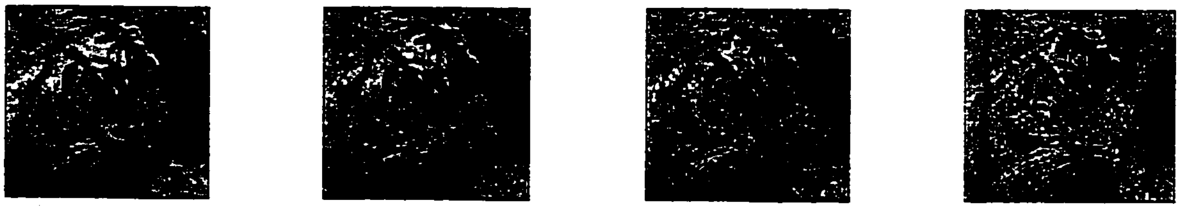
(b) 92%

FIGURE 3-8. JPEG compressed embedded Lena image. The tiger signature image is used in this experiment with scale factor of α = 7.



(a) 49% (b) 79% (c) 89% (d) 92%

FIGURE 3-9. Recovered signature images at different compression rates for scale factor α = 5. The JPEG compression factors used for the embedded image are indicated below each image.



(a) α = 3 (b) α = 5 (c) α = 9 (d) α = 11

FIGURE 3-10. Recovered tiger signature images from 92% JPEG (lossy) compressed embedded images for different scale factors.

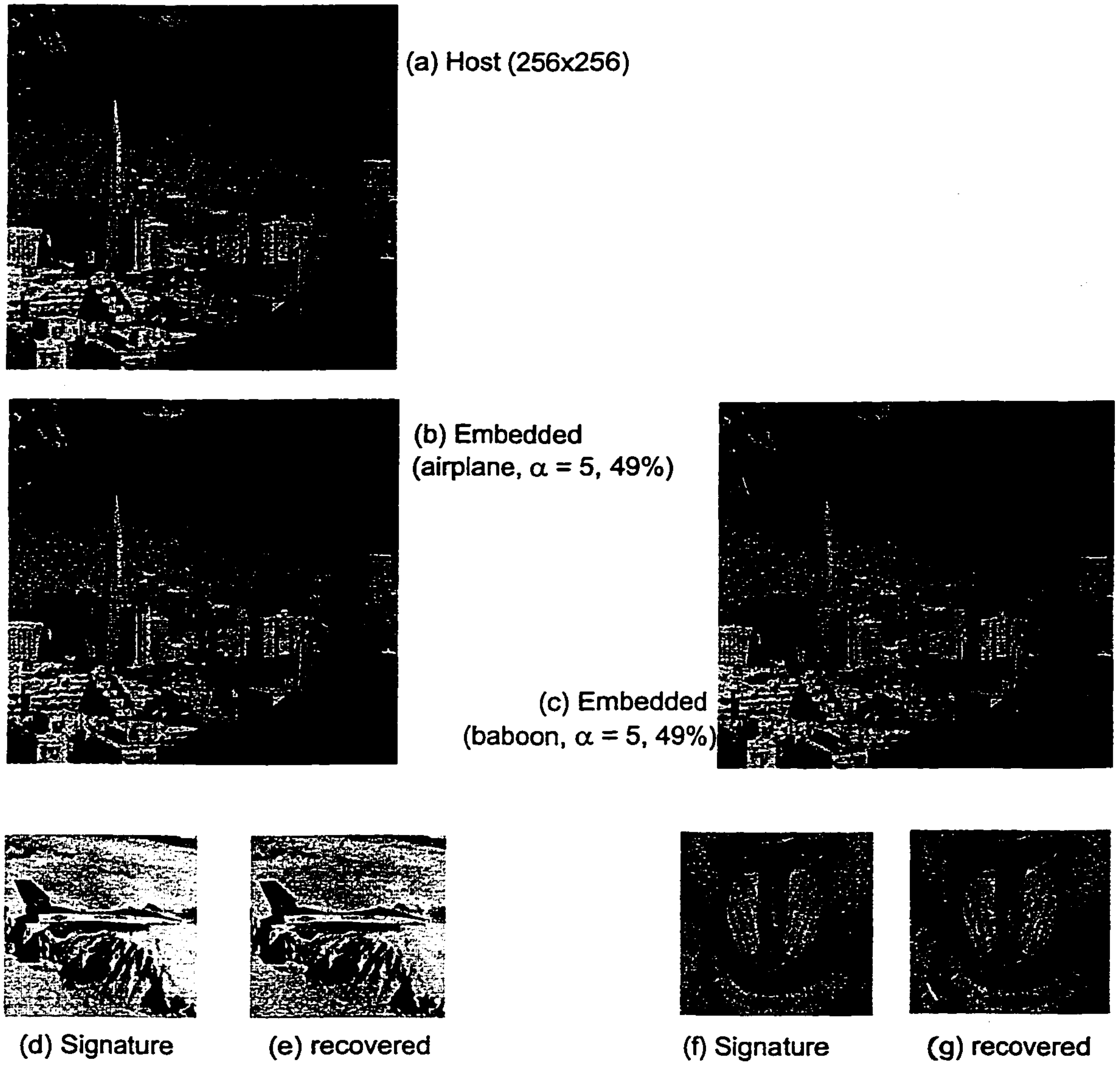
there is no further image processing of the watermarked images. This can be seen from the reconstructed tiger images (at low JPEG compression levels) in Figure 3-9.

For copyright authentication purposes it is important that the watermarked images are robust to typical image processing operations. In such cases it is reasonable to assume that the signatures require significantly fewer bytes than the host image and as such can be spatially distributed. The results we show here are for lossy JPEG compression where the signatures are gray scale images. It is reasonable to expect that one can obtain much better results if the signatures are binary images or pseudo-random numbers, as is typically done in digital watermarking. Lower values for the scale factor in (3.12) should be used when it is likely that the image will undergo significant distortion. Figure 3-10 shows recovered signatures for JPEG compression of 92% for varying scale factors. As expected, images embedded with a larger scale factor result in poor reconstruction for the same compression factor. Figure 3-11 shows another example wherein a different host image is used to embed the signatures.

Figure 3-12 shows the similarity measures computed with variable JPEG compression. One can set a fairly high threshold value, of about 0.85, to detect the presence or absence of an embedded watermark. Figure 3-13 shows the PSNRs of recovered signature images under JPEG compression. Almost perfect reconstruction is possible when there is no further image processing of the watermarked images.

3.6 Discussions and Summary

A robust scheme for image hiding using an extended spread spectrum technique in the wavelet transform domain is presented. This approach could be used for both digital watermarking related applications as well as for data hiding purposes. The scale factor controls the relative amount of host and signature image data in the embedded image. A larger scale factor can be used for data hiding in situations when it is desirable to maintain a high perceptual quality of the embedded image. A lower scale factor is better suited for watermarking where robustness under typical image processing operations is needed. Experimental results demonstrate that good quality signature recovery and authentication is possible when the images are



(a) Host (256x256)

(b) Embedded (airplane, $\alpha = 5$, 49%)

(c) Embedded (baboon, $\alpha = 5$, 49%)

(d) Signature

(e) recovered

(f) Signature

(g) recovered

FIGURE 3-11. Another example of data embedding.

quantized and JPEG compressed by as much as 90%. Even if the PSNR quality of the recovered signature image is of a low value, the recovered signature image is still perceivable.

Even though the Haar wavelet basis is used in these experiments, this method can be easily adopted to other orthogonal wavelet transformations and for more than one level of decomposition. Other basis functions may be worth exploring depending on the characteris-

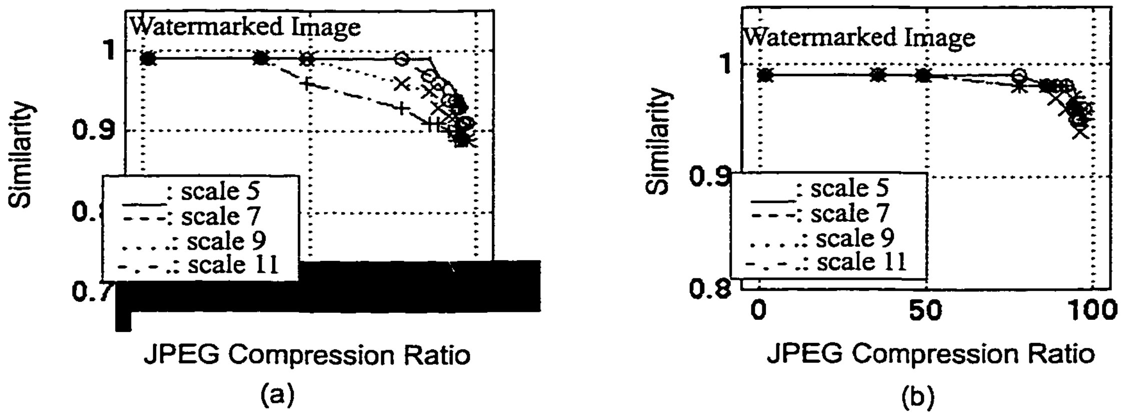
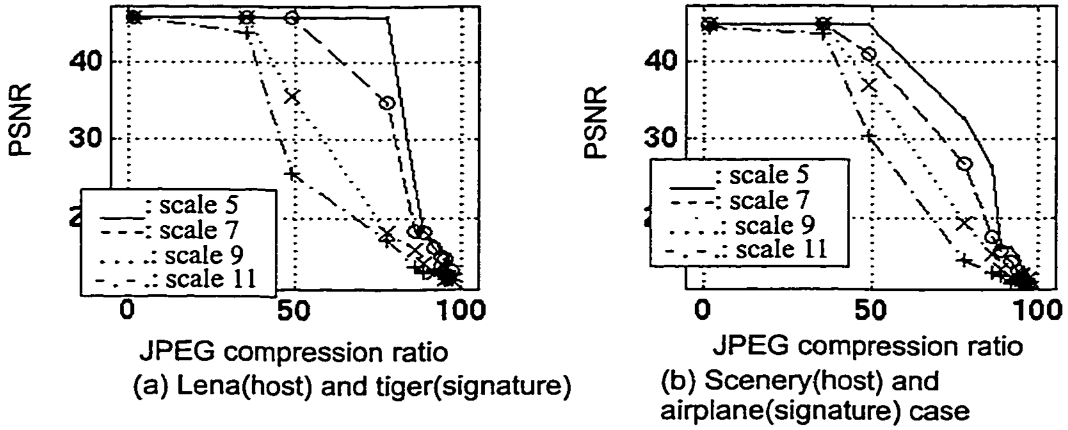


FIGURE 3-12. Similarity versus JPEG compression ratio using (a) Lena (host) and tiger (signature), and (b) pyramid (host image) and airplane (signature image).



(a) Lena(host) and tiger(signature)

(b) Scenery(host) and airplane(signature) case

FIGURE 3-13. The PSNRs of the extracted signature image

tics of the host and signature images. In some cases, particularly when the host image background lacks texture and the signature image has considerable amount of texture, one can see a *noisy* background in the embedded image.

In digital watermarking, the signatures are usually of a much smaller dimension compared to the host image. Since the proposed method can manage a significantly larger amount of signature data, it is possible to distribute the signature spatially as well, thus making watermarking robust for operations such as image cropping.

Chapter 4

Data Hiding using Lattice Codes

This work generalizes the MDWT method for data hiding presented in the previous chapter by introducing channel coding techniques into the embedding procedure. The new method, referred to as the MLDWT method, uses noise resilient channel codes derived from lattice structures. Similar to communication channel noise, the watermarked image might undergo undesirable transformations, such as intentional manipulations to remove or degrade the quality of the watermarking, or signal compression that may affect the watermark. The lattice coding of the signature data helps in further improving the robustness of the watermarked image to lossy compression [28,29,30,48].

If the original host image is available, the operations of data injection and retrieval are, in fact, very similar to the channel coding and decoding operations in a typical digital communication system [93]. Channel coding refers to the gamut of signal processing operations performed before the transmission of data over a noisy channel. For watermarking in the transform domain, the original host data is transformed, and the transformed coefficients are perturbed by a small amount in one of several possible ways in order to represent the signature data. When the watermarked image is compressed or modified by image processing operations, it is equivalent to adding noise to the already perturbed coefficients. The retrieval operation subtracts the received coefficients from the original ones to obtain the noisy perturbations. The true perturbations that represent the injected data are then estimated from this data as best as possible.

The proposed method, MLDWT, adopts a vector-based approach to hidden data injection [28, 29,30,48]. We group N transformation coefficients to form an N-dimensional vector, and modify it by codes that represent the data to be embedded. The motivation for using vector perturbations as opposed to scalar perturbations follows from the realization that higher dimensional constellations usually result in a lower probability of error for the same rate of data injection and the same noise statistics.

It is well known that embedding in the low-frequency bands is more robust to manipulations such as enhancement and image compression. However, changes made to the low frequency components may result in visible artifacts. Embedding in the N-dimensional space allows a low level of signal data injection while maintaining robustness to compression.

The methodology using lattice codes is described in the next section. Embedding using lattice codes is presented in Section 4.2, and signature extraction is described in Section 4.3. Section 4.4 shows experimental results using gray scale and color images. We conclude with discussions and summary in Section 4.5.

4.1 Multidimensional Lattice Codes

Multidimensional lattices are frequently used in vector quantization [28,29,30,48]. The Voronoi regions of various n-dimensional lattices can be used to construct n-dimensional quantizer cells for uniformly distributed inputs [93]. The Voronoi region around any lattice point is the set of points in $\Re^n$ closest to the lattice point. Therefore, the Voronoi region V(0) around the origin is given as:

$$V(0) = \{\mathbf{x} \in \Re^n | \|\mathbf{x}\| \le \|\mathbf{x} - \mathbf{u}\| \text{ (for all nonzero } \mathbf{u} \in \Lambda)\} \tag{4.1}$$

It has been shown by Conway and Sloane [28,30] that some of these lattices produce very good channel codes, and yield high values of nominal coding gain. That is, for the same power constraint on the channel, the channel codes are maximally separated from each other so that they are most robust to noise.

4.1.1 Description of Lattices

If $a_1, ..., a_n$ are *n* linearly independent integer vectors in an m-dimensional Euclidean space with $m \ge n$, then the set of all vectors

$$\mathbf{x} = u_1 a_1 + \ldots + u_n a_n \tag{4.2}$$

where $u_1,\ldots,u_n$ are arbitrary integers, constitute an n-dimensional *root lattice* Λ_n [28,29,30]. A lattice *coset* of Λ_n is obtained from a lattice Λ_n by adding a fixed translation vector $\boldsymbol{t}$ to the points of the lattice,

$$\mathbf{x}^* = u_1 a_1 + \ldots + u_n a_n + \mathbf{t}. \tag{4.3}$$

The *norm* of a vector $\mathbf{x}$ is defined as

$$N(x) = \mathbf{x}\cdot\mathbf{x} = (x, x) = \sum x_i^2. \tag{4.4}$$

The minimal norm of Λ is simply the minimal squared distance between distinct lattice vectors,

$$\min\{N(x-y): x,y \in \Lambda, x \neq y'\} = \min\{N(x): x \in \Lambda, x \neq 0\} \tag{4.5}$$

For any lattice Λ, let N_m be the number of vectors $x \in \Lambda$ of norm m, i.e. with $x \cdot x = m$. So the number of ways of writing m as a sum of lattices squares is equal to the number of vectors of norm m in the lattice. Further, if Λ is a lattice in $\Re^n$, the dual lattice Λ^* consists of all points $\mathbf{x}$ in the span of Λ such that $\mathbf{x}\cdot\mathbf{y} \in Z$ for all $\mathbf{y} \in \Lambda$. Some common lattices and definitions are given below.

- The n-dimensional lattice Z^n is the set of integers $Z^n = \{(x_1, \ldots, x_n): x_i \in Z\}$. For example, Z^2 is a square lattice.
- For $n \geq 1$, A_n is the n-dimensional lattice consisting of the points $(x_0, x_1, \ldots, x_n)$ in Z^{n+1} with $\sum x_i = 0$ in R^{n+1}, which uses $n+1$ coordinates to define an n-dimensional lattice.
- For $n \geq 3$, D_n consists of the points $(x_1, x_2, \ldots, x_n)$ in Z^n with $\sum x_i$ even. In other words, if we color the integer lattice points alternately red and blue in a checkerboard coloring, D_n consists of the red points. In 4 dimensions the D_4 lattice is known to yield the best coding gain.

The even coordinate system of E_8 consists of the points $\{(x_1, x_2, \ldots, x_n): \text{all } x_i \in Z$ or all $x_i \in Z + 1/2$, $\sum x_i \equiv 0$ (modulo 2)$\}$. The odd coordinate system is obtained by changing the sign of any coordinate with $\sum x_i \equiv 2x_8$ (modulo 2).

E_6 is a subspace of dimension 6 in E_8, consisting of the points $(u_0, u_1, \ldots, u_n) \in E_8$ with $E_6 = \{x \in E_8 : x_1 + x_8 = x_1 + \ldots + x_6 = 0\}$ in an even coordinate system.

The lattice codes of D_4, E_6 and E_8 have certain desirable properties, such as giving the minimum mean squared quantization error, making their choice attractive for coding applications.

4.1.2 Lattice Code Assignments

The squared norm is the distance of a lattice point to the all-zero origin vector. One can organize the lattice points in shells such that points on a given shell are all equidistant from the origin. Shell 1, for example, refers to those points that are closest to the origin. In Shell 2, the squared norm is 4 and consist of points that are at a distance of 2 units from the origin. Table 4-1 lists the shell numbers and the corresponding distances for some of the commonly used lattices.

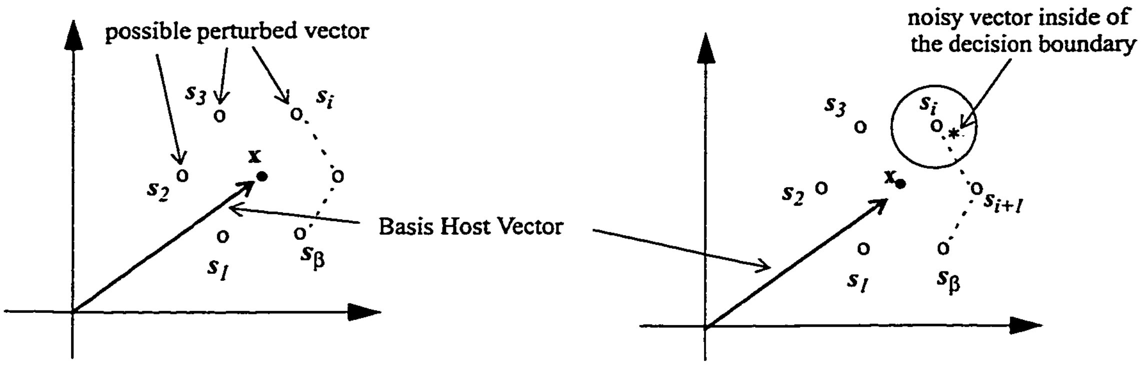
Consider, for example, the D_4 lattice. It consists of the lattice points $(x_1, \ldots, x_4)$ having integer coordinates with an even sum. As in all lattices, the lattice points of the D_4 lattice fall on concentric shells of increasing distance from the all zero vector. For example, the 24 lattice points given by all permutations of $(\pm 1, \pm 1, 0, 0)$ lie on the first shell of the lattice at a distance $\sqrt{2}$ from the center. The second shell at distance 2 from the center contains 24 lattice points again, 8 of which are of type $(\pm 2, 0, 0, 0)$, and 16 are of type $(\pm 1, \pm 1, \pm 1, \pm 1)$. Table 4-1 shows the shell number, the squared norm, the number of lattice points, and the lattice point types for the first few shells of the D_4 lattice. The superscript 'p' in the last column of the table denotes *all permutations of* the elements in the parenthesis.

4.2 Embedding Using Lattice Codes

The basic steps in embedding using lattice codes are quite simple. The host and signal data are first transformed using DCT or DWT. The transformed host coefficients are grouped to form N-dimensional vectors, where N depends on the type of lattice quantizer used. Let 'x' represent a host vector in an N-dimensional space, as shown in Figure 4-1(a). The signal coefficients are quantized and encoded into one of β–symbols, $\{s_1, s_2, \ldots, s_\beta\}$. To embed data

TABLE 4-1. Code types and structure of the multi-dimensional lattices.

Shell No.	Squared Norm	Number of codes				Examples of source codes in D_4
		D_4	E_6	E_8	Λ_{16}	
1	2	24	72	240	0	$(\pm1,\pm1,0,0)^p$
2	4	24	270	2160	4320	$(\pm2,0,0,0)^p$, $(\pm1,\pm1,\pm1,\pm1)^p$
3	6	96	720	6720	61440	$(\pm2,\pm1,\pm1,0)^p$
4	8	24	936	17520	522720	$(\pm2,\pm2,0,0)^p$
5	10	144	2160	30240	2211840	$(\pm2,\pm2,\pm1,\pm1)^p$, $(\pm3,\pm1,0,0)^p$



(a) Possible perturbations

(b) Estimation from received noisy vector

FIGURE 4-1. Basic embedding methodology. (a) Possible β-ary perturbations of the host vector. (b) Possible noisy vector positions of original perturbed vector s_i after transformation. (all points shown above are in an n-dimensional space)

from an β-ary source, we perturb the original vector so that the perturbation coincides with one of β corresponding channel codes. The perturbed vector is denoted by one of the lattice point 'o's in the Figure 4-1(a), depending on the particular source symbol it represents.

At the receiver, let us assume that a perturbed vector '*' is received (see Figure 4-1(b)). The received '*' may not coincide with one of the β–symbols due to changes in the embed-

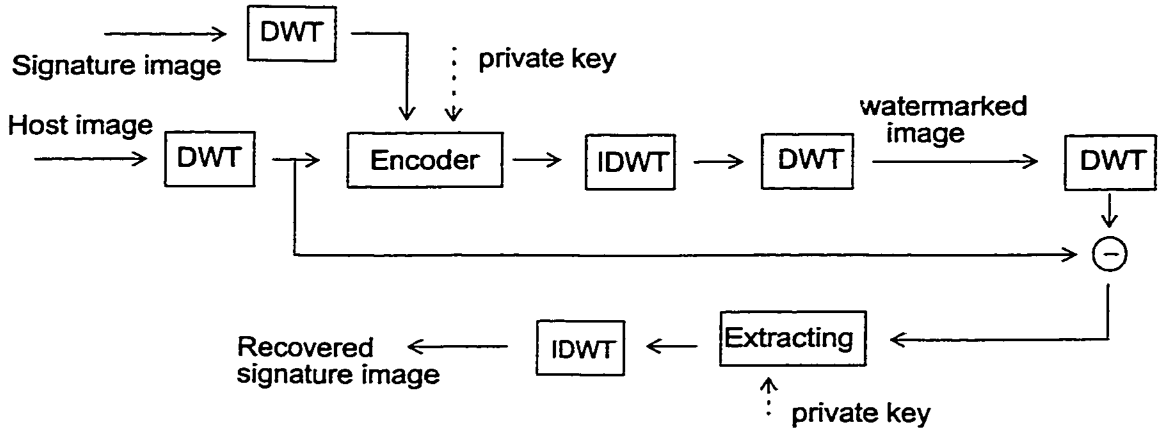


FIGURE 4-2. A schematic of the MLDWT.

ded data. It is then a simple estimation problem to compute the transmitted symbol given the noisy observation '*'. Assuming an additive Gaussian noise model, the received vector is decoded as representing the symbol whose channel code it is closest to in the Euclidean space.

The coefficient vectors perturbed in our implementations are typically multidimensional (4-D, 6-D, and 8-D), and the channel code used to embed the data is a subset of one of the D_4, E_6, or E_8 lattices. As the quantity of embedded data increases, higher order shells of the embedding lattice are included in the channel code to accommodate them. Figure 4-2 shows a schematic of our watermarking procedure.

The number of quantization levels for the signature image data and a scale factor α control the quality and the quantity of the embedded data. A larger scale factor α means better signature reconstruction at the expense of the quality of the embedded data. The number of quantized levels β, on the other hand, determines the coarseness of quantization and therefore the quality of the signature image hidden in the host.

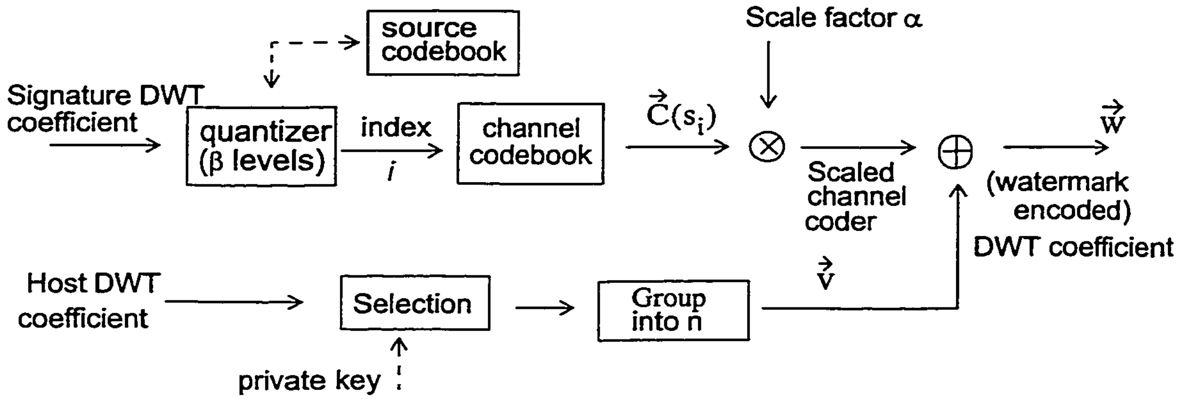


FIGURE 4-3. Embedding procedure

4.2.1 Embedding Procedure: MLDWT

A single level of the discrete wavelet transform decomposition of both the host and the signature image is made before data embedding. The encoder block of Figure 4-2 is further expanded in Figure 4-3. The signature DWT coefficients are quantized using a lattice source codebook and the quantized index is then encoded by the lattice channel codebook. The embedded DWT coefficients merge the basis host vector and the scaled channel coder.

Each coefficient of the signature image is quantized into β levels. In order to embed the quantized coefficient information, a set of *n* coefficients in the host image is grouped to form an n-dimensional vector. For example, n = 4 for the D_4 lattice coding. This *n*-dimensional vector then perturbed according to a β-ary channel code consisting of a subset of an n-dimensional lattice scaled by a factor α. If $\vec{v}$ represents a vector of host DWT coefficients after grouping, and the index of the quantized signature coefficient is *i*, then the perturbed vector $\vec{w}$ is given by:

$$\vec{w} = \vec{v} + \alpha \cdot \vec{C}(s_i) \qquad (4.6)$$

where $\vec{C}(s_i)$ represents the channel code (subset of the n-dimensional lattice) corresponding to the symbol s_i, where $i = 1, \ldots, \beta$.

TABLE 4-2. Example of code assignments for the D_4 lattice.

Quantizer Levels b	Lattice points in channel code
2	(0, 0, 1, 1), (0, 0, -1, -1)
24	Shell_1
32	Shell_1, $(\pm 2,0,0,0)^p$
48	Shell_1, Shell_2
144	Shell_1, Shell_2, Shell_3
168	Shell_1, Shell_2, Shell_3, Shell_4

Each subband of the signature image is embedded into the corresponding subband of the host. As in Chapter 3, we consider a data hiding rate of 25%. Thus, each coefficient in the LL band of the signature image is hidden in four coefficients in the LL band of the host, and so on. The scale factor chosen for embedding in the higher bands is usually less than the scale factor chosen for the LL band.

4.2.2 Signature Quantization

The choice of channel codes used depends on the signature quantization. Recall that the D lattice consists of all integer n-tuples with an even sum. As the quantity of embedded data increases, higher order shells of the lattice are included in the channel code to accommodate them. Table 4-2 lists subsets of the 4-dimensional D_4 lattice chosen for various values of source quantization levels β. A larger value of β quantizes the signature finely, but this requires a larger scale factor α to keep the probability of error sufficiently low. This in turn degrades the transparency of the watermarked image. The choice of the parameters α and β determines the trade-off between the transparency and the quality of the hidden data.

For security, we can select special regions in the transformation domain to embed data, or randomly group the coefficients to form a vector using a private key. Pseudo-random sequences can be used for random grouping.

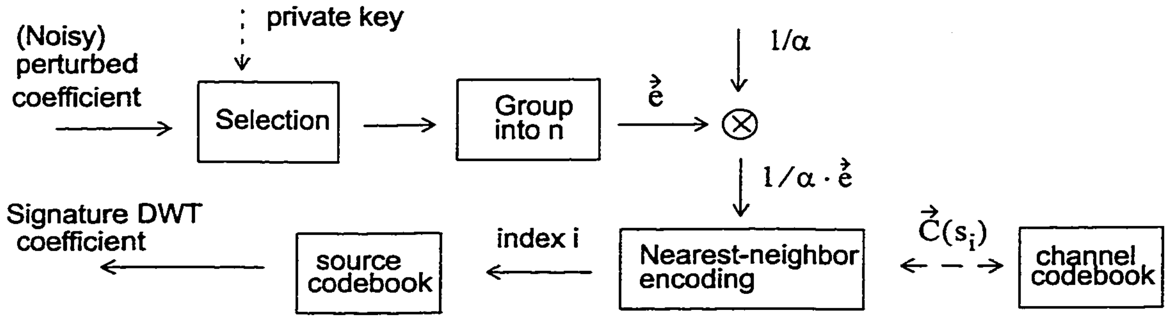


FIGURE 4-4. Signature extraction.

4.3 Signature Extraction

A watermarked image may be subject to lossy compression or image processing operations such as enhancement. Under the assumption that the resulting perturbations in the wavelet transform domain can be modeled by additive Gaussian noise, a nearest-neighbor search with the Euclidean distance measure can be used to recover the embedded symbols. The decoder block in Figure 4-2(b) is expanded in Figure 4-4 to show the details of symbol recovery and signature extraction.

Recovering the hidden data starts with computing the DWT of the embedded image. As before, we assume that the true host image coefficients are available at the decoder. These original host image coefficients are then subtracted from the coefficients of the received image to obtain the noisy perturbations. Note that the perturbations recovered can be "noisy", because of various possible transformations of the watermarked data.

4.3.1 Determining the Closest Point

These coefficients are now grouped into sets of *n* in the same manner as they were grouped during encoding (possibly using a private key) to obtain a vector $\vec{e}$ (Figure 4-4). This is then scaled by the factor $1/\alpha$. The resulting vector $1/\alpha \cdot \vec{e}$ is then the nearest-neighbor encoded to find the index *i* of the channel code nearest to it in Euclidean distance. In particular, we find an index *i* such that:

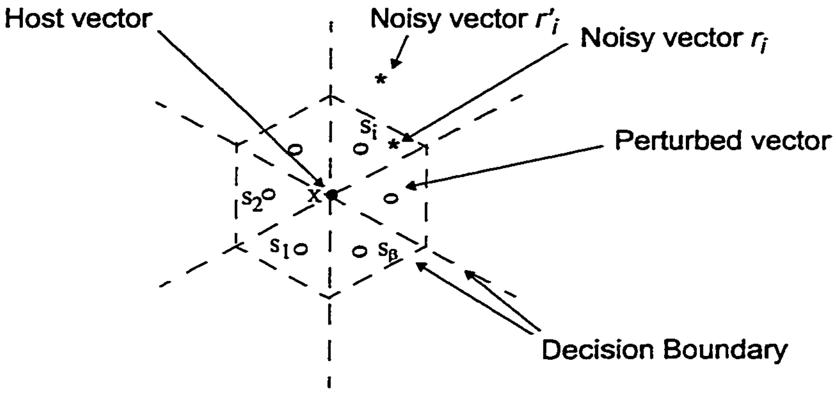


FIGURE 4-5. Decision boundary for a perturbed lattice point.

$$\left\|\vec{C}(s_i)-\frac{1}{\alpha}\vec{e}\right\| \le \left\|\vec{C}(s_j)-\frac{1}{\alpha}\vec{e}\right\|,\quad \forall j \in \{1,2,\ldots,\beta\} \tag{4.7}$$

where $\vec{C}(s_i)$ refer to the β code vectors in the channel codebook. For lattice-based channel codes, this is equivalent to finding the lattice point in whose Voronoi region (see (4.1)) the vector $1/\alpha \cdot \vec{e}$ lies. From the index i, the quantized DWT coefficients can be obtained.

Consider Figure 4-5. Let a perturbed vector corresponding to a channel code s_i be received as a noisy vector r_i. As long as it is inside the decision boundary of the original perturbed vector s_i, we can receive the data perfectly. However, if the embedded vector is recovered as r'_i, located outside of the decision boundary, the symbol detected will not be the original perturbed value s_i. To reduce the incidence of erroneous detection, one can expand the decision boundary by using a larger scale factor, at the expense of quality of the embedded data.

4.3.2 A Fast Algorithm [28,30]

One of the motivations for using lattice-based channel codes in our implementation is the existence of fast encoding and decoding algorithms. We present a fast encoding algorithm for the D_n lattice that is used to extract the hidden symbols from the noisy vectors received, if the number of channel symbols β is sufficiently large.

The algorithm for finding the closest point on the lattice to an arbitrarily scaled noisy perturbation $\mathbf{r}_i = (1/\alpha)\vec{e} \in \Re^n$, is particularly simple. Let $\mathbf{r}_i = [x_1, x_2, ..., x_n]$. Note that all points of D_n are included in the n-dimensional cubic integer lattice I^n. For any $x \in \Re$, let $f(x)$ = *closest integer to x*. We define $f(x)$ and $w(x)$ as follows:

If $x - \lfloor x \rfloor \leq 0.5$, then $f(x) = \lfloor x \rfloor$, $w(x) = \lceil x \rceil$

else $f(x) = \lceil x \rceil$, $w(x) = \lfloor x \rfloor$

We can write $x = f(x) + \delta(x)$, so that $|\delta(x)| \leq 1/2$ is the distance from x to the nearest integer. The vectors $f(x)$ and $g(x)$ are defined by

$$f(\mathbf{x}) = [f(x_1), f(x_2), ..., f(x_k), ..., f(x_n)] \tag{4.8}$$

and

$$g(\mathbf{x}) = [f(x_1), f(x_2), ..., w(x_k), ..., f(x_n)], \tag{4.9}$$

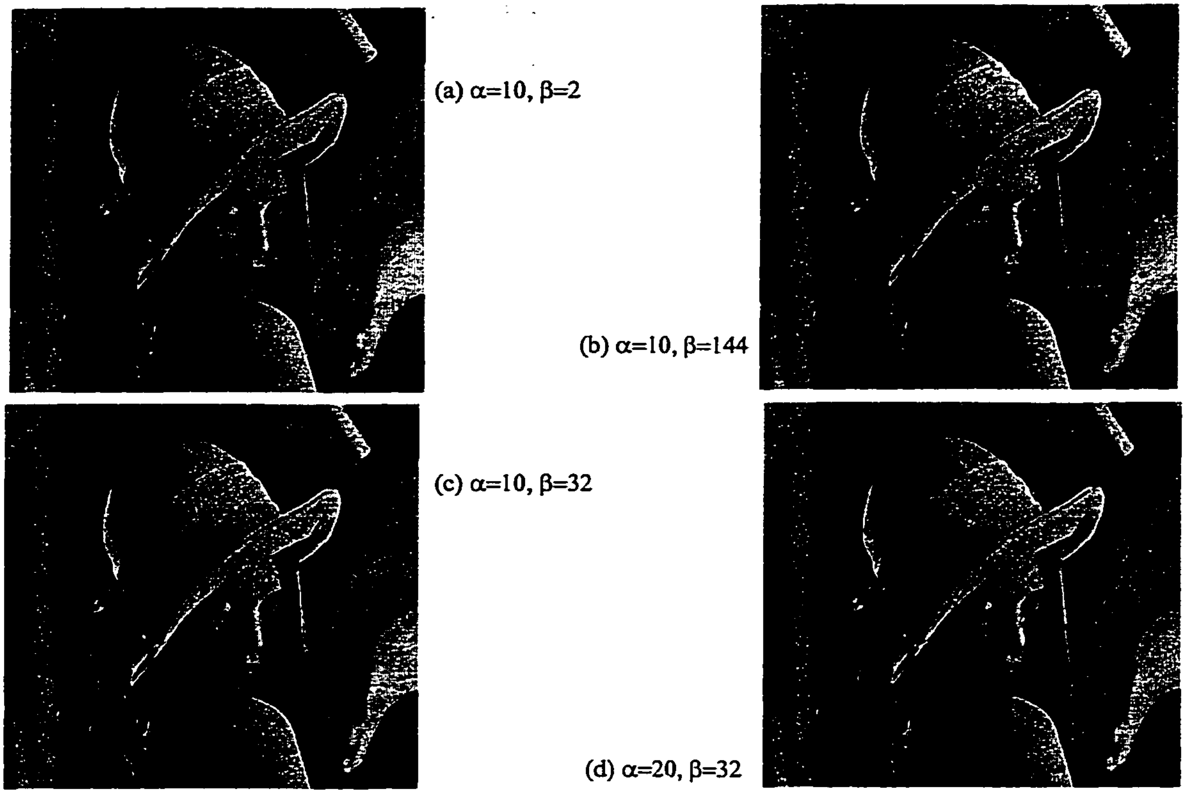
where $k = \arg(\max_i \delta(x_i))$ and $\delta(x_i) = |x_i - f(x_i)|$. The nearest point to x in the D_n lattice structure is chosen as whichever of $f(x)$ and $g(x)$ that has an even sum of components. The fast algorithm works quite well for lossy JPEG compressions of up to 70%. However, if $\mathbf{x}$ computed using (4.8) and (4.9) is equidistant from two or more points of the lattice, we must calculate the nearest point from the relation (4.7) in the previous section.

4.4 Experimental Results

We present results on three different types of embedding: gray image-in-gray image; gray image-in-color image; color image-in-color image; We assume that the original host image is available at the decoder.

4.4.1 Gray Scale Images

Figure 3-4 shows the host and signature images used. A one-stage discrete Haar wavelet transform is used in the following experiments. The D_4 lattice is used for encoding. Figure 4-6 shows the Lena image watermarked with the hat-girl image, at different scale factors α, and various quantization levels β, without any compression. Note that the scale factor α controls the relative weight of host and signature image contributions to the fused image. As α



(a) α=10, β=2

(b) α=10, β=144

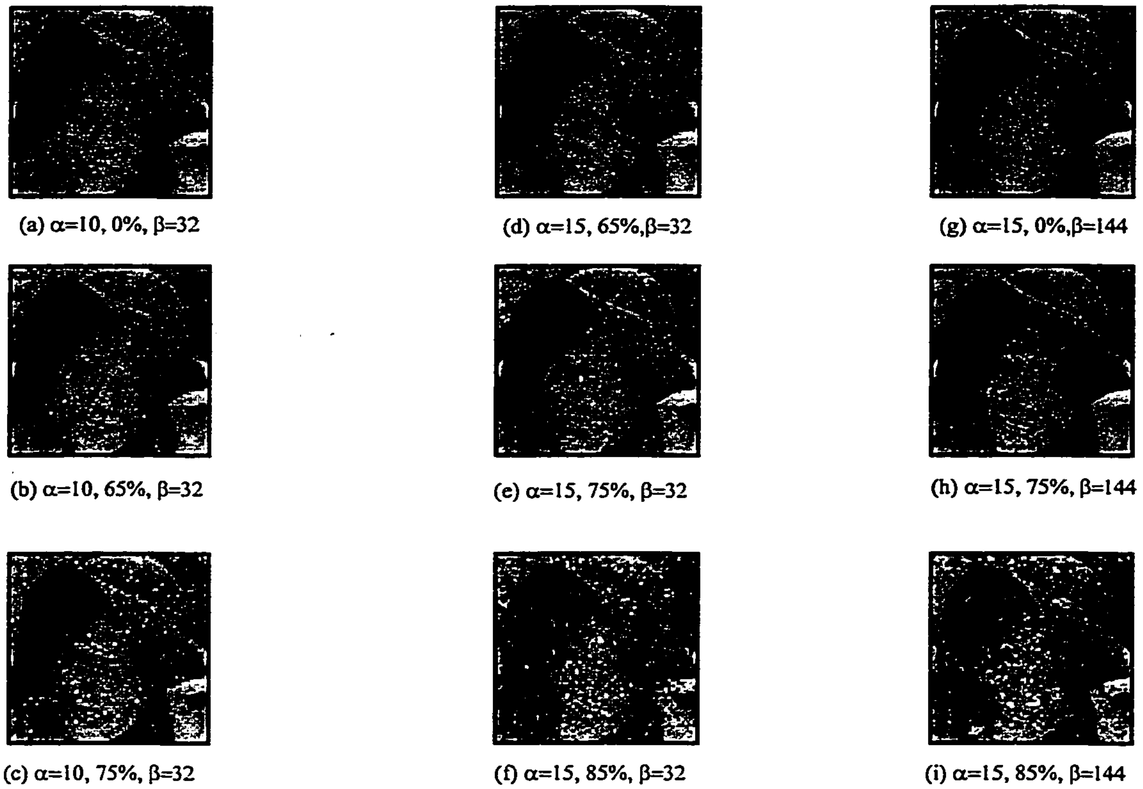
(c) α=10, β=32

(d) α=20, β=32

FIGURE 4-6. Host image "lena" with embedded "hat-girl" image for different scale factors and quantization levels.

increases, the quality of the watermarked image degrades. For example, in Figure 4-6(d), one can see artifacts in the background for α=20. α=10 appears to be a reasonable choice in terms of the trade-off between quality of the watermarked image and robustness to signature recovery under image compression.

Figure 4-7 shows the recovered signature images from the watermarked image after 0%, 65%, 75% and 85% JPEG compression. In general, most of the recovered signature images are of very high quality for 85% JPEG compression, when the scale factor α is in the range 10-15. The quality of the recovered signature with a large scale factor α is obviously much better than those with a smaller α. The number of quantizer levels β, on the other hand, determines the coarseness of quantization and therefore the quality of the signature image hidden in the host. Figure 4-8(a) shows the similarity (computed as described in the previous



(a) α=10, 0%, β=32

(d) α=15, 65%,β=32

(g) α=15, 0%,β=144

(b) α=10, 65%, β=32

(e) α=15, 75%, β=32

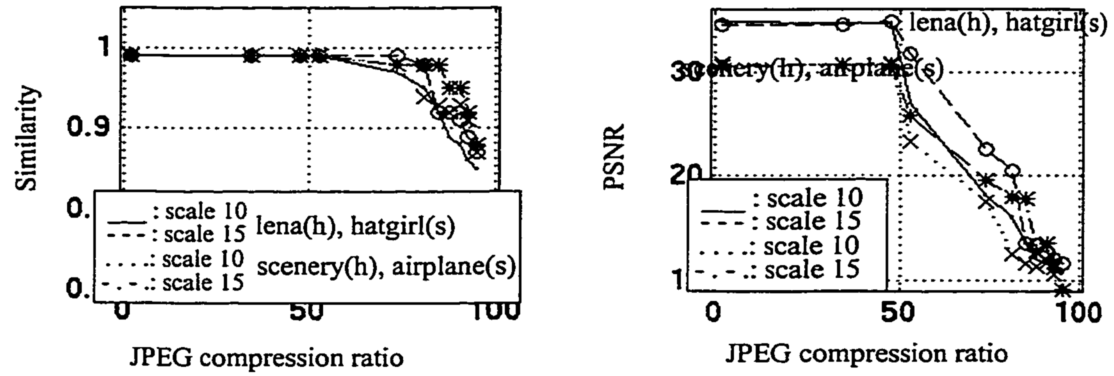
(h) α=15, 75%, β=144

(c) α=10, 75%, β=32

(f) α=15, 85%, β=32

(i) α=15, 85%, β=144

FIGURE 4-7. Extracted "hat-girl" signature images for different scale factors, JPEG ratios, and quantizer factors.



(a) Similarity measure

(b) PSNR results

FIGURE 4-8. Similarity and PSNR results for the multidimensional lattice based embedding algorithm. In this experiment, signature quantization level β = 32.

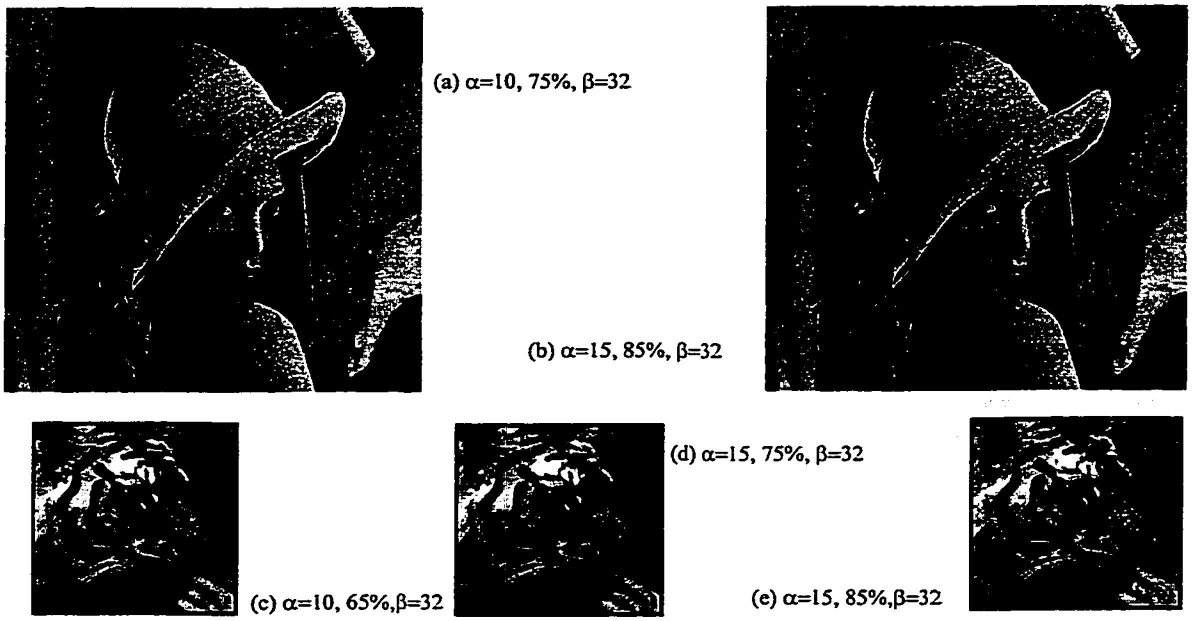
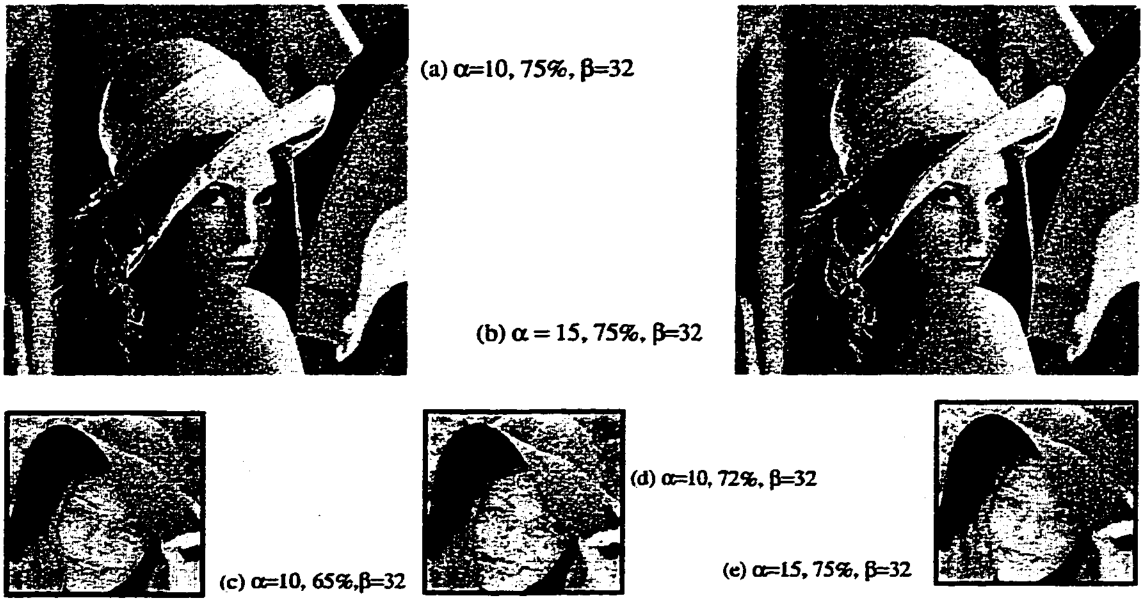

(a) $\alpha=10$, 75%, $\beta=32$
(b) $\alpha=15$, 85%, $\beta=32$
(c) $\alpha=10$, 65%,$\beta=32$
(d) $\alpha=15$, 75%, $\beta=32$
(e) $\alpha=15$, 85%,$\beta=32$

FIGURE 4-9. Another example: (a),(b) Watermarked images using "tiger" signature with JPEG compression, (c)-(e) The recovered signature images following JPEG lossy compression, at various scale factors.

chapter) as a function of the JPEG compression factor, for authentication applications. It is observed that good authentication is possible even at 90% JPEG compression. Here the number of quantization levels used is $\beta=32$. Figure 4-8(b) shows the PSNR of the recovered signature, also as a function of the JPEG compression factor. The recovered signature images are of acceptable quality for up to 85% compression.

Figure 4-9 shows few more examples. Figure 4-9 (a),(b) are the watermarked images with $\beta=32$, with 75% and 85% JPEG compression, respectively. Figure 4-9 (c-e) shows the recovered image from 65%, 75% and 85% JPEG-compressed watermarked images.

Figure 4-10 shows some results for the case using a lossy wavelet transform-based compression method [39]. In this case the recovered signature images are of high quality for up to 75% compression.



(a) $\alpha=10$, 75%, $\beta=32$

(b) $\alpha = 15$, 75%, $\beta=32$

(c) $\alpha=10$, 65%, $\beta=32$

(d) $\alpha=10$, 72%, $\beta=32$

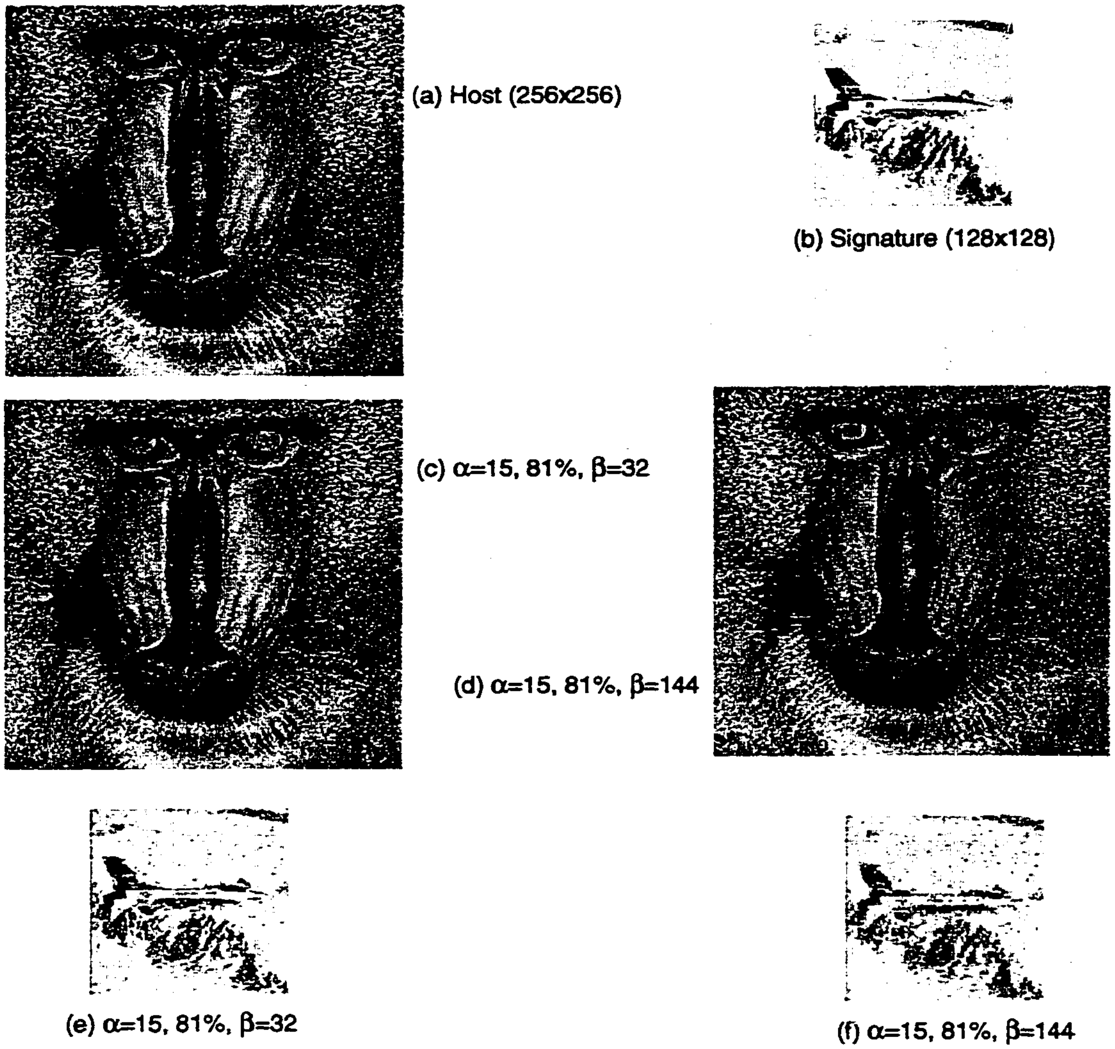
(e) $\alpha=15$, 75%, $\beta=32$

FIGURE 4-10. The results from the lossy wavelet transform based compression: (a), (b), the compressed watermarked images, (c)-(e), the recovered signatures.

4.4.2 Embedding Color Images

The MLDWT method can be extended to embed data in color images. The color images are represented in the YUV color space where the Y component is the luminance part of the signal, and U and V represent the chrominance components. The U and V components are down-sampled by a factor of two. Adopting the YUV color space facilitates a simple extension from images to digital video such as those in the MPEG format. Signature data is embedded only in the luminance component Y so as not to distort the color information.

Figure 4-11 shows an example. Figure 4-11(a) shows a 256x256 color image and Figure 4-11(b) shows a 128x128 gray scale signature. The signature is injected only into the Y component of the transform coefficients of the host image. Figure 4-11(c) shows an 81% JPEG compressed watermarked image using 32 channel codes and Figure 4-11(d) shows the same compressed image using 144 channel codes. Note that there are no visible distortions in



(a) Host (256x256)

(b) Signature (128x128)

(c) α=15, 81%, β=32

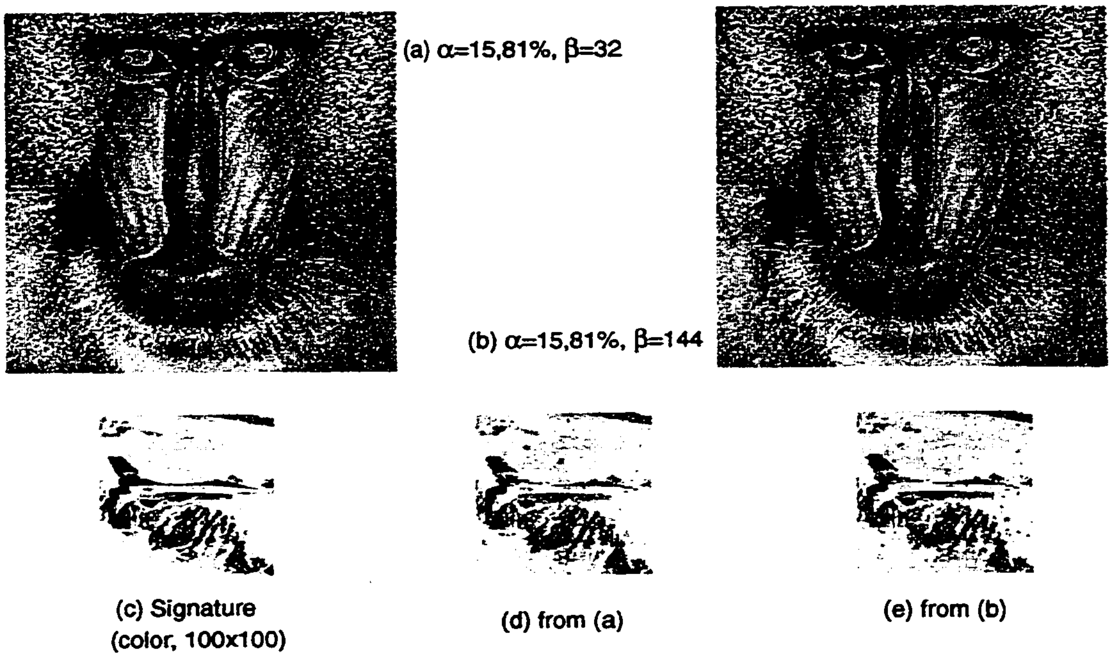
(d) α=15, 81%, β=144

(e) α=15, 81%, β=32

(f) α=15, 81%, β=144

FIGURE 4-11. (a) A color host image, (b) a gray-scale signature image, (c),(d) Watermarked and JPEG lossy compressed images at two different quantization levels, and (e),(f) recovered signature images from (c) and (d), respectively.

the watermarked and JPEG compressed images. Figure 4-11(e) and Figure 4-11(f) show the recovered signatures for the two quantization levels. The reconstructed images are of very good quality for authentication purposes.



(a) α=15,81%, β=32

(b) α=15,81%, β=144

(c) Signature (color, 100x100)

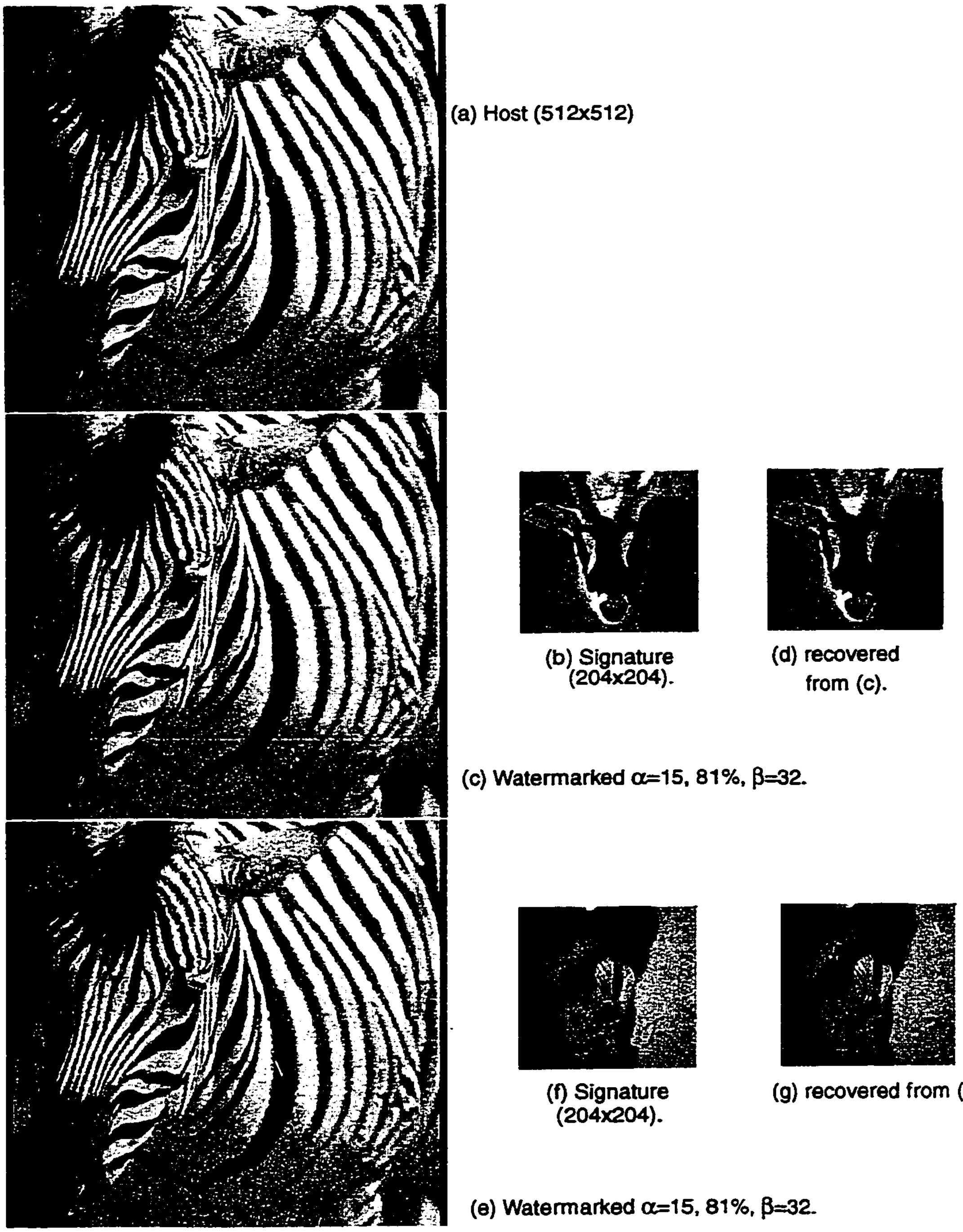
(d) from (a)

(e) from (b)

FIGURE 4-12. (a) Watermarked image, (b) JPEG lossy compressed image of (a), (c) signature image used in (a) and (b), and (d),(e) recovered images.

Figure 4-12 shows another example of a color signature embedding. The entire signature data is embedded in the Y component of the host data in order not to distort the color in the watermarked image. For this reason, the size of the signature image is less than that for a gray-scale embedding. Another example of color image embedding is shown in Figure 4-13.

Figure 4-14(a) shows the similarity of the reconstructed image to the original signature image for various levels of JPEG compression. As can be seen from the graph, the watermarked image can be easily authenticated even at 85% lossy JPEG compression. Figure 4-14(b) shows the Peak Signal to Noise Ratio (PSNR) of the reconstructed signature image as a function of the JPEG compression factor. The PSNR is computed with respect to the original signature before quantization. Note that good quality reconstruction is possible for up to 75% JPEG compression at α=15.



(a) Host (512x512)

(b) Signature (204x204).

(d) recovered from (c).

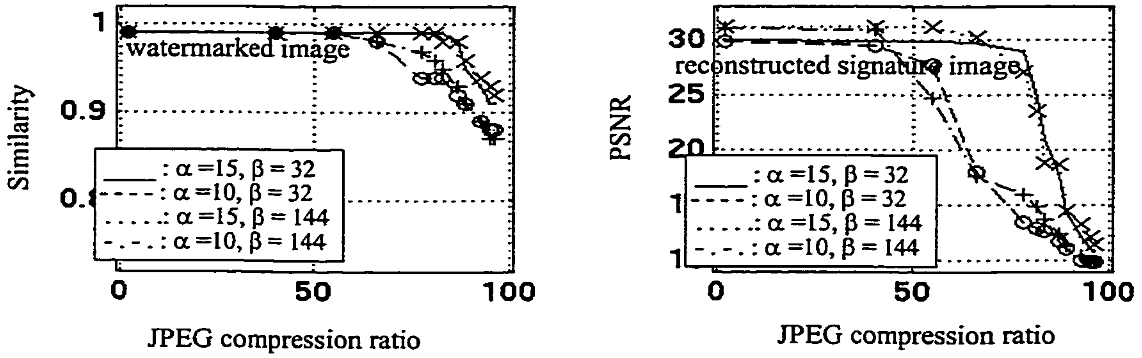
(c) Watermarked α=15, 81%, β=32.

(f) Signature (204x204).

(g) recovered from (e).

(e) Watermarked α=15, 81%, β=32.

FIGURE 4-13. Another example of color image embedding



(a) Similarity measure

(b) PSNR results

FIGURE 4-14. Similarity and PSNR results for the host and signature images shown in Figure 4-11.

4.5 Discussion and Summary

We have presented a scheme for data embedding using the multidimensional lattice in the DWT domain. The scheme presents a framework for a more structured digital watermarking scheme, aimed at embedding large amounts of data into a host. As the results demonstrate, there are no visible distortions in the watermarked images and signature recovery is possible even at 85% lossy JPEG compression.

One can further improve the quality of the recovered signature by using higher dimensional lattice structures. However, the high dimensional lattice can not support large amounts of data hiding.

By properly indexing the scalar codebook used for the wavelet coefficients of the signature image, the recovered signature quality can be substantially improved for the same scale factor of embedding and for the same number of levels for quantization. More sophisticated schemes for error resilience, such as trellis-coded modulation, could also be used.

In the next chapter, we will present an adaptive data hiding method and propose a new technique that can recover the hidden signature data without host information.

Chapter 5

Reconstruction without the Original Host Image

Much of the prior work in signature authentication and in data hiding assumes that the host source is available. Examples of methods that do not require the original host data for signature recovery include [34,31,40,90]. In this chapter we propose an approach to signature recovery that does not require knowledge of the original host by using adaptive hiding techniques. The main contribution here is a technique that has the potential for embedding a significant amount of data which can then be recovered without any additional knowledge of the host.

The proposed embedding and extracting methods utilize the DCT domain. The DCT transform has good energy compaction properties and is used in the current image/video compression standards such as JPEG and MPEG. For this reason it has been frequently used in digital watermarking research [17,60,61,90,91,118,123,129]. By taking into account the specific details of JPEG/MPEG compression methods, one can also make the DCT based embedding more robust to such compression. In this context, we will explore block masking and signature image quantization for adaptively embedding the data into a host image.

We present an adaptive data hiding method with texture masking in the next section, and discuss the signature quantization matrix in Section 5.2. Section 5.3 presents the no-host recovery scheme using host block partitioning that is based on the lattice embedding scheme. The experimental results are given in Section 5.4 and we conclude with discussions in Section 5.5.

5.1 Texture Masking

The human visual system is more sensitive to changes in low frequency regions than to changes in highly textured regions. Thus highly textured regions can be subject to more distortions before such changes are perceivable. This is the basic idea behind texture masking wherein the perturbations to the host data are adaptive to the local texture content.

Many of the recent work on watermarking have used adaptive embedding to improve the quality of the embedded image. These include visual masking [25,70,90,110,118] and watermark generation using a visual model [15,92]. The main objective of these adaptive methods is to improve the quality of the embedded image.

Selective visual masking can be used to make the embedding locally adaptive. For example, Tewfik's group [109-114,132] has used a model for visual masking. One form of visual masking is frequency masking, which refers to a situation wherein a signal raises the visual threshold for other signals around it. A spatial, sinusoidal pattern will lower the detectability of other sinusoidal patterns whose frequencies are close to that of the sinusoidal pattern. This model [73] predicts the detection threshold at a frequency f given the masking frequency f_m and local contrast c_m. In particular, they use a model based on the DCT domain [132]. Similarly, spatial patterns can affect the visibility of other features that are spatially close to them. For example, luminance edges and fine details reduce the visibility of other signals around them.

Huang *et al.* [61] define three different block type categories under the DCT coefficients. They estimate the average brightness and texture complexity from each block and classify into the following the dark and weak texture class, bright and strong texture class, and normal class.

Tao *et al.* [118] propose an adaptive watermarking technique using a noise-sensitivity index from regional classification in the DCT domain. They proposed classifying each block into different noise sensitivity classes and inserting the signals of different energies accordingly. Such properties as luminance masking, edge masking and texture masking effects are exploited according to the human visual system. They divide the data into 6 classes and embed the data adaptively.

In the visual model approach, one can adapt each watermark sequence to the local properties of the image thus providing a watermark that is transparent and robust. Podilchuk *et al.* [91,92] propose an image-adaptive watermarking scheme based on utilizing visual models which have been developed in the context of image compression. Perceptual coders based on the *just noticeable distortion* (JND) thresholds determine optimum quantization step sizes or bit allocations for different parts of the image as determined by a model of the human visual system and local image characteristics. The JND profile estimator can be very useful in determining the maximum amount of energy that can be inserted without causing visual artifacts.

5.1.1 Texture Block Classification

Since the human visual system is more sensitive to the changes in low frequency regions than highly textured regions, data insertion in the textured regions is less likely to result in visible distortions compared to *flat* regions. In the following, we suggest an alternative texture masking scheme that determines the amount of signature data to embed for each 8x8 host block. A scale factor γ controls the amount of inserted signature data. For *flat* regions this scale factor is kept low, whereas for textured regions this is set to a higher value. Since the decisions are made in an 8x8 window, estimation of γ is quite robust and resistant to JPEG compression. Further, at the decoding end the scale parameter can be directly computed from the received (embedded) signal. This is particularly important since we assume that the original host data is not available for reconstruction.

Consider a host 8x8 DWT block using a one-level wavelet decomposition. Let *B={LH, HL, HH}* be the set of subbands (Figure 3-2). A Haar wavelet decomposition is used in our experiments (see Chapter 3). For $b \in B$, Let $\mu_W(b)$ be the average energy in band b of the host image after a one level decomposition. Let $\mu_D(b)$ be the average energy in band b for the block under consideration. Define the block texture energy to be

$$\mu_T(b) = \frac{\mu_D(b)}{\mu_W(b)} \tag{5-1}$$

If $\mu_T(b)$ exceeds a given high threshold, say $T_H(b)$, then the corresponding block is considered to have significant texture in band b. If the block texture energy exceeds the threshold for two out of three bands, then the block is considered to be highly textured. Similarly, if two out of three band energies fall below the low threshold $T_L(b)$, then the corresponding block is considered to be low in texture.

5.1.2 Texture Scale Factor

Each host image DCT block is thus classified into one of *highly textured, normal, or low textured* block. The texture block factor γ is appropriately set for each of these three classes as follows:

$$T_H(b) = \frac{4}{3}, \forall b \in B \tag{5-2}$$

$$T_L(b) = \frac{3}{4}, \forall b \in B \tag{5-3}$$

$$\gamma(\text{high}) = 2, \gamma(\text{normal}) = 0, \gamma(\text{low}) = -2.$$

These values are determined empirically using a set of host and signature images, and subjectively evaluating the quality of the embedded data in different textured regions. Since the inserted signal level varies according to the local texture content, this adaptive masking results in good quality embedded images. Figure 5-9 shows an example of image embedding with and without texture masking. Notice that the distortions in the sky region in the images are less visible with texture masking.

5.2 Signature Quantization Matrix

Quality and quantity of the signature data of as much concern as the quality of the watermarked image. The number of bits required to code and embed the signature depends on how the signature image is quantized. We propose two signature quantization matrices and encoding using lattice codes that can be used to embed signature images up to 25% of the host image size.

The signature image quantization follows along similar lines as the JPEG compression using block DCT. There is an obvious trade-off between the quantity of the signature data and the quality of the reconstructed image. Due to the use of DCT, the proposed embedding

1232	1232	342	342	48	48	0	0
1232	342	342	48	48	0	0	0
342	48	48	48	0	0	0	0
342	48	48	48	0	0	0	0
48	48	0	0	0	0	0	0
48	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)

1232	1232	1232	342	342	342	48	48
1232	1232	342	342	342	48	48	0
1232	342	342	342	48	48	0	0
342	342	342	48	48	0	0	0
342	342	48	48	0	0	0	0
342	48	48	0	0	0	0	0
342	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(b)

FIGURE 5-1. Example of a signature quantization matrix for an 8 x 8 DCT coefficient block. (a) This requires 112 host image coefficients to encode based on the lattice coder. (b) This signature quantization matrix requires 192 host coefficients.

method is especially robust to JPEG compression, as we will demonstrate in Section 5.4.

The signature coefficients are quantized in two steps: first, by using the standard JPEG quantization matrix [1,124], and then by a user-specified signature quantization matrix. The signature quantization matrix determines the relative size of signature data compare to the host data, thus controlling the quantity and quality of the embedded data. These quantized signature coefficients are then encoded using the multidimensional lattices and inserted into the host DCT coefficients.

Consider an 8 x 8 DCT coefficient matrix. From image compression and information theory, it is well known that low frequency coefficients require more bits than the high frequency ones. One such quantization matrix, indicating the number of quantization levels for each of the 64 coefficients, is shown in Figure 5-1(a). These quantized coefficients are embedded in a lattice structure. For simplicity, we will consider only those shells in the lattice structure whose elements are $\{0, \pm 1, \pm 2\}$. One method for distributing these coefficients is given below:

- **Quantization Level=1232.** *Use Lattice type* E_8 : The first and second shells of E_8 lattice combined have 2400 code words. We use the 1232 code words from the combination of first shell and part of second shell in this lattice. Since an E_8 code has eight components, it requires 8 host coefficients to embed one E_8 code. There are 3 coefficients with this quantization, requiring 24 host coefficients to embed.

- **Quantization Level=342.** *Use Lattice type* E_6: The first and second shells of E_6 lattice contain 342 code words. Six host coefficients are needed to embed an E_6 code. The six coefficients in the DCT matrix thus need 36 host image coefficients to embed.
- **Quantization Level =48.** *Use Lattice type* D_4: The first two shells of D_4 are used to encode 48 levels. Each D_4 code requires four host coefficients. There are thirteen coefficients with this quantization, thus requiring 52 host coefficients.

The scheme outlined above thus needs a total of 112 host coefficients (3x8 + 6x6 + 13x4 = 112 coefficients) to embed one 8x8 DCT block of coefficients from the signature image. The quantized coefficients are transformed to a lattice code, and the code is embedded into a partitioning of the host DCT block (see Section 5.3.2)

Another example of signature image quantization and the corresponding host coefficient allocation are shown in Figure 5-1(b). Notice that 192 host coefficients are needed for this case (6x for E_8, 16x for E_6, and 12x for D_4 = 6x8 + 16x6 + 12x4 = 192 coefficients).

5.3 MLDCT: A Method for Image Embedding and No-Host Recovery

We now outline the data embedding and reconstruction procedure. The main components include host DCT block partitioning, signature image quantization, and texture masking. A schematic of this method is shown in Figure 5-2.

5.3.1 Host Image Block Partitioning

First step in the embedding procedure is to identify the host coefficients that are modified due to insertion. Consider a host 8x8 DCT block (Figure 5-3(a)). The DCT 8x8 block is partitioned into three frequency parts - low, middle, and high - as shown in Figure 5-3(b).

The low frequency components contain most of the host signal energy but cannot be easily modified as such changes may become visible. The high frequency components, which

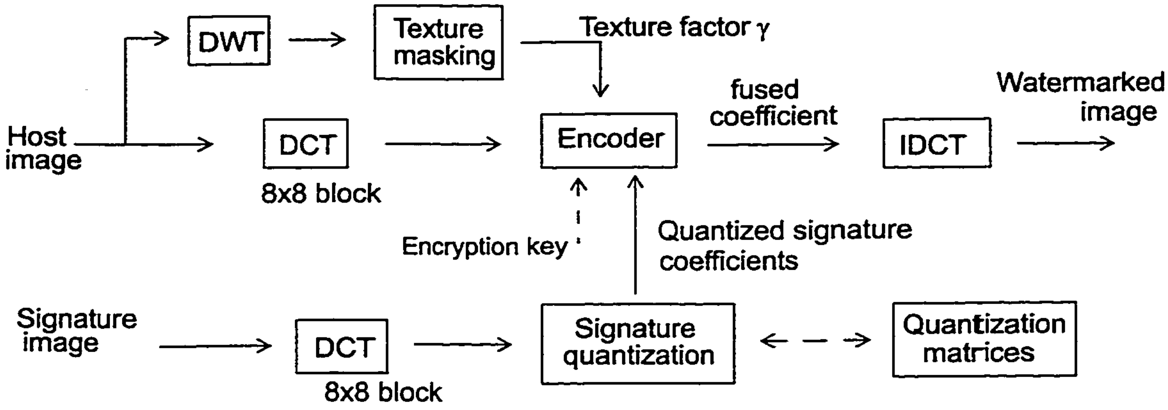
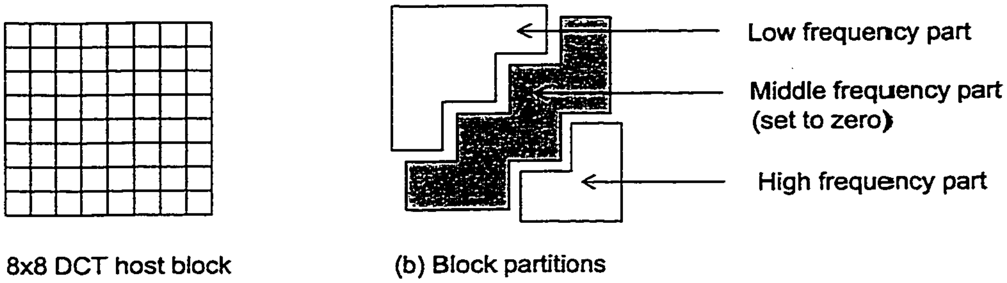


FIGURE 5-2. A schematic of the MLDCT data embedding procedure.



(a) 8x8 DCT host block

(b) Block partitions

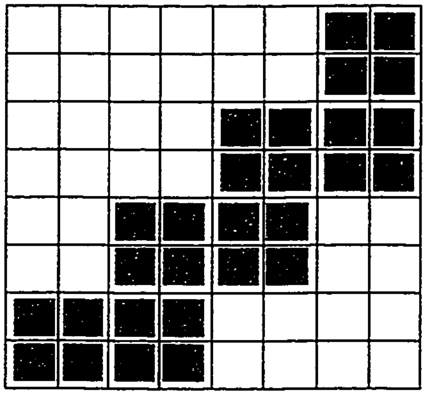
FIGURE 5-3. High and Low frequency masks.

usually pack the least amount of energy, could be easily removed because of signal processing operations. This leaves us with the middle frequency components (shaded region in Figure 5-3(b)). This mid frequency band in each host DCT 8x8 block is set to zero before replacement by the signature information. The zero setting creates a *zero host vector* which is needed for recovery when the original host signal is unknown. Private keys can be used to appropriately select a subset of host coefficients for modification.

Figure 5-4(b) shows an example of selected 28 coefficients (shaded components) which are available in each of the DCT blocks for embedding. One signature 8x8 DCT block, as shown in Figure 5-4(a), is quantized for embedding, which will require 112 host coefficients.

1232	1232	342	342	48	48	0	0
1232	342	342	48	48	0	0	0
342	48	48	48	0	0	0	0
342	48	48	48	0	0	0	0
48	48	0	0	0	0	0	0
48	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a)



(b)

FIGURE 5-4. (a) Example of a signature quantization matrix for an 8 x 8 DCT block. This requires 112 host image coefficients to encode (see text for details). (b) A partitioning of the host DCT block for signal insertion (shaded regions). 28 coefficients are used in each block. Thus, four host DCT blocks (4x28 =112) are needed to embed one 8x8 signature DCT block.

Note that four host DCT blocks are needed to embed one 8x8 signature DCT block. Figure 5-5 shows another example. In this case, the 12 host coefficients (shaded components) selected from each host block. The quantization levels, as shown in Figure 5-5(a), require 192 host coefficients for embedding. This requires sixteen host blocks for embedding one 8x8 signature block.

5.3.2 Embedding Procedure: MLDCT

Figure 5-6 shows the details of the encoding block from Figure 5-2. The host and signature images are transformed by the block-based DCT. Each 8x8 host image block is analyzed for its texture content, as explained in Section 5.1.2, and the block factor γ is computed. The steps in embedding are:

1. The signature coefficients are quantized according to the method described in Section 5.2. The quantized coefficients are encoded using a lattice coder (Chapter 3), chosen such that the code vectors contain only ± 2, ± 1 or zeros.
2. The signature codes are then appropriately scaled using the total scale factor $\delta = \alpha + \gamma$ and the JPEG quantization matrix [124]. The JPEG quantization matrix

1232	1232	1232	342	342	342	48	48
1232	1232	342	342	342	48	48	0
1232	342	342	342	48	48	0	0
342	342	342	48	48	0	0	0
342	342	48	48	0	0	0	0
342	48	48	0	0	0	0	0
342	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(a) signature quantization

						■	■
						■	
				■	■		
				■			
		■	■				
		■					
■	■						
■							

(b) Selected positions for embedding

FIGURE 5-5. Another example of a signature quantization matrix and the corresponding host coefficient allocation. This requires 192 host coefficients, which are distributed over 16 blocks, 12 coefficients per block, as shown by the shaded regions in (b).

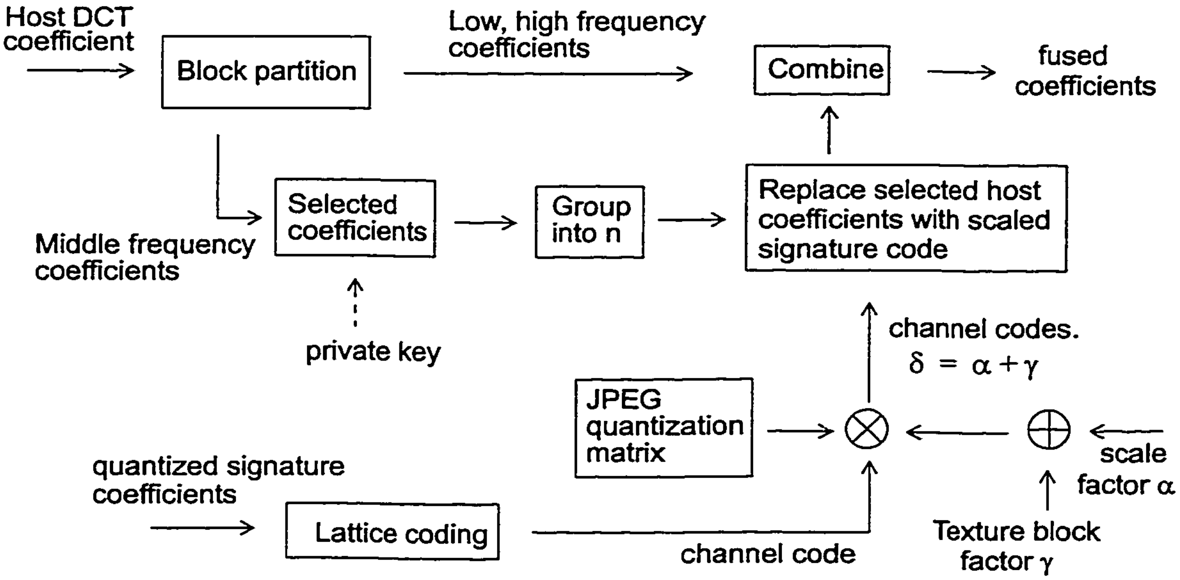


FIGURE 5-6. A schematic of the encoder

helps in re-normalizing the code vectors so that they have a similar dynamic range as a typical DCT block. Note that $\delta \geq 0$, which in turn constrains the choice of α and γ.

3. The selected host coefficients are then replaced by the scaled signature codes and combined with the original (unaltered) DCT coefficients to form a fused block of DCT coefficients. Note that more than one host coefficient is needed to encode a single signature code.
4. The embedded coefficients are combined with low and high components of the original host DCT block.
5. The fused coefficients are then inverse transformed to give an embedded image.

The choice of signature quantization matrix affects the quantity and quality of the embedded data. Choice of the scale parameter α depends on the application. A larger value for α results in a more robust embedding at the cost of quality of the embedded image, i.e., there could be perceivable distortions in the embedded image. A smaller α may result in poor quality recovered signature when there is a significant compression of the embedded image.

5.3.3 Extracting Procedure

Figure 5-7 shows the schematic of the decoder without the host image. Signature reconstruction essentially follows an inverse sequence of operations. Note that signature extraction does not require the original host image. The degraded watermarked image is first transformed by DCT. The coefficients corresponding to the signature and host data are identified. The high frequency components are neglected during the reconstruction phase. The host image is recovered by the inverse DCT of the low frequency coefficients.

By appropriately scaling the coefficients corresponding to the signature data, the lattice codes representing the quantized signature coefficients are recovered. Assuming the zero-host vector, computing the signature coefficient values is straightforward and very similar to the method described in the previous chapter (Section 5.3.1). Finally, the signature image is obtained by inverse transforming the block DCT coefficients.

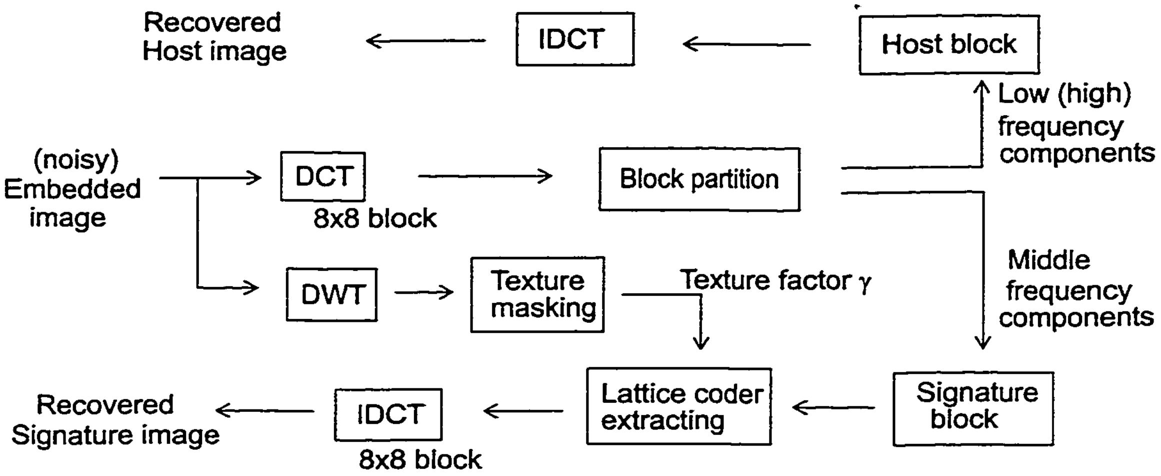


FIGURE 5-7. Signature and host image recovery.

5.4 Experimental Results

Figure 5-8 shows the host and signature gray images used in the experiments. We use two different sizes for the host image: For embedding using the signature quantization matrix of Figure 5-1(a), a 256x256 host image is used, resulting in 25% data embedding. A 512x512 host image is used with the quantization matrix of Figure 5-1(b), using only a one-sixteenth size signature image.

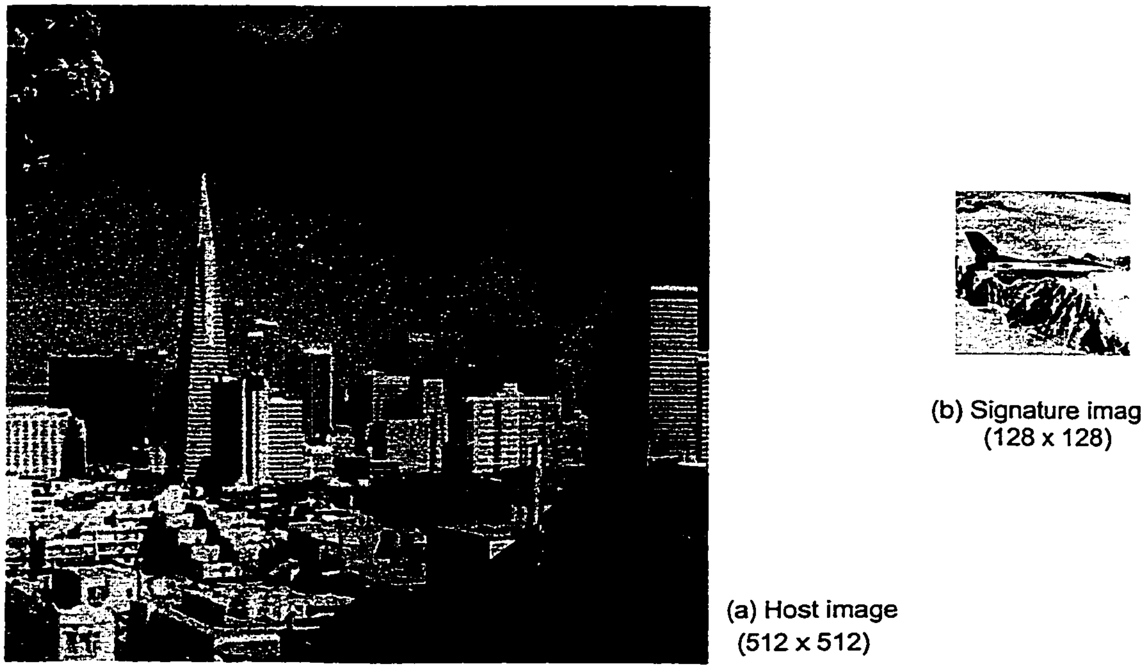
Figure 5-9 shows the embedded images using the MLDCT with and without texture masking. The signature quantization matrix shown in Figure 5-1(b) is used for this. From Figure 5-9(b), it is clear that texture masking reduces visible distortions in regions that are *flat*, as in the sky region of the image. Figure 5-10 shows recovered host and signature images for two different quantizations of the signature data, using texture masking. In this case, the embedded images are lossy JPEG compressed to 89%. Obviously, the quantization matrix of Figure 5-5 yields better results than the one shown in Figure 5-4 at the cost of more host bits per signature coefficient.

Figure 5-11 shows the quality of the embedded and recovered images using the PSNR as a measure. It is clear from these graphs that one can achieve better quality embedding using the quantization matrix of Figure 5-5 at the cost of lower bit rate for the hidden data. Even at 25% embedding, one can recover visually acceptable quality results for up to 90% lossy JPEG compression.

Notice that the PSNR of the embedded image stays almost constant, independent of the JPEG compression (Figure 5-11(a)). This is because the embedding procedure is based on block DCT, as is JPEG compression. This also explains why the PSNR actually *increases* beyond 90% JPEG compression. The mid-frequency components, which encode the signature signal, are being removed at this point, thus making the embedded image more similar to the original host image!

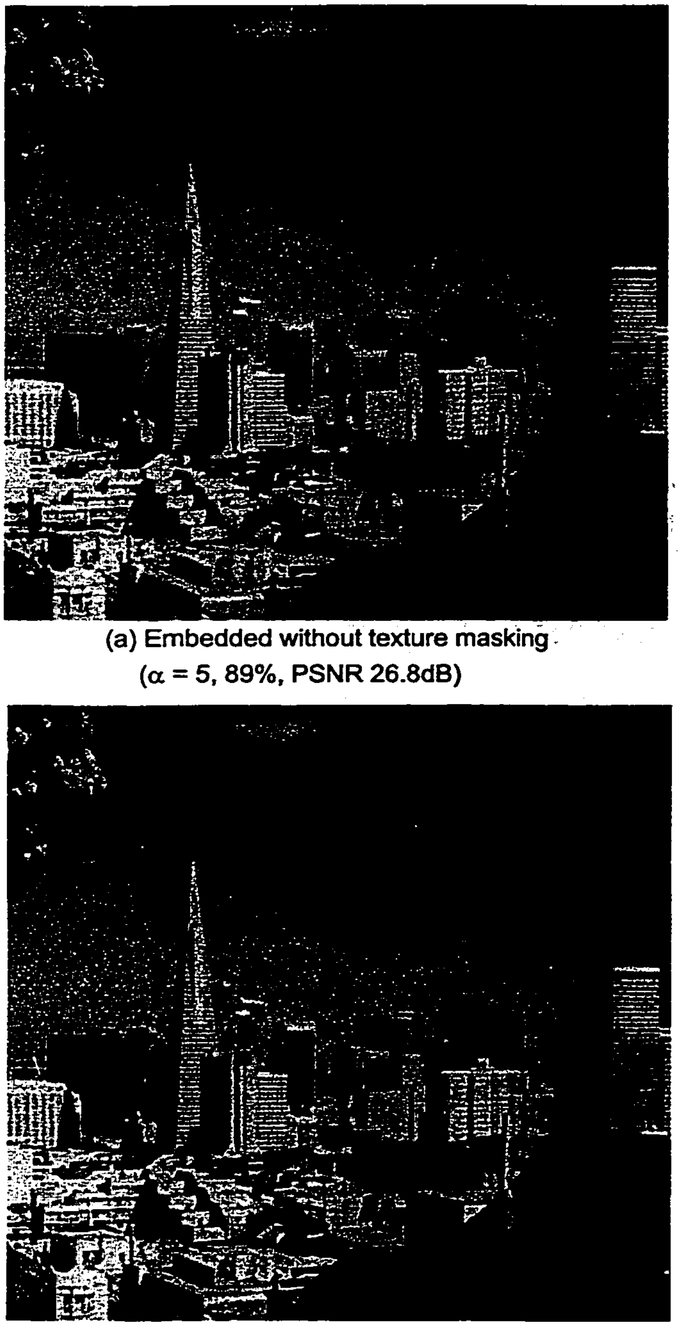
5.5 Discussions

Figure 5-12 shows the PSNR of the embedded image as a function of *embedding* random Gaussian noise at varying standard deviations. This is not additive Gaussian noise added to the host, but rather random Gaussian noise embedded using the MLDCT method. The


(b) Signature image (128 x 128)

(a) Host image (512 x 512)

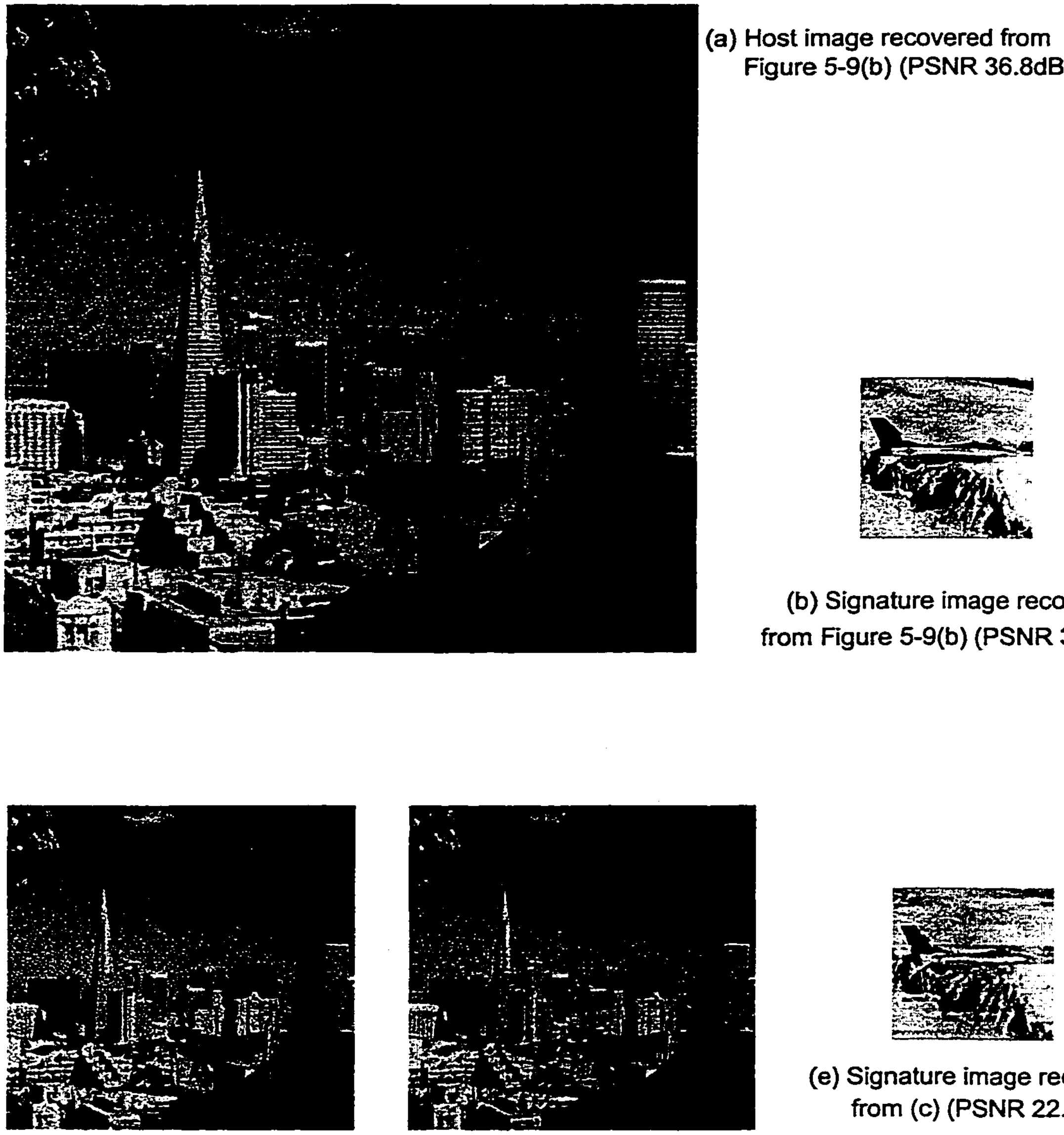
FIGURE 5-8. Test images.



(a) Embedded without texture masking
($\alpha = 5$, 89%, PSNR 26.8dB)

(b) Embedded with texture masking
($\alpha = 5$, $\gamma = 2$, 89%, PSNR 29.4dB)

FIGURE 5-9. Watermarked images with and without texture masking. The signature quantization matrix shown in Figure 5-1(b) is used. Host image is 512x512 pixels and the signature image is 128x128 pixels. Notice the visible distortions in the sky region in (a).



(a) Host image recovered from Figure 5-9(b) (PSNR 36.8dB)

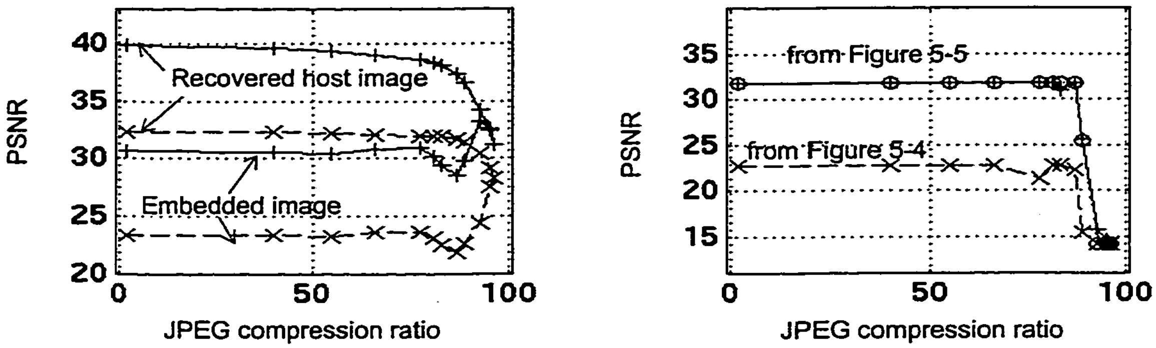
(b) Signature image recovered from Figure 5-9(b) (PSNR 31.7dB)

(c) Embedded image (256x256) (a=5, 89%, PSNR 22.7dB)

(d) Host image recovered from (c) (PSNR 31.4 dB)

(e) Signature image recovered from (c) (PSNR 22.2dB)

FIGURE 5-10. Data embedding and recovery at two different bit rates using texture masking. (a), (b) show results at 6% embedding; (c), (d) and (e) show the results for embedding signature data whose size is 25% of the 256x256 host image.



(a) Watermarked and Recovered host images

(b) Recovered signature image

FIGURE 5-11. PSNRs of embedded, recovered host and signature images (with scale factor 5) for different lossy JPEG compression factors. The solid lines are for 6% embedding using the quantization matrix of Figure 5-6. The dashed line shows the results at 25% embedding using the quantization matrix of Figure 5-4.

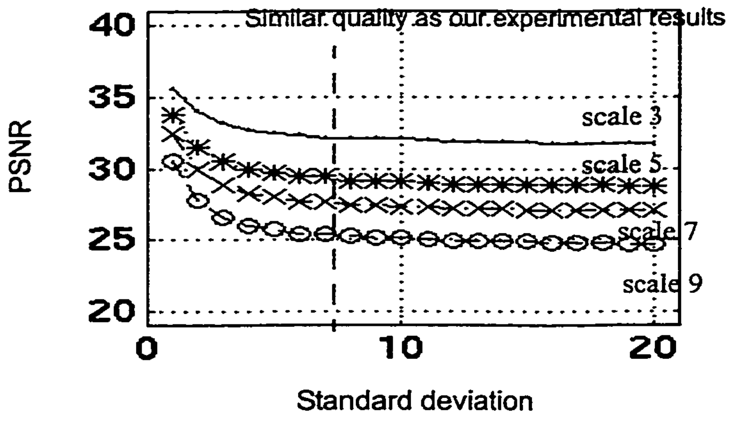


FIGURE 5-12. PSNR of embedding gaussian noise with variable variance and at different scale factors.

noisy image is of size 128x128 and the signature quantization matrix of Figure 5-1(b) is used (6.25% embedding). Results for different scale factors are given. The purpose of this exercise is to draw a similarity between signature image embedding and random noise insertion, in terms of PSNR. From the graph, it appears that a Gaussian distribution of standard deviation 8, would result in a similar PSNR as a signature. This provides a rough estimate of how much one can corrupt the image before the image shows perceivable distortions.

In summary, we have proposed a robust data hiding technique for embedding images in images. A key component of this scheme is the use of multidimensional lattice codes for encoding signature image coefficients before inserting them into the host image DCT coefficients. Texture masking is used to reduce distortions in the embedded image by adaptively controlling the weights associated with the hidden data. The hidden signature data can be recovered in the absence of the original host image. Experimental results show that this method is robust to lossy image compression using JPEG. One can trade-off quantity for quality of the embedded image by choosing appropriate signature quantization matrices.

Chapter 6

Vector Embedding

In Chapters 3 and 4, we examined data embedding using wavelet transform. The data is fused in the wavelet transform domain. For example, signature image coefficients in the LL band are combined with the host image coefficients in the LL band to result in the LL-band fused image coefficients. Since the coefficients are merged in the corresponding sub-bands, we refer to this class of techniques as scalar embedding.

In this chapter, we explore a novel extension to the scalar methods, wherein the signature and host coefficients are considered to be part of complementary (orthogonal) sub-spaces. As we will see in the next section, direct embedding using coefficients from two orthonomal vector spaces is not practical. In the proposed method, using the host and signature data, a new signal for insertion is created. This signal is referred to as the β-Signal. This β-Signal is then inserted into the host using the MLDCT method described in Chapter 5.

We will demonstrate the feasibility of reconstructing the β-Signal, and hence the original signature, from the embedded signal. As in the MLDCT method, this does not require the original host data in the decoding process. We call this method *Vector Embedding* to differentiate it from the scalar embedding methods described earlier, and refer to this method as MVDCT.

The next section explains the embedding approaches for the wavelet coefficients with a review of scalar embedding and the theoretical approach of vector embedding. Section 6.1 and Section 6.2 discusses the orthogonal vector addition embedding technique including β-Signal calculation. Section 6.3 represents the embedding method with β-signal. Experimental results are given in Section 6.4 and conclusions in Section 6.5.

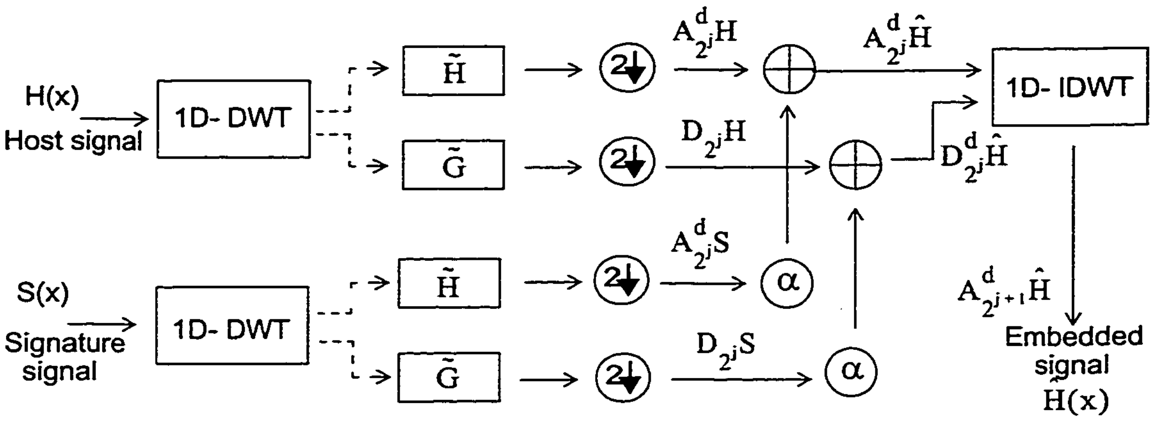


FIGURE 6-1. A schematic of embedding in the DWT domain.

6.1 Embedding in Wavelet Domain Revisited

Figure 6-1 which shows a schematic of the scalar embedding approach discussed in Chapter 3. As mentioned earlier, in scalar embedding the host and signature image coefficients are added in the corresponding subbands to obtain the fused coefficients, as given by (6.1) and (6.2).

$$A^d_{2^j}\hat{H}(x) = A^d_{2^j}H(x) + \alpha \cdot A^d_{2^j}S(x) \tag{6.1}$$

$$D_{2^j}\hat{H}(x) = D_{2^j}H(x) + \alpha \cdot D_{2^j}S(x). \tag{6.2}$$

The combined coefficients are then inverse transformed to obtain the embedded signal $\hat{H}(x)$.

Thus the approximation signal $A^d_{2^j}\hat{H}$ is located in the vector space V_{2^j}. Similarly, the detail signal $D^d_{2^j}\hat{H}$ is located in the orthonomal vector space O_{2^j}.

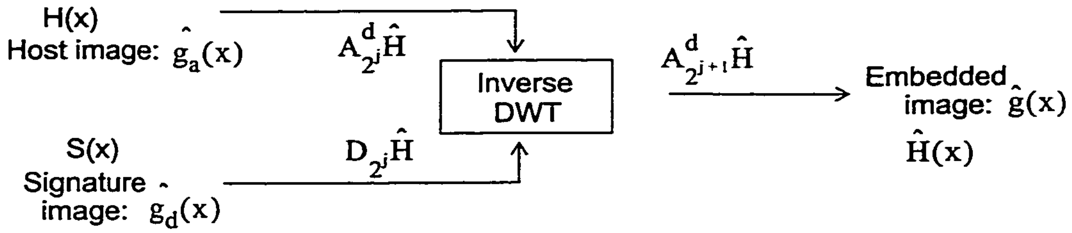


FIGURE 6-2. A schematic of vector embedding. $\hat{H}(x)$ is the virtual embedded image.

6.1.1 Vector Embedding

Recall (Chapter 3) that the vector space of $V_{2^{j+1}}$ is the union of V_{2^j} and O_{2^j}.

$$O_{2^j} + V_{2^j} = V_{2^{j+1}}. \tag{6.3}$$

Thus the approximation and the detail signals at resolution 2^j combined to give the approximation at the resolution 2^{j+1}, i.e., given a signal $\hat{f}(x)$, one can write,

$$A^d_{2^{j+1}}\hat{f} = D_{2^j}\hat{f} \oplus A^d_{2^j}\hat{f}. \tag{6.4}$$

Consider now the following scenario. Let the host signal $H(x)$ be the approximation at resolution 2^j. Denote this signal by $\hat{g}_a(x)$. Let the signature signal be the corresponding detail signal at resolution 2^j. Denote this signal by $\hat{g}_d(x)$. Let

$$D_{2^j}\hat{g}_d \oplus A^d_{2^j}\hat{g}_a = A^d_{2^{j+1}}\hat{g}. \tag{6.5}$$

This is illustrated schematically in Figure 6-2.

Figure 6-3 illustrates this with an image examples. Here $H(x)$ is the host image, and $S(x)$ is the signature image. Assume that $H(x)$ and $S(x)$ have the same number of pixels. Further, it is assumed that

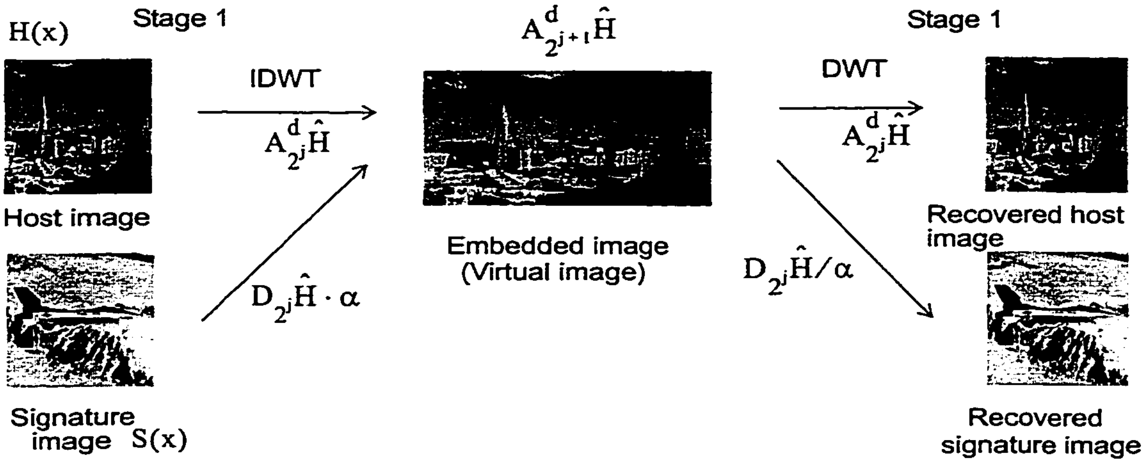


FIGURE 6-3. An embedding scheme which combines data from the approximation and detail signals.

$$A^d_{2^j}\hat{H}(x) = H(x), \tag{6-6}$$

$$D_{2^j}\hat{H}(x) = S(x), \tag{6-7}$$

$$A_{2^{j+1}}\hat{H}(x) = H(x) \oplus S(x). \tag{6-8}$$

In the above example, the images are considered as one-dimensional signals by row scanning the pictures. Note that the resulting fused image consists of twice as many pixels.

6.2 The β-Signal

The de-embedding of $\hat{H}(x)$ is robust to additive noise. However, the increase in the size of the embedded signal is clearly a disadvantage. One can sub-sample the embedded signal to reduce the dimension, but this seriously affects the reconstruction.

Consider $\hat{H}(x)$, as defined in (6-8). Let

$$\tilde{H}(x) = (\hat{H}(x))\downarrow_2 \tag{6-9}$$

Thus $\tilde{H}(x)$ is generated by keeping all the even samples of $\hat{H}(x)$.

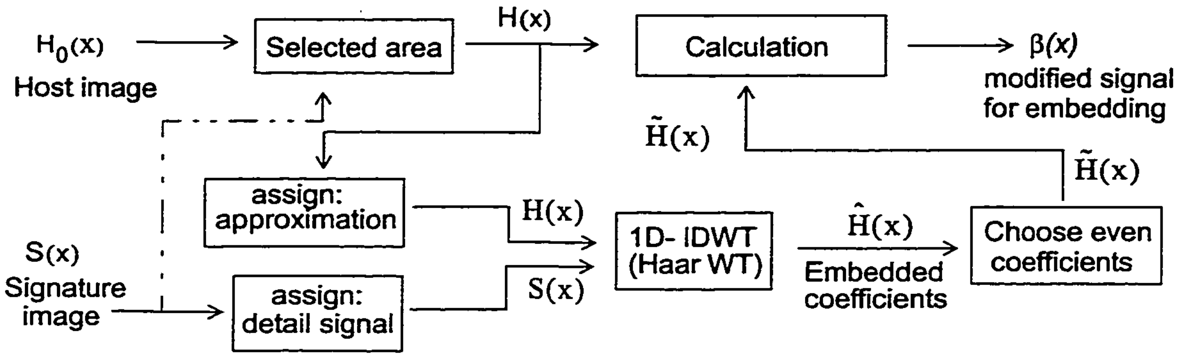


FIGURE 6-4. Generating the β -signal.

Note that, from (3.8), we have

$$\tilde{H}(x) = \frac{1}{\sqrt{2}}(H(x) + \alpha \cdot S(x)) \qquad (6\text{-}10)$$

Ideally, we would like to have

$$\tilde{H}(x) \approx H(x). \qquad (6\text{-}11)$$

Consider now the modulation of $H(x)$ by a signal $\beta(x)$ such that

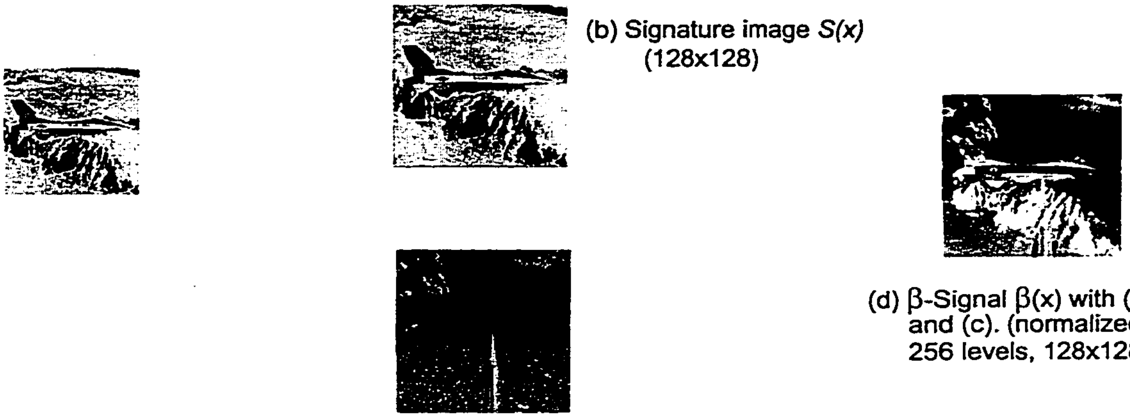
$$\tilde{H}(x) \approx H(x) = \beta(x) \cdot \frac{1}{\sqrt{2}}(H(x) + C) + \frac{1}{\sqrt{2}}(\alpha \cdot S(x) + C) \qquad (6\text{-}12)$$

where *C* ia a constant. From (6-12),

$$\beta(x) = \frac{\sqrt{2}H(x) - (\alpha \cdot S(x) + C)}{H(x) + C}. \qquad (6\text{-}13)$$

$\beta(x)$ is simply a non-linear gain factor that is dependent on the deviation of $H(x)$ from $S(x)$. We refer to $\beta(x)$ as the β-*Signal.* In the embedding procedure, the β-*Signal is our modified signature that is now embedded into the host signal* instead of the original signature image.

Figure 6-4 shows a schematic of this procedure for generating the β signal. Figure 6-5(d) shows the $\beta(x)$ for the airplane signature image and the pyramid host image. The host and the signature test images are gray scale 256x256 and 128x128 size images, respectively.



(b) Signature image *S(x)* (128x128)

(d) β-Signal β(x) with (b) and (c). (normalized 256 levels, 128x128)

(a) Host image $H_o(x)$ (256x256)

(c) Selected host image region *H(x)* (128x128)

FIGURE 6-5. An example of a β–signal.

Figure 6-5(c) is the selected host image used for computing this β signal. Summarizing the steps:

1. Chose the selected host image region $H(x)$ that is of the same size as $S(x)$. In our examples, the signature size is one quarter the size of the host image. A private key can be used to encrypt the data.
2. Consider $H(x)$ as the approximation signal $A^d_{2^j}\hat{H}(x)$ and $S(x)$ as the detail signal $D_{2^j}\hat{H}(x)$. The virtual embedded approximation signal $A^d_{2^{j+1}}\hat{H}(x)$ is then obtained by the one-dimensional inverse DWT operation.
3. Choose only the even coefficients of $\tilde{H}(x)$.
4. Calculate the β-Signal using (6-13).

6.3 Embedding Procedure with β-Signal: MVDCT

The β-Signal is embedded into the host using the MLDCT method presented in Chapter 5. After computing $\beta(x)$, we compute the 8x8 block DCT and quantize the DCT coefficients using the quantization matrix (see Chapter 5). The quantized coefficients are then lattice coded and adaptively embedded into the host image using texture masking. Figure 6-6 shows

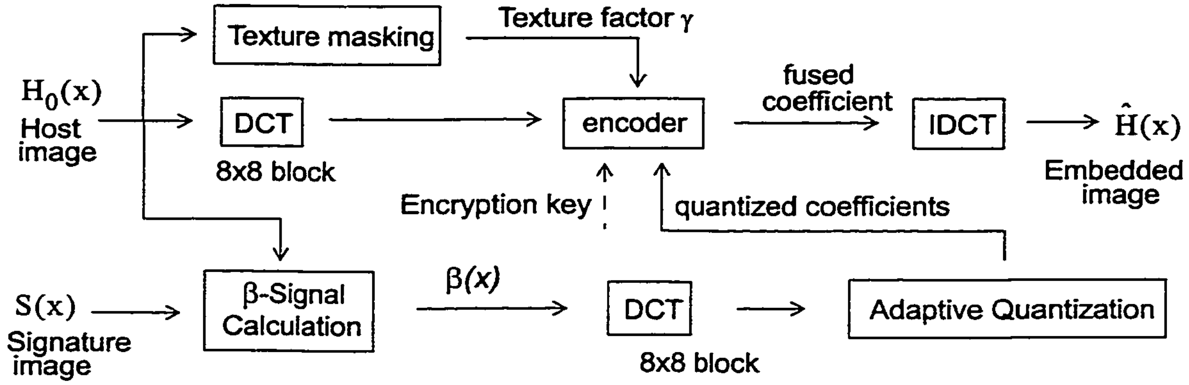


FIGURE 6-6. A schematic of the MVDCT procedure.

a schematic of our proposed data embedding algorithm. The embedded image $\hat{H}(x)$ is obtained by the inverse DCT of the encoded coefficients.

6.3.1 Extraction

Decoding the embedded image follows an inverse sequence of operations. Given an embedded image $H_E(x)$, we compute the host and β-Signal images as explained in Section 5.3.2. Let a degraded embedded image be denoted by $\widehat{H}_E(x)$. Figure 6-7 shows a schematic for extracting the hidden data. From (6-13), the signature is then computed as

$$\widehat{S}(x) = \frac{1}{\alpha}\cdot(\widehat{H}_E(x)\cdot(\sqrt{2}-\widehat{\beta}(x))-C\cdot(\widehat{\beta}(x)+1)) \tag{6-14}$$

where $\widehat{H}_E(x)$ and $\widehat{\beta}(x)$ are the reconstructed host and signature data after being recovered from the MLDCT method.

As compared to the MDWT and MLDWT methods in Chapter 3 and 4, signature recovery using (6-14) is quite stable even though the original host image is not available. This is due to the fact that the β-Signal adds only the non-redundant information about the signature image. This is further supported by our experimental results detailed below.

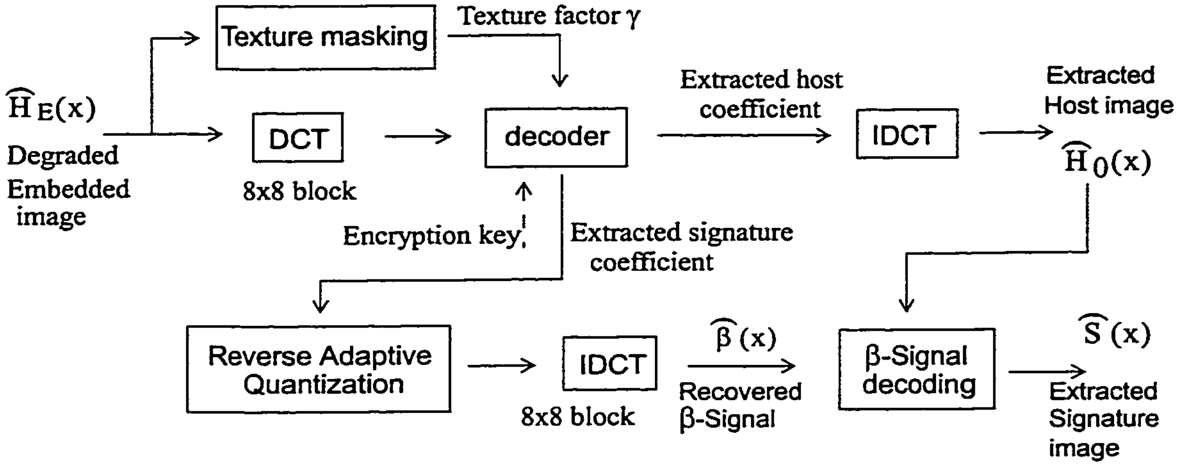


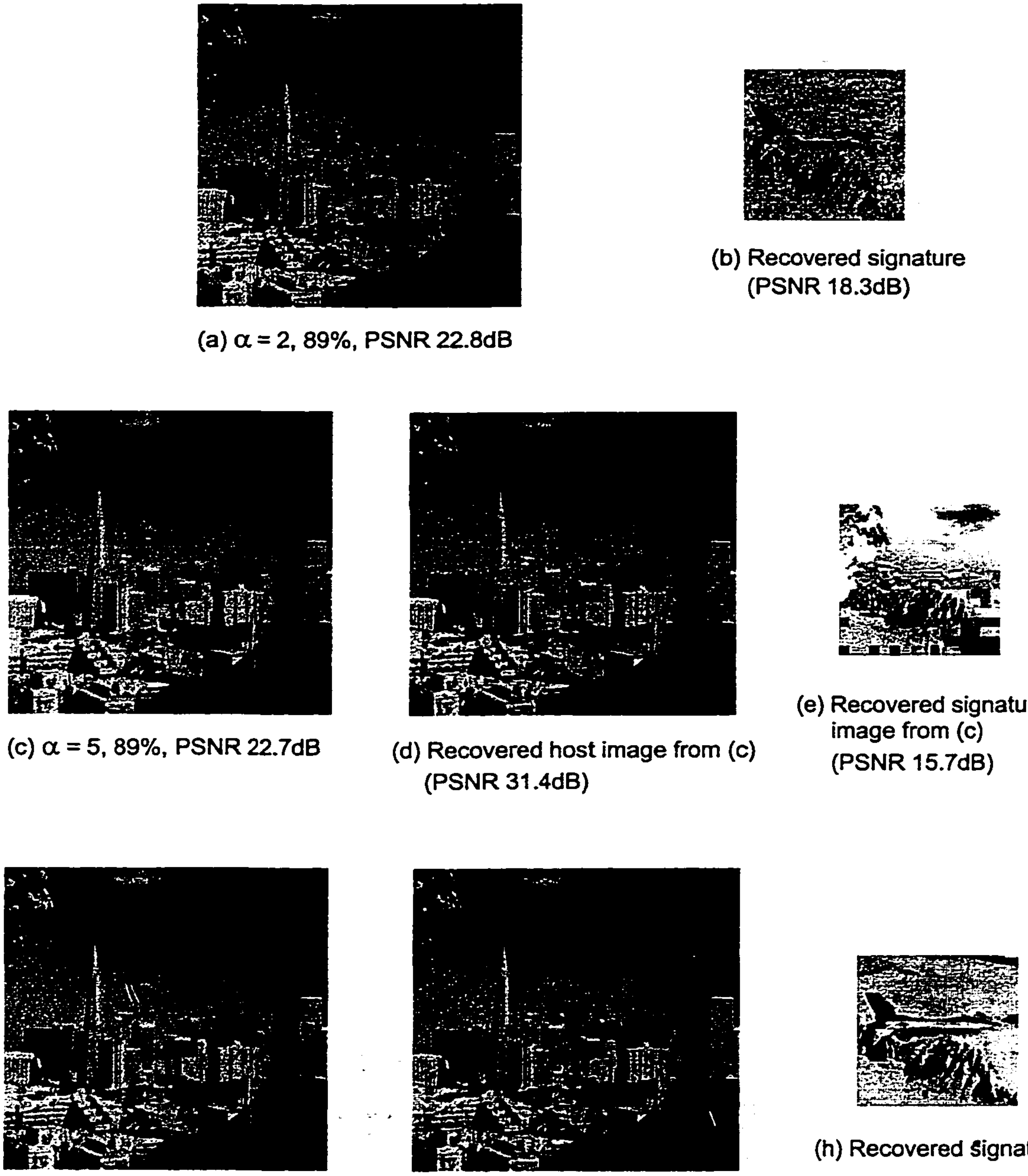
FIGURE 6-7. A schematic for extracting hidden data.

6.4 Experimental Results and Discussions

The following experiments demonstrate the superiority of the MVDCT embedding procedure compared to the methods described in the previous chapters. Note that the MLDCT and MVDCT methods do not require access to the original host data for signature reconstruction.

For the following experiments we use the pyramid image in Figure 6-5(a) as the host and the airplane image in Figure 6-5(b) as the signature. Furthermore, the size of signature image is one quarter the size of the host image.

Figure 6-8 (a), (b) show the embedding using the MDWT method. For this example, we use the one-dimensional DWT. Both the signature and host images are row-scanned to get the one-dimensional vectors, and are combined using the methodology described in Chapter 3. The image shown in Figure 6-8(a) is the JPEG compressed watermarked image. The recovered signature is shown in Figure 6-8(b) with 18.3 dB of PSNR quality. In MDWT, it is assumed that the original host is available for recovery.



(a) $\alpha = 2$, 89%, PSNR 22.8dB

(b) Recovered signature (PSNR 18.3dB)

(c) $\alpha = 5$, 89%, PSNR 22.7dB

(d) Recovered host image from (c) (PSNR 31.4dB)

(e) Recovered signature image from (c) (PSNR 15.7dB)

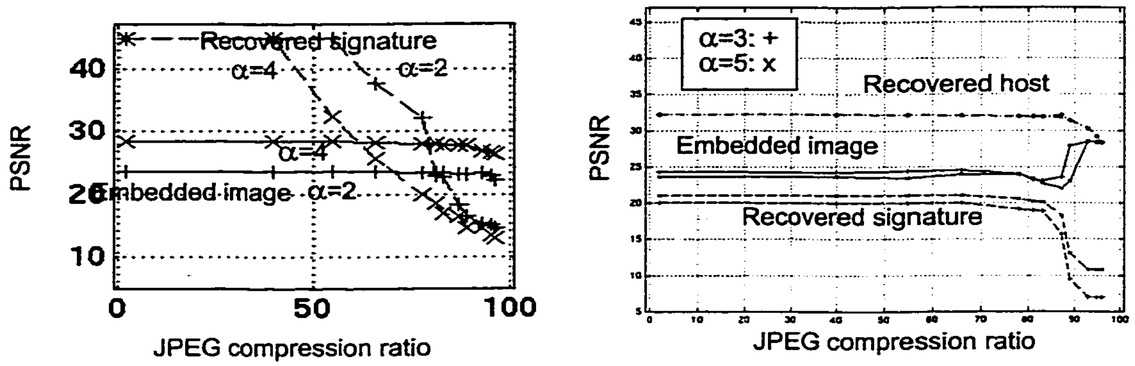
(f) $\alpha = 5$, 89%, PSNR 22.9 dB)

(g) Recovered host image from (f) (PSNR 31.5 dB)

(h) Recovered Signature

(PSNR 27.5 dB)

FIGURE 6-8. Examples of embedded and recovered signature image using (a), (b) the MDWT method, (c)-(e) the MLDCT method, and (f)-(h) the MVDCT method.



(a) PSNR results for MDWT.

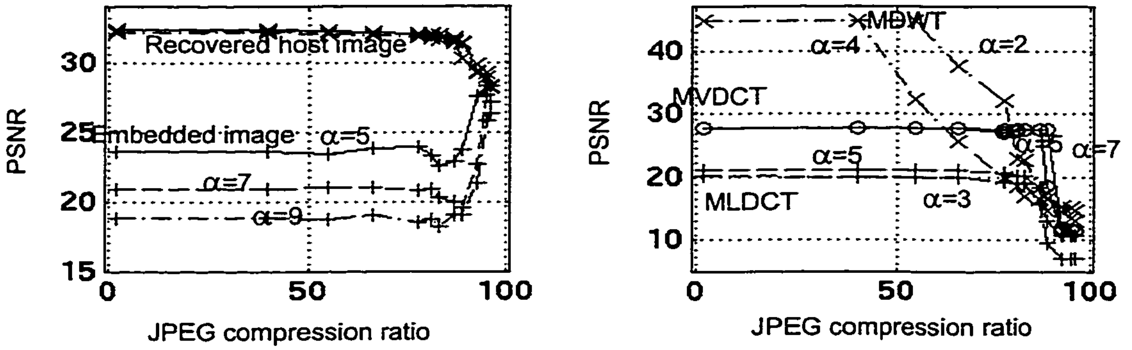
(b) PSNR results for MLDWT.

FIGURE 6-9. PSNR results for varying JPEG compression. (a) results for the MDWT, (b) results for the MLDCT.

Figure 6-8(c) shows the embedded and compressed image using MLDCT. The scale factor is chosen such that the PSNR of the embedded signal is approximately the same in all the cases described here. The MLDCT method does not require the original host. The reconstructed host and signature data are shown in Figure 6-8(d) and (e). The reconstructed signature image has a PSNR of approximately 16 dB, which is worse than the results of MDWT because the lattice coder cannot recover highly degraded signals. Some portions of the reconstructed image show the blocking artifact.

Figure 6-8(f)-(i) show the result of embedding and reconstruction using the MVDCT method. Similar to the MLDCT, the original host image is not needed. Notice, however, the significant improvement in quality for the recovered signature in Figure 6-8(i). The quality of the recovered host images from MLDCT and MVDCT are almost the same. The quality of the signature from the MVDCT method is 27.5 dB, which is more then 10 dB increase from the MLDCT method.

Figure 6-9 (a) and (b) show the PSNR quality relation of the embedded host and recovered signature as a function of the JPEG compression factor for the MDWT and MLDCT methods, respectively. Results for the two different values of the scale parameter alpha are



(a) PSNRs of recovered host image and embedded image

(b) PSNRs of recovered signature image

FIGURE 6-10. PSNR results for the MVDCT at varying JPEG compression rates.

shown. In order to compare the quality of the reconstructed signature image, we used embedded images of similar quality (about 23 dB PSNR). However, the quality of the recovered signatures were very different.

In Figure 6-9(a), the quality of the recovered signature is very high for low JPEG compression rates, and degrades rather quickly at higher compression rates. Recall that the basic embedding idea of the MDWT method is to simply merge coefficients from host and signature images. In the recovery process, the recovered signature coefficients are simply subtracted from the embedded coefficients of the original host. Figure 6-9(b) also shows the PSNR for the recovered host and signature using the MLDCT method. The quality of the recovered signature depends on the quantization levels in MLDCT. Compared to the MDWT, the MLDCT technique demonstrates much better quality reconstruction for up to 85% JPEG compression.

Figure 6-10(a) shows the PSNR of the embedded image and the recovered host using MVDCT. Since the host recovery steps are identical to MLDCT, the PSNR for the recovered host is almost identical to the MLDCT. However, note the improvement in the signature image quality, measured in PSNR, in Figure 6-10(b). There is an improvement of about 7-

8dB on average compared to the MLDCT method. Beyond 80% JPEG compression MVDCT appears to provide better quality of the reconstructed signature images compared to the MDWT as well.

6.5 Summary

We have presented a robust scheme for data hiding using the vector embedding technique on the orthogonal wavelet coefficients vector spaces. This scheme presents a framework for a more robust and adaptive digital watermarking scheme, aimed at hiding large amounts of data into a host. In the next Chapter, we will explore potential application of the MVDCT method to lossless data hiding and hiding in MPEG coded video.

Chapter 7

Applications

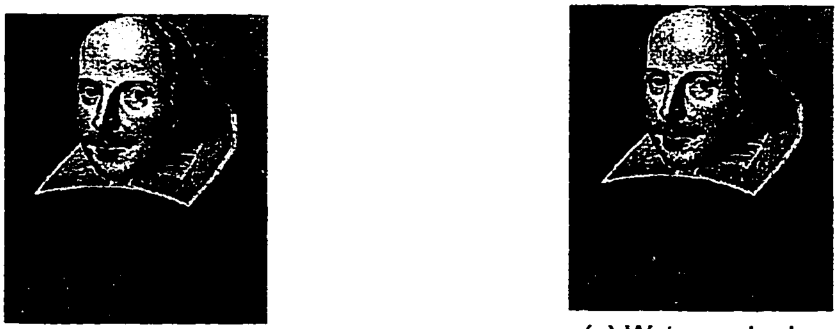
We present two applications of data hiding in this chapter. The first one demonstrates that lossless recovery of the signature is possible, even from lossy compressed images. In this example we embed text (ASCII) data into an image and recover it even when the embedded image is lossy compressed. To the best of our knowledge this is the first time that lossless recovery of large signature data from lossy compressed images is demonstrated. In the second application we demonstrate video-in-video embedding that is robust to motion compensated coding. The chapter concludes with discussions on a possible application to image quality measurement.

7.1 Lossless Data Hiding

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

> Steganography is the art and science of communicating in a way which hides the existence of the communication. In contrast to cryptography, where the "enemy" is allowed to detect, intercept and modify messages without being able to violate certain security premises guaranteed by a cryptosystem, the goal of steganography is to hide messages inside other "harmless" messages in a way that does not allow any "enemy" to even detect that there is a second secret message present [Markus Kuhn 1995-07-03].

(a) A text message



(b) Host image (from [5])

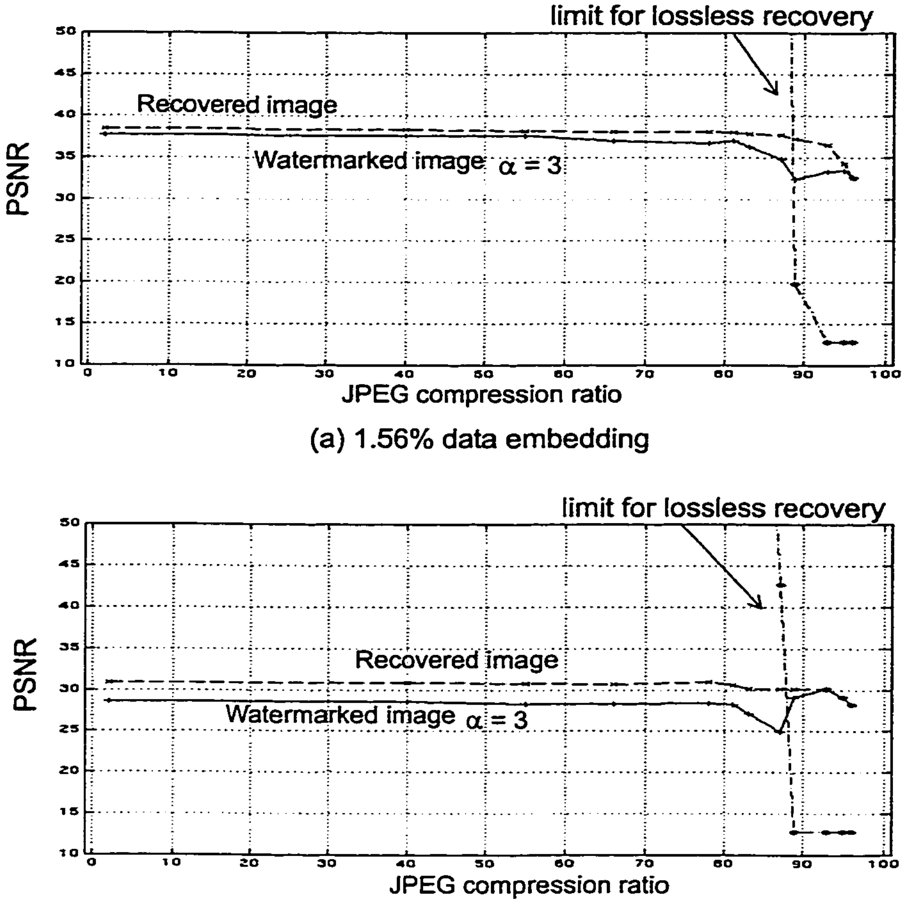
(c) Watermarked and compressed image (80% JPEG)

FIGURE 7-1. Text message embedded in an image.

7.1.1 Hiding Text Data in Images

Hiding text in images is useful in secure communications. Applications include encrypted e-mail, adding text metadata in images, and captions in images and video. While there are many commercial and shareware software packages that offer this functionality, all of them require that the watermarked image remains intact. The primary reason for this is that almost all these methods depend on encoding the information in the Least Significant Bit (LSB), and hence are very susceptible to simple compression schemes, such as the JPEG.

The MLDCT method described in the previous chapter can be easily extended to hide text or control data in images and video. Figure 7-1 shows an example where the text in



(a) 1.56% data embedding

(b) 6.25% data embedding

FIGURE 7-2. PSNR results for the embedding example of Figure 7-1.

Figure 7-1(a) is hidden into the host image of Figure 7-1(b). The alpha-numeric characters of the text are transformed to E_8 lattice codes and are then embedded using the MLDCT method. In this embedding, one byte of signature data is inserted in each 8x8 block of the host image coefficients. The embedded image, compressed by 80% using JPEG is shown in Figure 7-1(c). One can recover the original text without loss at up to 90% JPEG compression.

Figure 7-2(a) shows a plot of the PSNR of the embedded and recovered host image of Figure 7-1(b) as a function of JPEG compression factor. We used a scale factor of $\alpha = 3$ in these experiments. For this α and for the host image of Figure 7-1(b), one can obtain lossless

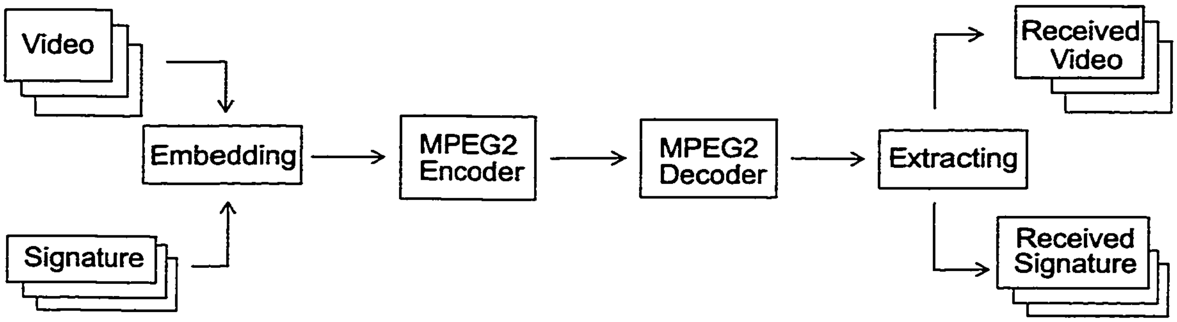


FIGURE 7-3. Schematic of video embedding technique

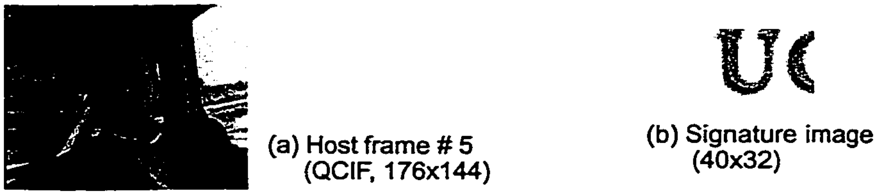
recovery of the text for up to 90% JPEG compression. Figure 7-2(b) shows a similar curve for embedding four bytes of signature data in every 8x8 block of host data. Predictably, the higher embedding rate is at the expense of lower recovery performance - in this case the limit is slightly below 90% JPEG compression. Note that the PSNR drops rapidly beyond the limit for lossless recovery.

7.2 Images and Video in Video

While there is much published work on video watermarking [54-56,80,83,95,96], very few address video data hiding. For example, Swanson and Tewfik describe hiding video in video [109-114]. Their algorithm can hide 2 bits per one 8x8 block. This embedding rate is typical of most video watermarking methods where the data hiding rate is about 0.5-1% of the host data.

Video watermarking techniques are straight-forward extensions of image watermarking techniques (see Figure 7-3). Each frame of the signature video can be embedded in the corresponding frame of the host video. This approach, though, may not be robust to motion compensated coding because of the frame-based embedding scheme. In many cases, one has access to coded video streams rather than the individual frames, and techniques that can insert the data directly into the compressed bit streams are of interest.

In the following we assume a YUV color space representation [24,26,83,109-114]. This facilitates a simple extension from images to digital video, such as those in the MPEG for-



(a) Host frame # 5
(QCIF, 176x144)

(b) Signature image
(40x32)

FIGURE 7-4. Test data. (a) The Y component of car QCIF video frame 5. (b) A signature image

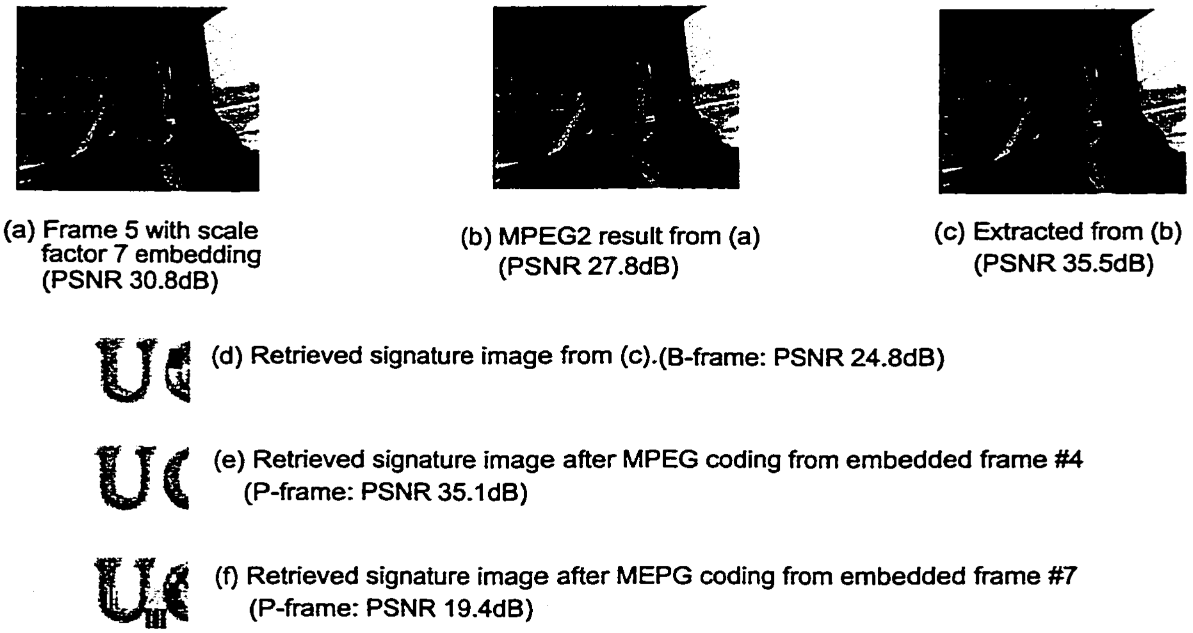
mat. Note that the chrominance components are spatially down-sampled by a factor of two in JPEG and MPEG.

In Figure 7-3, the signature video frame is first embedded into the host video frame, and the embedded video is subjected to MPEG2 coding. Embedded signature recovery follows an inverse sequence of operations. The MPEG stream is first decoded into its corresponding image frames. Then the signature is recovered for these individual frames. As we will demonstrate in the following, the MVDCT is quite robust to MPEG compression.

7.2.1 Images in Video

First, we consider hiding a still image in a video sequence using MLDCT (see Chapter 5). Figure 7-4(a) shows one frame of a QCIF resolution (176 x 144 pixels) video. Figure 7-4(b) shows the still image to be embedded, which is about 1/16-th the size of the video frame. The compressed video is at 550K bps with 30 frames/second.

Consider embedding a still image signature in each of the frames of the video. Then the embedded frames are compressed using MPEG-2 at 600K bps. Figure 7-5(a) shows the frame after embedding and Figure 7-5(b) is the result after the MPEG-2 encoding. Figure 7-5(c) shows the recovered video frame. Since I-frames in the MPEG-2 sequence are JPEG compressed, the results would be similar to image-to-image embedding described in the previous chapters. The frame #5 shown is a B frame. The reconstructed signature from this B frame is shown in Figure 7-5(d). The reconstructed signature image from two P frames are shown in Figure 7-5(e) and Figure 7-5(f). Figure 7-6 shows the PSNR of the compressed host, embedded host, recovered host video, and recovered signature image. In general, it appears that the



(a) Frame 5 with scale factor 7 embedding (PSNR 30.8dB)

(b) MPEG2 result from (a) (PSNR 27.8dB)

(c) Extracted from (b) (PSNR 35.5dB)

(d) Retrieved signature image from (c).(B-frame: PSNR 24.8dB)

(e) Retrieved signature image after MPEG coding from embedded frame #4 (P-frame: PSNR 35.1dB)

(f) Retrieved signature image after MEPG coding from embedded frame #7 (P-frame: PSNR 19.4dB)

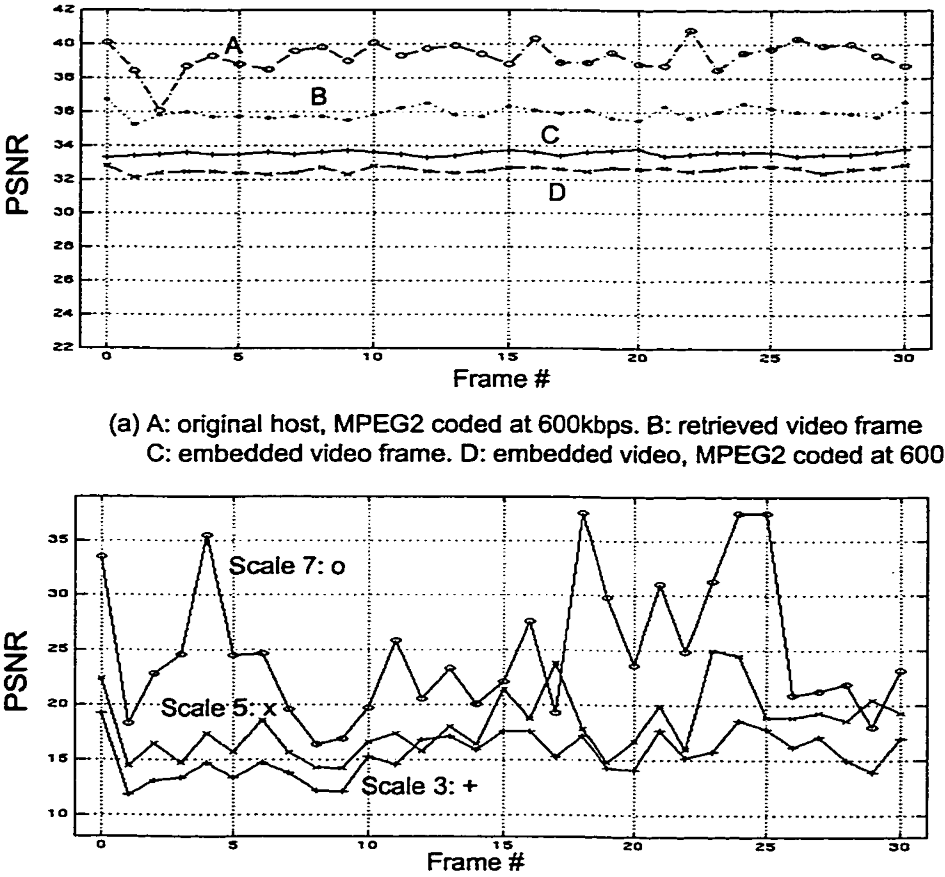
FIGURE 7-5. Embedding a still image in MPEG video frames.

MLDCT method is robust to motion compensation for applications that need embedding still image data into video.

7.2.2 Video in Video

The MLDCT and MVDCT methods can be easily extended to embedding video in video. The data embedding rate is 1/16th as before. One can embed, for example, a frame of similar resolution as the host video in every 16 frames of the host.

Figure 7-7(a) shows a host video frame and Figure 7-7(b) a frame from the signature video. Host and signature video are from the MPEG-7 video data collection (CD reference number #16. The host frames are from cm1002.02500 to cm1002.02800 and the signature frames are from cm1002.11700 to cm1002.11750.) The size of the video frames are 352x240 pixels. Each signature frame is divided into 16 blocks. Each block is then embedded into one host frame. We use a compressed bit rate of 2Mbps, which corresponds to the MPEG recom-



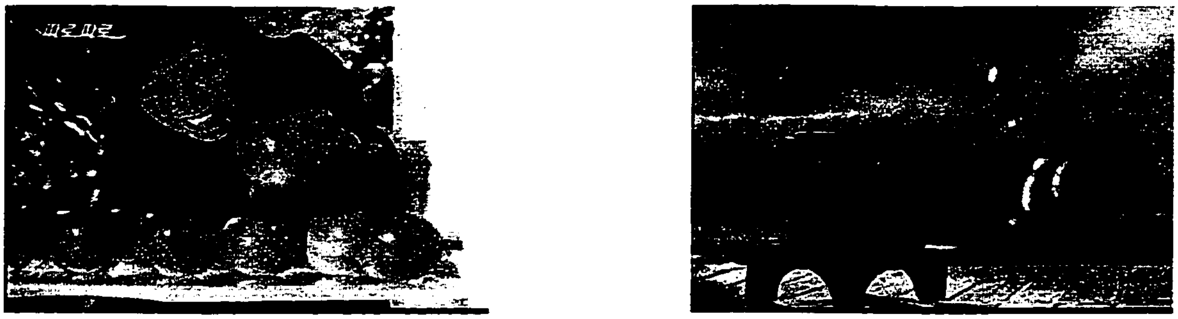
(a) A: original host, MPEG2 coded at 600kbps. B: retrieved video frame
C: embedded video frame. D: embedded video, MPEG2 coded at 600Kbps.

(b) PSNRs of retrieved signature image

FIGURE 7-6. PSNR of host video and recovered signature image

mended bit rate for the quarter resolution NTSC video at 30 frames per second (Note. for the full NTSC resolution this corresponds to 5Mbps.)

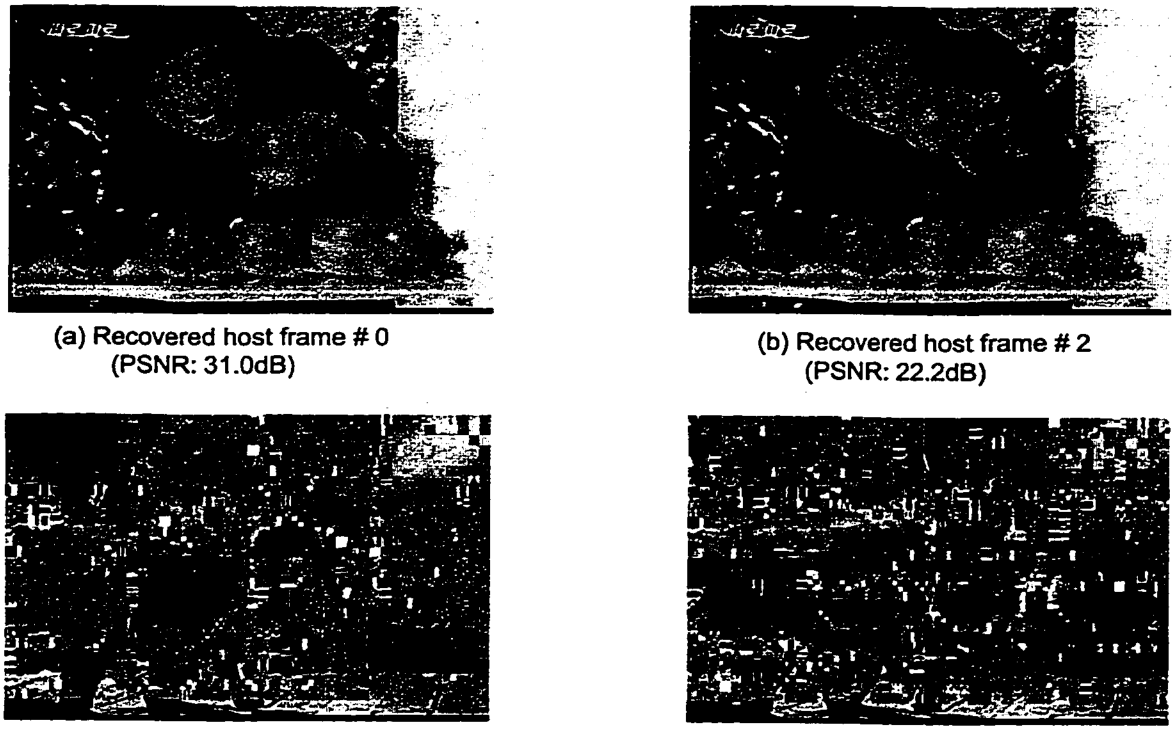
Even though the MLDCT method works reasonably well for hiding still images in video, the results for hiding video signatures in video are not quite satisfactory, as the following example demonstrates. Consider Figure 7-8. Figure 7-8(a) and (b) show the recovered host frames after MPEG2 coding from the watermarked frame # 0 (frame #cm1002.02500) and frame # 2 (cm1002.02502). Figure 7-8(c) and (d) show the recovered signature frame # 0 (exactly, cm1002.11700) and frame # 1 (cm1002.11701) from each of the sixteen recovered embedded host frames. The PSNRs of the recovered signature video frames are 14.5dB and



(a) Host video frame # 0
(size: 352x240)

(b) Signature video frame # 0
(size: 352x240)

FIGURE 7-7. test data. (a) The Y component of a video frame. (b) The Y component of a signature frame.



(a) Recovered host frame # 0
(PSNR: 31.0dB)

(b) Recovered host frame # 2
(PSNR: 22.2dB)

(c) Recovered signature frame # 0
(PSNR: 14.5dB)

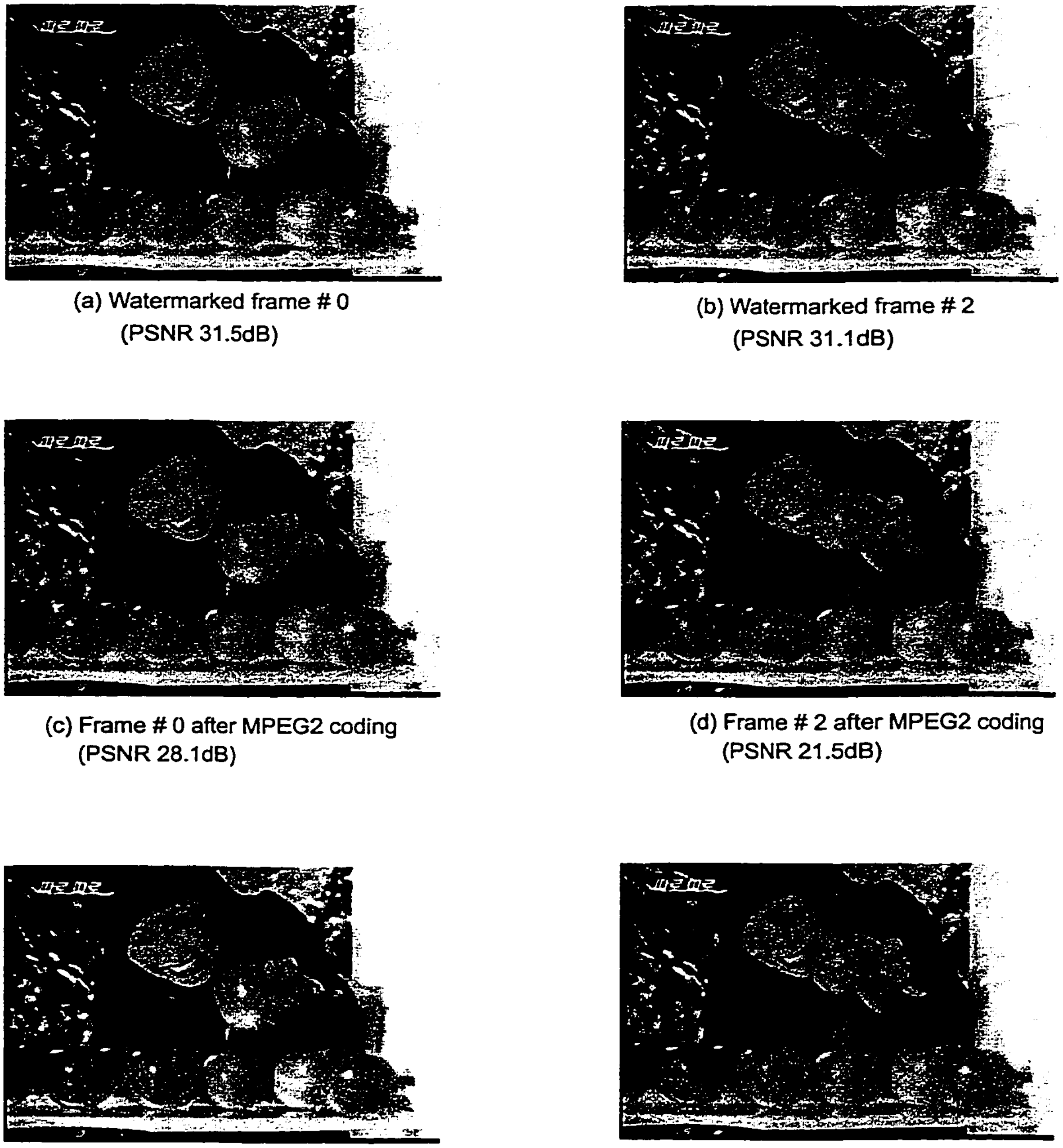
(d) Recovered signature frame # 2
(PSNR: 14.2dB)

FIGURE 7-8. Results using the MLDCT algorithm. (a), (b) Recovered host frames. (c), (d) recovered signature frame # 0 and frame # 2.

14.2dB, respectively. Notice the difference in quality between the I-frame embedded and P-frame embedded blocks. For example, the top-left block of the Figure 7-8(c) is of good quality because this block is embedded in an I-frame in the MPEG 2 sequence. Blocks embedded in P- or B-frames show significant amount of visible noise in the reconstructed image, making them practically useless.

However, the MVDCT method for this video in video embedding offers significantly improved results. Figure 7-9 shows some examples of using the MVDCT method. The watermarked video frame # 0 and frame # 2 are shown in Figure 7-9(a) and (b), respectively, with scale factor $\delta = 5$. Figure 7-9(c) and (d) show its degradation by MPEG2 coding from Figure 7-9 (a), (b), respectively. Figure 7-9(e) and (f) show the recovered video frames from Figure 7-9 (c), (d), with a PSNR of 35.7dB and 31.7dB, respectively. The quality of recovered host frames of MVDCT are similar to that of MLDCT method, as is to be expected. Compared to the MLDCT, the quality of the recovered video frames of MVDCT are much improved, as shown in Figure 7-10. In this experiment, the MPEG-2 bit rate is 2M bps, 30 frames/second. Figure 7-10(a) and (d) show the original and recovered β-Signals. The recovered signature frame blocks are shown in Figure 7-10(e)-(f). These reconstructed signature images have PSNRs of around 45 dBs. It is observed that, in general, MVDCT offers stable and robust embedding in I-, P- and B-frames in MPEG-2 sequences.

Figure 7-11(a) shows the PSNR of the original MPEG compressed host video and the watermarked frames for the MLDCT method for the first 32 frames of the video. It also shows the PSNR of the MPEG compressed embedded video and the recovered host video. The plot 'o' shows the PSNRs of the MPEG coded original frames without embedding. The plot 'x' shows the PSNRs of the watermarked frames and their MPEG decoded frames are shown in the plot '+' at the bottom. The plot ' · ' shows the PSNRs of the extracted host frames. Figure 7-11(b) shows the corresponding PSNR for the MVDCT method. As mentioned above, the PSNR of the recovered and watermarked hosts are very similar to Figure 7-11(a). Figure 7-11(c) shows the PSNRs of the recovered signature video of the MVDCT and MLDCT methods. These show about 30dB PSNR difference between the two methods.



(a) Watermarked frame # 0
(PSNR 31.5dB)

(b) Watermarked frame # 2
(PSNR 31.1dB)

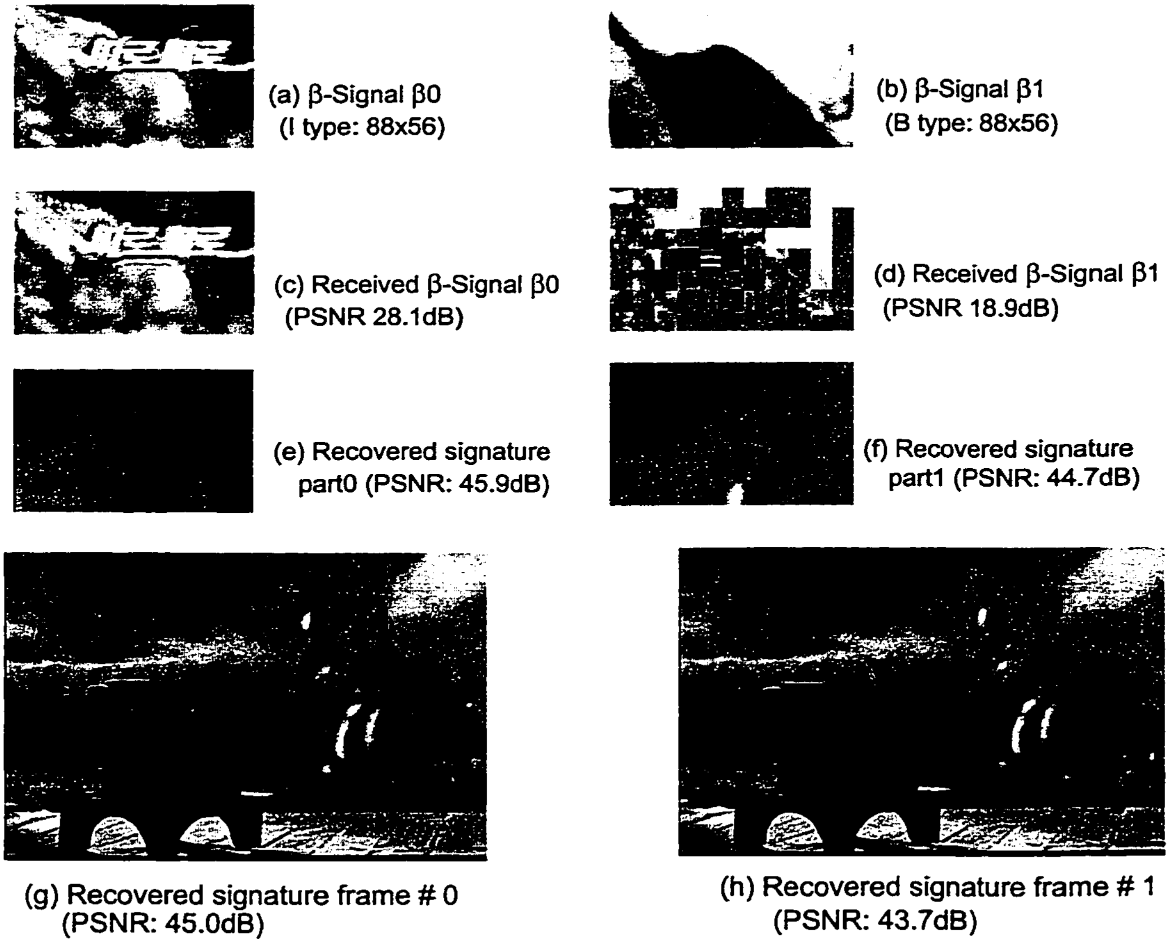
(c) Frame # 0 after MPEG2 coding
(PSNR 28.1dB)

(d) Frame # 2 after MPEG2 coding
(PSNR 21.5dB)

(e) Recovered frame # 0
(PSNR 35.7dB)

(f) Recovered frame # 2
(PSNR 31.7dB)

FIGURE 7-9. Host video recovery: embedding scheme used is MVDCT. (a), (b) Watermarked frames. (c), (d) Embedding frames after MPEG2 compression at 2M bps. (e), (f) Recovered frames from (c), (d), respectively.



(a) β-Signal β0 (I type: 88x56)

(b) β-Signal β1 (B type: 88x56)

(c) Received β-Signal β0 (PSNR 28.1dB)

(d) Received β-Signal β1 (PSNR 18.9dB)

(e) Recovered signature part0 (PSNR: 45.9dB)

(f) Recovered signature part1 (PSNR: 44.7dB)

(g) Recovered signature frame # 0 (PSNR: 45.0dB)

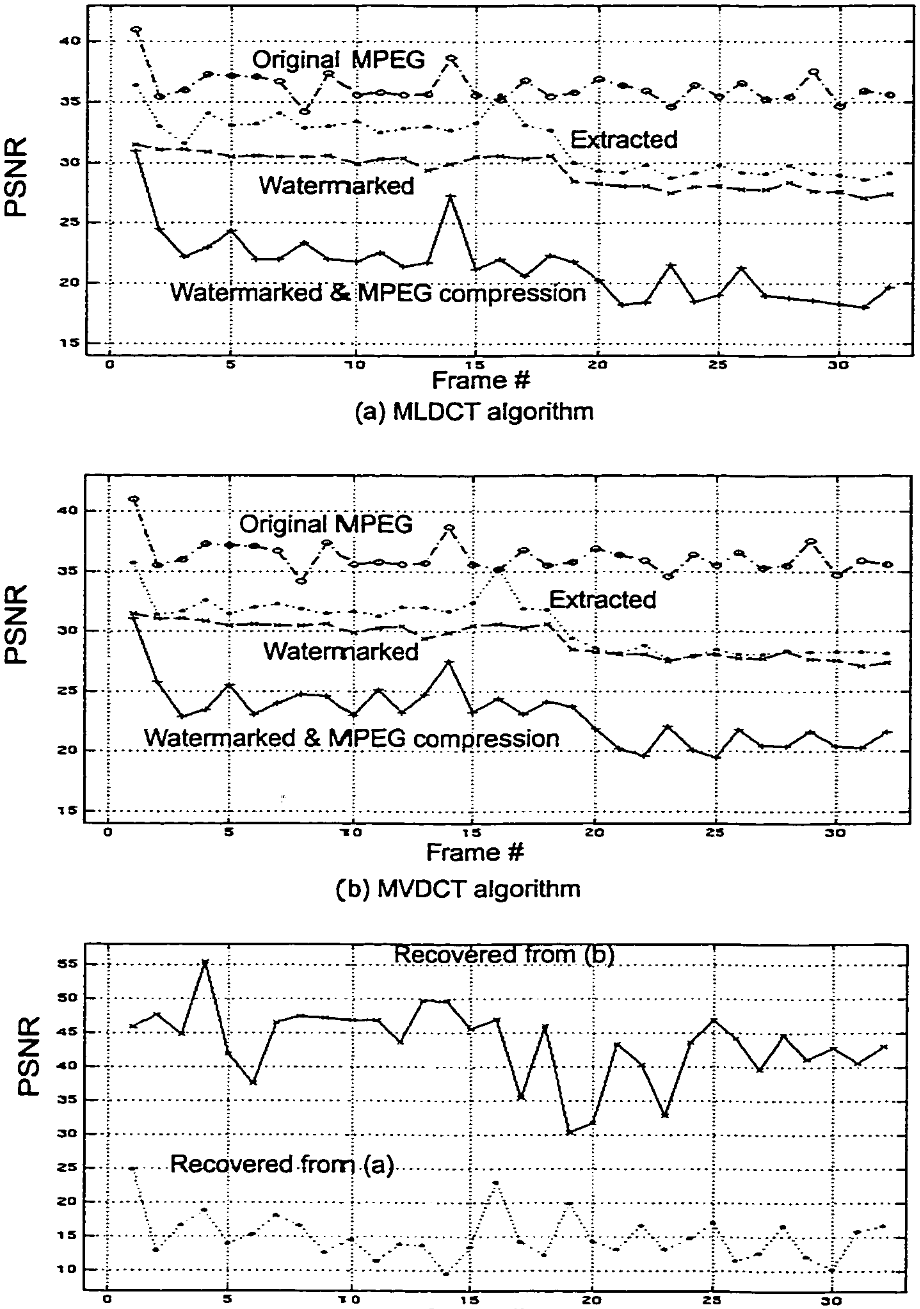
(h) Recovered signature frame # 1 (PSNR: 43.7dB)

FIGURE 7-10. Signature video recovery. (a), (b) β-Signals. (c), (d) Received β-Signals. (e), (f) Recovered signatures. (g), (h) Whole recovered signature frame # 0 and frame # 2.

7.3 Discussions

7.3.1 Image Quality Estimation

Considerable work has been done on measuring the *quality* of images and video [119]. Many of these are based on models of human visual system. The basic problem that is addressed is, given the original and the degraded images, what is the metric that best captures the perceptual degradation. It is well known that simple metrics such as the mean-squared



(a) MLDCT algorithm

(b) MVDCT algorithm

(c) Recovered signature

FIGURE 7-11. (a) PSNRs of the host video frames with MLDCT algorithm. (b) PSNRs of the host frames with MVDCT algorithm. (c) PSNRs of recovered signature using MLDCT and MVDCT methods.

error are not good for quantifying perceptual distortion. One promising measure in this context is the just noticeable difference, proposed in [94].

One can use the lossless embedding method described in this chapter to include information regarding the original image such that, at the receiver, one can "estimate" some measure of the received image quality. Note that we are not interested in defining what this quality measure is or how it is to be computed. However, the user, by choosing the data to embed, can embed enough information to compute the performance metrics of his/her choice. The following presents some preliminary experimental results.

The following assumptions are made: image pixels are sampled such that statistics computed from these pixels are representative of the overall image properties. For the experiment, we randomly choose one pixel from every other 8x8 block in the image, say the black blocks in a checker board pattern of blocks, alternately colored as black and white. The pixel intensity is then embedded in the preceding white block, similar to embedding text in the previous section. This embedded signal then forms the "signature" of the original image.

Since we can recover this signature without error, this can be used to estimate the PSNR of the received image. Since the white blocks are not changed, if the image is not modified then the reconstructed host samples forming the signature are identical to the recovered hidden signature. Again we assume that any distortion of the embedded image is spread uniformly over the entire data. In particular, we assume that the degradation does not selectively affect the specific set of pixel samples used to form the signature signal. Furthermore, if the intention is to find out if any pixel in the image has changed, one can form the signature in a different way that could contain sufficient information about the image so as to find out any pixel modifications.

Figure 7-12 shows the plot of the estimated PSNR as a function of JPEG compression. The curve A is for the measurements comparing the recovered signature and recovered host. In practice, this is the only estimate that we can compute. Curve B is the PSNR computed using the recovered host and the original host. Ideally, this is what we would *like* to compute. The curves shown are average performance characteristics, averaged over 30 different images. Beyond 85%, on the average, the signature recovered is no longer guaranteed to be

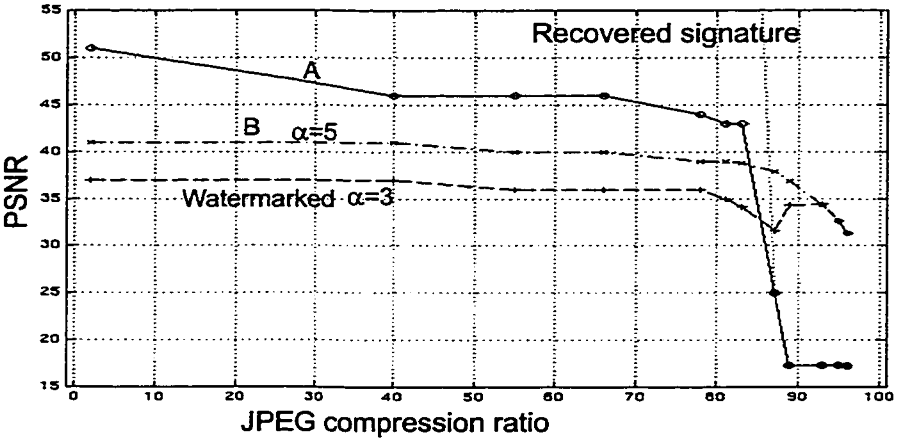


FIGURE 7-12. PSNR vs. JPEG compression. Curve A is for comparing recovered signature with recovered host. Curve B is for recovered host and original host.

lossless. This can be see by a sudden dip in the PSNR beyond about 85% compression. We can make the following general observations:

1. one can detect whether the image has been changed or not. This is quite simple
2. if the image has changed, but the change is not significant, it is hard to quantify the degradation using PSNR. This is true for compressions of up to 80% in the above experiment.
3. if the image has undergone severe compression (>85%), it is possible to detect such changes, even if the reconstructed image is of good perceptual quality.

The existing literature on image quality estimation invariably makes the assumption that one has access to the original data. This may not be the case in most practical broadcast applications, and may be expensive to make the original data available in other applications as well. Lossless data hiding may be an alternative approach to consider for quality estimation. However, much more work remains to be done in developing a practical methodology for reliably estimating the image quality.

7.3.2 Summary

In conclusion, we note that the methods developed in Chapters 3-6 enable new application domains in multimedia processing. In addition to applications to watermarking and steganography, the proposed methods have potential applications to multimedia data control and in the emerging MPEG standards such as the MPEG-4 and MPEG-7. The easy of access to and manipulation of content makes it all the more important that the multimedia content be protected from unauthorized uses. MPEG has stared working on a new standard for multimedia that enables content to be searched for and delivered based on usage rights, and data hiding technologies such as the ones described in this dissertation are very useful for such applications.

Chapter 8

Conclusions and Future Work

8.1 Conclusions

In this research we have developed several new techniques for robust hiding of image and video data. These techniques enable embedding large amounts of data and facilitate signal recovery in the absence of the original host. The proposed methods include the MDWT method (Chapter 3) that uses an extended spread spectrum technique in the discrete wavelet transform domain, and the MLDWT (Chapter 4) method that uses a channel/source coding technique. The MLDCT (Chapter 5) method enables hidden data extraction without additional host information, and the MVDCT (Chapter 6) method uses vector embedding that is robust to motion compensated video coding. Two interesting applications of these embedding methods to lossless text data recovery from lossy compressed images, and video-in-video hiding, are presented in Chapter 7.

8.2 Summary of Contributions

One of the main contributions of this thesis is the development of methodologies for large quantity data embedding. The results presented here are among the first to report gray scale image hiding in images. As the results demonstrate, the signature recovery is quite robust to DCT and wavelet based image compression. Since these techniques enable large quantity data embedding, one can achieve robustness to image manipulations for watermarking related applications by having redundancy in the signature data.

In Chapter 3, we presented an extended spread-spectrum technique, the MDWT, that distributes the message data in the wavelet sub-bands. Using this method, one can embed images which are up to 25% of the host data size. This method is quite robust to JPEG and wavelet compression.

Chapter 4 extends the wavelet-based embedding technique by first encoding the signature data using lattice codes (MLDWT). The lattice codes add error resilience to signature recovery and offers a structured embedding methodology.

In Chapter 5, we present the MLDCT algorithm which combines lattice encoded signature coefficients and an adaptive texture masking that embeds data using the DCT. Compared to MDWT and MLDWT, this method does not require the original host to recover the hidden data. Methods that do not require the original host are very desirable in applications such as hidden communications. This method is also shown to be quite robust to JPEG compression.

Chapter 6 extends the wavelet-based embedding in a new direction. In this method, which is referred to as the MVDCT, a new signal is derived from the signature and host images. This new signal, called the β-Signal, is then inserted into the host using the MLDCT. The β-Signal captures the non-redundant data that needs to be inserted into the host. This enables high quality embedding and recovery that is resistant to both JPEG and MPEG compression.

In Chapter 7, we demonstrate applications using MVDCT for lossless data hiding. We consider the example of hiding a text message into an image, and the message is recovered without error even when the embedded image is lossy compressed. This is among the first demonstrations of lossless message recovery from lossy (compressed) embedded data. Furthermore, we demonstrate that the MVDCT method is robust to motion compensated coding. This is illustrated by hiding images and video in video, which is then MPEG compressed.

In summary, the methods presented in this dissertation advance the current data hiding technology both in terms of the quantity of the data that can be hidden (up to 25% compared to 1% reported in the literature), and the quality of the embedded and recovered data even under significant JPEG/MPEG compression (of up to 90% in some cases).

8.3 Directions for Further Research

8.3.1 Robustness to Signal Manipulation

The primary emphasis in this dissertation is on large quantity data embedding that is robust to data compression. There is a trade-off between the quantity of the data that can be embedded and the robustness of the hidden data to signal processing. In digital watermarking for authentication, intentional or unintentional attacks may include, in addition to signal compression, scaling, cropping, rotation of images, and digital-to-analog and analog-to-digital conversions. No single technique can be resistant to all these attacks simultaneously. However, the methods proposed in this dissertation have the advantage of embedding large amounts of data. Since watermarks typically require very few bits compared to the host data size, one may be able to distribute these bits intelligently so that the embedded data is resistant to specific attacks other than compression. This needs further investigation.

D/A and A/D Conversions: Authentication of printed documents is an interesting problem that has not received much attention. This is especially important in detecting forgery (for example, printing of currency notes). With the availability of inexpensive yet very high quality printers, it is quite easy to forge such documents to get by casual inspections. In the earlier days, expensive devices costing thousands of dollars were installed in high-end color copiers to prevent forging of bank notes. The current ink jet printers cost few hundred dollars and are capable of producing even better quality prints. Watermarks that are robust to D/A and A/D conversions are of much interest in this context. Hiding data in printed text documents is another interesting area for further research.

Printing and scanning often result in scaling and/or rotation of the images, in addition to the D/A and A/D artifacts. One possibility is intensity modulation to encode signature data. Another possibility is the use of shape modulation of printed text to embed data.

Geometric Distortions: · Pixel based methods are, in general, not quite robust to geometric modifications such as scaling, cropping, and rotation. Scaling and rotation affect the pixel values due to interpolation and the embedding schemes need to be resistant to such transformations. One possibility is to use adaptive region based encoding wherein the signature information is duplicated at many salient image locations.

8.3.2 Applications

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

Image and video quality control is a very important application domain. We presented some preliminary results in Chapter 7 on estimating the image quality. We are not aware of any related work in image quality measurements that are based only on the received signal. The thought of a "black box" that can automatically compute the quality of the received signal without any reference signal is very appealing in many applications, including broadcasting. Data hiding can perhaps be further extended to create "smart multimedia" that can self-correct itself under modifications using embedded control information.

Other potential applications include multimedia databases where the objects in the database "contain" self-information that could be used in navigating the database, or in providing different levels of access to the users depending on the service that is requested. For example, object based representations (using region masks) could be embedded into the video stream that would enable object based functionality using existing video data formats such as MPEG-2. While many standard bit streams allow for header information where such control data bits could be stored, embedding the control data in the host data stream has the advantage that it can not be accidentally or otherwise stripped off of the host data. Recently, the MPEG has started working on developing a new standard - MPEG-21 "Multimedia Framework". The scope of the standard can be described *as the integration of two critical technologies: how consumers can search for and get content - by themselves or through the use of intelligent agents - and how content can be decoded for consumption according to usage rights associated with the content* (quoted from a web paper by Leonardo Chiariglione, Convener, MPEG; http://www.cselt.it/leonardo/paper/technoreview99/index.htm). We believe that the technologies developed in this dissertation would enable such applications.

References

[1] Jpeg- V4, ftp://ftp.funet.fi/pub/crypt/steganography

[2] MPEG2 CODEC program Ver. 1.2, http://www.mpeg.org/MSSG, July, 1996.

[3] MPEG Official WWW, "http://www.mpeg.org"

[4] PictureMarc, *Digimarc*, http://www.digimarc.com

[5] Steganography on line web link, http://www.jjtc.com/Steganography/

[6] StegoDos, *Block Wolf's Picture Encoder v0.9B*, ftp://ftp.csua.berkeley.edu/pub/cypherpunks/steganography/stegodos.zip

[7] SureSign, Signum Technologies, http://www.signumtech.com

[8] Watermarking and Data hiding on line web link, http://www-nt.e-technik.uni-erlangen.de/~hartung/watermarkinglinks.html

[9] Watermarking on line web link, http://ltssg3.epfl.ch:1248/kutter/watermarking/wetlinks.html

[10] UCSB Official WWW, "http://www.ucsb.edu"

[11] B. Alpert, G. Beylkin, R. Coifman and V. Rokhlin, "Wavelet-like Bases for the Fast Solution of Second-kind Integral Equations," *SIAM Journal of Science Computations*, Vol. 14, no. 1, pp. 159-184, January 1993.

[12] R. Anderson, *Lecture Notes in Computer Science, Vol. 1174, Information Hiding, First International Workshop*, Springer, Berlin, Germany, 1996.

[13] R, Arachelian, White Noise Strom (program source), ftp://ftp.csua.berkerley.edu/pub/cypherpunks/steganography/wms210.zip

[14] D. Aucsmith, *Lecture Notes in Computer Science vol. 1525: Information Hiding, Second International Workshop*, Springer, Berlin, Germany, 1998.

[15] F. Bartolini, M. Barni, V. Cappellini and A. Piva, "Mask Building for Perceptually Hiding Frequency Embedded Watermarks," *Proceeding of IEEE International Conference of the Image Processing,* Vol. 1, pp. 450-454, Chicago, Illinois, October, 1998,

[16] W. Bender, D. Gruhl and N. Morimoto, "Techniques for Data Hiding," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database III,* vol. 2430, pp.164-173, San Jose, February, 1995

[17] A. G. Bors and I. Pitas, "Image Watermarking using DCT Domain Constraints," *Proceeding of IEEE International Conference of Image Processing,* Vol. III, pp. 231-234, Lausanne, Switzerland, September. 1996.

[18] L. Boney, A. H. Tewfik and K. N. Hamdy, "Digital Watermarks for Audio Signals," *Proceedings of the IEEE International Conference on Multimedia Computing and Systems*, pp. 473-480, Hiroshima, Japan, June 1996.

[19] G. Beylkin, R. COifman and V. Rokhlin, "Fast Wavelet Transforms and Numerical ALgorithms I," Communications on Pure and Applied Mathematics, Vol. 44, pp. 141-183, 1991.

[20] A. Brown, S-Tools (program source), ftp://idea.sec.dsi.unimi.it/pub/security/crypt/code/s-tools2.zip

[21] A. K. Choudhury, N. F. Maxemchuk, S. Paul and H. G. Schuzrinne, "Copyright Protection for Electronic Publishing Over Computer Networks," *IEEE Networks Journal*, pp. 12-20, May/June 1995.

[22] J. J. Chae and B. S. Manjunath, "A Robust Embedded Data from Wavelet Coefficients," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database VI,* Vol. 3312, pp. 308-317, San Jose, February, 1998.

[23] J. J Chae, D. Mukherjee and B. S. Manjunath, "A Robust Data Hiding Technique using Multidimensional Lattices," *Proceedings of the IEEE Forum on Research and Techninology Advances in Digital Libraries,* pp. 319-326, Santa Barbara, April 1998.

[24] J. J. Chae, D. Mukherjee and B. S. Manjunath, "Color Embedding using Lattice Structures," *Proceeding of IEEE International Conference of the Image Processing,* Vol. 1, pp. 460-464, Chicago, Illinois, October, 1998.

[25] J. J. Chae and B. S. Manjunath, "A Technique for Image Data Hiding and Reconstruction without Host Image," *Proceeding of SPIE EI '99, Security and Watermarking of*

Multimedia Contents, Vol. 3657, pp. 386-396, San Jose, California, January, 1999.

[26] J. J. Chae and B. S. Manjunath, "Data Hiding in Video," *Proceeding of IEEE International Conference on Image Processing* , Vol. 1, pp. 311-315, Kobe, Japan, October, 1999.

[27] B. Chen and G. W. Wornell, "Digital Watermarking and Information Embedding Using Dither Modulation," *Proceeding of IEEE second workshop on Multimedia Signal Processing,* pp. 273-278, Los Angeles, California, 1998.

[28] J. H. Conway and N. J. A. Sloane, "Voronoi Regions of Lattices, Second Moments of Polytopes, and Quantization," *IEEE Trans. Information Theory,* Vol. IT-28, No. 2, pp.211-226, March, 1982.

[29] J. H. Conway and N. J. A. Sloane, "Fast Quantizing and Decoding Algorithms for Lattice Quantizers and Codes," *IEEE Trans. Information Theory,* Vol. IT-28, No. 2, pp. 227-232, March, 1982.

[30] J. H. Conway and N. J. A. Sloane, *Sphere Packings, Lattices and Groups*, Second edition, Springer-Verlag, New York, 1991.

[31] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "A secure Robust watermark for Multimedia," *IEEE Trans. Image Processing,* Vol. 6. no. 12, pp. 1673-1687, December 1997.

[32] I. J. Cox, J. Killian, T. Leighton, and T. Shamoon, "A secure Robust watermark for Multimedia," *Information Hiding, Lecture Notes in Computer Science*, Vol. 1174, pp.183-206, 1996.

[33] I. J. Cox and J. M. G. Linnartz, "Some General Methods for Tampering with Watermarks," *IEEE Journal on Selected Areas in Communications,* Vol. 16. no. 4, pp. 587-593, May, 1998.

[34] S. Craver, N. Memon, B. Yeo, and M. Yeoung, "Can Invisible Watermarks Resolve Rightful Ownership?," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database V,* Vol. 3022, pp.310-321, San Jose, 1997.

[35] S. Craver, N. Memon, B. Yeo, and M. Yeoung, "Resolving Rightful Ownerships with Invisible Watermarking Techniques: Limitations, Attacks, and Implications," *IEEE Journal Areas in Communications,* Vol. 16, no. 4, pp. 573-586, May 1998.

[36] S. Craver, N. Memon, B. Yeo, and M. Yeoung, "Can Invisible Watermarks Resolve Rightful Ownership?," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database V,* Vol. 3022, pp. 310-321, San Jose, 1997.

[37] Z. Cvetkovic and M. Vetterli, "Oversampled Filter Banks," *IEEE Tr. on Signal Processing,* Vol. 46, no. 5, May 1998.

[38] R. Dugar, K. Rataknda and N. Ahuja, "A New Wavelet-Based Scheme for Watermarking Images," *Proceedings of IEEE International Conference of Image Processing,* Vol. 2, pp. 419-423, Chicago, Illinois, October, 19978.

[39] G. Davis, *Program source files (wavelet.0.3.tar.gz),* "http://www.cs.dartmouth.edu/~gdavis/wavelet/wavelet.html"

[40] J. -F. Delaigle, C. De Vleeschouwer and B. Marq, "Digital Watermarking," *Proceeding of SPIE Optical Security and Counterfeit Deterrence Techniques,* Vol. 2659, pp. 99-110, San Jose, February, 1996.

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

[42] N. Farvardin, "A study of Vector Quantization for Noisy Channels," *IEEE Tr. Information Theory,* Vol. 36, no. 4, pp. 799-809, July, 1990.

[43] N. Farvardin and V. Varishampayan, "On the Performance and COmplexity of Channel-Optimized Vector Quantizers," *IEEE Tr. Information Theory,* Vol. 37, no. 1, pp. 155-160, January, 1991.

[44] N. Farvardin, "A study of Vector Quantization for Noisy Channels," *IEEE Tr. Information Theory,* Vol. 36, no. 4, pp. 799-809, July, 1990.

[45] N. Farvardin and V. Varishampayan, "On the Performance and Complexity of Channel-Optimized Vector Quantizers," *IEEE Tr. Information Theory,* Vol. 37, no. 1, pp. 155-160, January, 1991.

[46] D. J. Fleet and D. J. Heeger, "Embedding Invisible Information in Color Image," *Proceeding of IEEE International Conference of Image Processing,* Vol. 1, pp. 532-535, Santa Barbara, October, 1997.

[47] A. Fuldseth, *Robust Subband Video Compression for Noisy Channels with Multilevel*

Signaling - ph.d dissertation, Norwegian University of Science and Technology, Norway, 1997.

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

[49] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*, Addison-Wesley, New York, 1992.

[50] A. A. Giordano and F. M. Hsu, *Least Square Estimation with Applications to Digital Signal Processing*, A Wiley-Interscience Publication, John Wiley and Sons, New York, 1985.

[51] F. Goffin, J. F. Delaigle, C. D. Vleeschouwer, B. Marq and J. -J. Quisquater, "A Low Cost Perceptive Digital Picture Watermarking Method," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database V,* Vol.3022, pp. 264-276, San Jose, February, 1997.

[52] B. G Haskell, A. Puri and A. N. Netravaili, *Digital Video: Introduction to MPEG-2,* Chapman & Hall, New York, 1977

[53] F. Hartung and B. Girod, "Fast Public-key Watermarking of Compressed Video," *Proceeding of IEEE International Conference of Image Processing,* Vol. 3, pp. 528-531, Santa Barbara, California, October, 1997.

[54] F. Hartung and B. Girod, "Watermarking of MPEG-2 encoded video without decoding and re-encoding," *Proceeding of SPIE EI '97, Multimedia Computing and Networking,* Vol. 3020, pp. 264-274, San Jose, California, January, 1997.

[55] F. Hartung and B. Girod, "Digital Watermarking of Raw and Compressed Video," *Proceeding of the SPIE, Digital Compression Technologies and System for Video Communication,* Vol. 2952, pp. 205-213, San Jose, California, February, 1996.

[56] F. Hartung and B. Girod, "Copyright Protection in Video Delivery Networks by Watermarking of Pre-compressed Video," *Springer Lecture notes in Computer Science,* Vol. 1242, pp. 423-436, Springer, February, 1996.

[57] F. Hartung and B. Girod, "Watermarking of MPEG-2 encoded video without decoding and re-encoding," *Proceeding of the SPIE, Digital Compression Technologies and System for Video Communication,* Multimedia Computing and Networking, Vol. 3020, pp. 264-274, San Jose, February, 1996.

[58] C. Hsu and J Wu, "Hidden signatures in Images," *proceeding of IEEE International Conference on image Processing,* Vol. 3, pp.223-226, September, Switzer, 1996.

[59] C. Hsu and J. Wu, "Hidden Digital Watermarks in Images," *IEEE Tr. on Image Processing,* Vol. 8, no. 1, January 1999.

[60] C. Hsu and J Wu, "DCT-based Watermarking for Video," *IEEE TR. on Consumer Electrics,* Vol. 44, no. 1, pp. 206-216, February 1998.

[61] J. Huang and Y. Q. Shi, "Adaptive Image Watermarking Scheme based on visual masking," *IEE Electronics Letters* vol. 34, no. 8, pp. 748-750, April 1998.

[62] F. Idris and S. Panchanathan, "Storage and Retrieval of Compressed Images using Wavelet Vector Qauantization," *Journal of Visual Languages and Computing,* no. 8, pp. 289-301, 1997.

[63] H. Imai, *Essentials of Error-Control Coding Techniques*, Academic Press, New York, 1990.

[64] International Standard, *ISO/IEC 13818-1, Information Technology - General Coding of Moving Picture and Associated Audio Information: systems*, ISO/IEC Recommendation, 1996.

[65] International Standard, *ISO/IEC 13818-2, Information Technology - General Coding of Moving Picture and Associated Audio Information: Video*, ISO/IEC Recommendation, 1996.

[66] International Standard, *ISO/IEC 13818-3, Information Technology - General Coding of Moving Picture and Associated Audio Information: Audio*, ISO/IEC Recommendation, 1996.

[67] A. K. Jain, *Fundamentals of Digital Image Processing*, Prentice Hall, New Jersey, 1989.

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

[69] M. Kutter, "Digital Signature of Color Images using Amplitude Modulation," *Proceedings of the SPIE, Storage and Retrieval for Image and Video Databases V,* Vol. 3022, pp. 518-525, San Jose, Feb. 1997.

[70] M. Lesk, "Digital Libraries meet Electronic Commerce: On-Screen Intellectual Prop-

erty," *Proceedings of the Third Forum on Research and Technology Advances in Digital Libraries*, pp. 58-64, Washington, DC, May, 1996.

[71] J. Lacy, J. H. Snyder and D. P. Maher, "Music on the Internet and the Intellectual Property Protection Problem," *Proceedings of the IEEE International Symposium on Industrial Electronics,* Vol. 1635, pp. 77-83, Guimaraes, Portugal, July, 1997.

[72] J. P. M. G. Linnartz and J. C. Talstra, "MPEG PTY-Marks: Cheap Detection of Embedded Copyright Data in DVD-Video," *5th European Symposium on Research in Computer Security. Proceedings,* pp. 221-40, Louvain-la-Neuve, Belgium, September, 1998.

[73] G. Legge and J. Foley, "Contrast Masking in Human Vision," *Journal of Optical Science, America,* Vol. 70, no. 12, pp.1458-1471, December 1990.

[74] C. Y. Lin and S. F. Chang, "A robust Image Authentication Method Surviving JPEG Lossy Compression," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database VI,* Vol.3312, pp. 296-307, San Jose, February, 1998.

[75] G. C. Langelaar, J. van der Lubbe and R. L. Lagendijk, "Robust Labeling Methods for Copy Protection of Images," *Proceeding of the SPIE, Storage and Retrieval for Image and Video Database V,* Vol.3022, pp. 298-309, San Jose, February, 1997.

[76] S. G. Mallat, "A theory for Multiresolution Signal Decomposition: The Wavelet Representation," *IEEE Tr. Pattern Analysis and Machine Intelligence,* Vol. 11, no. 7, pp. 674-693, July 1989.

[77] S. G. Mallat, Multifrequency Channel Decompositions of Images and Wavelet Models," *IEEE Tr. Acoustic Speech and Signal Processing,* Vol. 37, pp. 2091-2110, December 1990.

[78] K. Matsui and K. Oka, "Embedding a Watermark to Binary Pictures in Hardcopy System," Memoirs of the National Defence Academic, Japan, Vol. 36, no. 2, pp. 13-20, 1997.

[79] C. Maroney, Hide and Seek (program source), ftp://ftp.csua.berkerly.edu/pub/cypherpunks/steganography/hdsk41b.zip

[80] B. G. Mobasseri, "Direct Sequence Watermarking of Digital Video using m-frames," *Proceedings of IEEE International Conference on Image Processing.* Vol. 2, pp. 399-403, Chicago, Illinois, 1998.

[81] N. F. Maxemchuk, "Electronic Document Distribution," *AT&T Technical Journal*, vol. 73, no. 5, pp. 73-80, 1994.

[82] L. M. Marvel, C. T. Retter and C. G. Boncelet Jr., "Hiding information in images," Proceedings IEEE International Conference on Image Processing, Vol.2, pp.396-398, Chicago, Illinois, October 1998.

[83] D. Mukherjee, J. J. Chae and S. K. Mitra, "A Source and Channel Coding Approach to Data Hiding with Application to Hiding Speech in Video," *Proceeding of IEEE International Conference of the Image Processing*, Vol. 1, pp. 348-352, Chicago, October, 1998.

[84] J. L. Mitchell, *MPEG Video: Compression Standard*, Chapman & Hill, New York, 1996.

[85] J. Meng and S. -F. Chang, "Embedding Visible Video Watermarkings in the Compressed Domain," *Proceedings of IEEE International Conference on Image Processing*. Vol. 1, pp. 474-477, Chicago, Illinois, October, 1998.

[86] NIST, *Data Encryption Standard*, FIPS Publication 46-2, 1993

[87] J. Ohnishi and K. Matsui, "Embedding a Seal into a Picture under Orthogonal Wavelet Transform," *International conference on Multimedia and Computing and System*, pp. 514-512, 1996

[88] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking Three-Dimensional Polygonal Models Through Geometric and Topological Modifications," *IEEE Journal on Selected Areas in Communications*, Vol. 16, no. 4, pp. 551-560, May. 1998

[89] R. Ohbuchi, H. Masuda and M. Aono, "Watermarking Three-Dimensional Polygonal Models," *IEEE Journal on Selected Areas in Communications*, vol. 16, no. 4, pp. 551-560, May, 1998.

[90] A. Piva, M. Barni, F. Barolini and V. Cappellini, "DCT-based Watermark Recovering without Resorting to the Uncorrupted Original Image," *Proceeding of IEEE International Conference of Image Processing*, Vol. 1, pp. 520-523, Santa Barbara, California, October 1997.

[91] C. Podilchuk and W. Zeng, "Perceptual Watermarking of Still Images," *IEEE Workshop on Multimedia Signal Processing*, pp. 363-368, New York, 1997.

[92] C. Podilchuk and W. Zeng, "Image-Adaptive Watermarking using Visual Models," *IEEE Journal on Selected Areas in Communications*, Vol. 16, no. 4, pp. 525-539, May 1998.

[93] J. G Proakis, *Digital Communications*, McGraw Hill, 3rd ed., New York, 1995

[94] E. Peli, *Vision Models for Target Detection and Recognition*, Chapter: A Visual Discrimination Model for Image System Design and Evalutation, Singapore, World Scientific Publishing, 1995.

[95] L. Qiao and K. Nahrstedt, "Watermarking Methods for MPEG Encoded Video: Towards Resolving Rightful Ownership," *Proceedings of IEEE International Conference of Multimedia Computing and Systems*, pp. 276-285, Austin, Texas, June 1998.

[96] L. Qiao and K. Nahrstedt, "Watermarking Schemes and Protocols for Protocols for Protecting Rightful Ownership and Customer's Right," *Journal of Visual Communication and Image Representation*, Vol. 9, no. 3, pp. 194-210, September, 1998.

[97] L. Qiao and K. Nahrstedt, "Is MPEG Encryption by using Random List instead of ZIg-zag Order Secure?" *IEEE international Symposium on Consumer Electronics*, pp. 226-229, Singapore, December, 1977.

[98] A. Ravishankar Rao, G. W. Braudaway and F. C. Mintzer, "Automatic Visible Watermarking of Images," *Proceeding of SPIE, Optical Security and Counterfeit Deterrence Techniques II*, vol.3314, pp.110-121. San Jose, California, January, 1998.

[99] M. Ramkumar and A. N. Akansu, "Information theoretic Bounds for Data Hiding in Compressed Images," IEEE Signal Processing Society 1998 Workshop on Multimedia Signal Processing, pp. 267-272, Los Angeles, California, December, 1998.

[100] K. Ramachandran, M. Vetterli and C. Herley, "Wavelets, Subbands Coding and Best Bases," *Proceeding of IEEE*, vol. 84, no. 4, pp. 541-560, April 1996.

[101] L. A. Rowe, J. S. Boreczky and D. A. Berger, "A Distributed Hierarchical Video-on-Demand System," *Proceedings of IEEE International Conference on Image Processing*, Vol. 1, pp. 334-337, Washington, DC, October, 1995.

[102] J. J. O Ruanaidh, W. J. Dowling, and F. M. Boland, "Phase Watermarking of Digital Images" *Proceeding of IEEE International Conference of the Image Processing*, Vol. 3, pp.239-242, Switzerland, September, 1996.

[103] J. J. O Ruanaidh, W. J. Dowling, and F. M. Boland, "Watermarking Digital Images for Copyright Protection," *IEE Proceeding of Vision, Image, Signal Processing,* Vol. 143, no. 4, pp. 250-256, August, 1996.

[104] A. Said and W. A. Perlman, "An Image Multiresolution Representation for Lossless and Lossy Compression," *IEEE Tr. on Image Processing,* Vol. 5, no. 9, September 1996.

[105] J. M. Shapiro, "Embedded Image Coding using Zerotrees of Wavelet Coefficients," *IEEE Tr. on Signal Processing,* Vol. 41, no. 12, December 1993.

[106] E. A. B. da Silva, D. G. Sampson and M. Ghanbari, "Image Coding using Successive Approximation Wavelet Vector Quantization," *1995 International Conference on Acoustics, Speech, and Signal Processing.* Vol. 4, pp. 2201-2204, Detroit, Michigan, May 1995.

[107] G. Strang and T. Nguyen, *Wavelets and Filer Banks,* Wellesley-Cambridge Press, 1996.

[108] G. C. M. Silvestre and W. J. Dowling, "Image Watermarking using Digital Communication Techniques," *Proceedings of 6th International Conference on Image Processing and its Applications,* Vol. 1, pp 443-437, Dublin, Ireland, July, 1997.

[109] M. D. Swanson, B. Zhu and A. H. Tewfik, "Data Hiding for Video-in-Video," *IEEE International Conference of Image Conference,* Vol. II, pp. 676-679, Santa Barbara, October, 1997.

[110] M. D. Swanson, B. Zhu and A. H. Tewfik, "Multiresolution Video Watermarking using Perceptual Models and Scene Segmentation," *IEEE International Conference of Image Conference,* Vol. II, pp. 558-561, Santa Barbara, October, 1997.

[111] M. D. Swanson, B. Zhu and A. H. Tewfik, "Transparent Robust Image Watermarking," *International Conference of the Image Processing '96,* Vol. 3, pp. 211-214, Switzerland, September, 1996.

[112] M. D. Swanson, B. Zhu and A. H. Tewfik, "Robust Data Hiding for Images," *IEEE Digital Signal Processing Workshop,* pp. 37-40, Norway, September, 1996.

[113] M. D. Swanson, M. Kobayashi and A. H. Tewfik, "Multimedia Data-Embedding and Watermarking Technologies," *Proceedings of the IEEE,* Vol. 86, no. 6, pp. 1064-1087, June, 1998.

[114] M. D. Swanson, B. Zhu and A. H. Tewfik, "Multiresoultion Scene-based Video Watermarking using Perceptual Models," *IEEE Journal on Selected Areas in Communications*, Vol. 16, no. 4, pp. 540-550, May, 1998.

[115] R. G. van Schyndel, A. Z. Tirkel and C. F. Osborne, "A Digital Watermark," *Proceeding of IEEE International Conference of Image Processing,* Vol. 2, pp. 86-90, Austin, Texas, November 1994

[116] S. D. Servetto, C. I. Podilchuk and Kannan Ramchandran, "Capacity Issues in Digital Image Watermarking," Proceedings of IEEE International Conference on Image Processing, Vol. 1, pp. 445-449, Chicago, Illinois, October, 1998.

[117] R. Todd Ogden, *Essential Wavelets for Statistical Applications and Data Analysis,* Birkhauser, Boston, 1997.

[118] B. Tao and B. Dickison, "Adaptive Watermarking in the DCT Domain," *1997 IEEE International Conference Acoustics, Speech, and Signal Processing,* Vol. 4, pp. 2985-2988, Munich, Germany, April, 1997

[119] UCSB ECE Image Processing Laboratory, "Literature Survey: Image Quality Project," *Technical Report*, 1998.

[120] S. E. Umbaugh, *Computer Vision and Image Processing: A Practice Approach using CVIPtools*, Prentices Hall, New Jersey, 1998.

[121] G. Voyatzis and I. Pitas, "Embedding Robust Watermarks by the Chaotic Mixing," *Proceeding of IEEE Digital Signal Processing Workshop,* Vol. 1, pp. 213-216, Greece, July 1997.

[122] M. Verterli and J. Kovacevic, *Wavelets and Subband Coding*, Prentice Hall, New Jersey, 1995.

[123] A. B. Watson, "DCT Quantization Matrics Visually Optimized for individual Images," *SPIE Human Vision, Visual Processing, and Digital Display IV,* Vol. 1913, p.202-216., San Jose, California, February. 1993.

[124] G. K. Wallage, "The JPEG Still Picture Compression Standard," *Communication of the ACM,* Vol. 34, no. 4, pp. 31-44, April, 1991.

[125] J. Z. Wang and G. Wiederhold, "WaveMark: Digital Image Watermarking Using Daubechies' Wavelets and Error Correcting Coding," *Proc. SPIE Symposium on Voice,*

Video and Data Comm., Vol. 3528, pp. 432-439, Boston, MA, November 1998.

[126] X. Xia, C. G Boncelet and G R. Arce, "A Multiresolution Wavelet for Digital Images," *IEEE International Conference of Image Processing,* Vol. 1, pp. 548-551, Santa Barbara, California, 1997.

[127] L. Xia and G R. Arce, "Joint Wavelet Compression and Authentication Watermarking," *Proceedings of IEEE International Conference of Image Processing,* Vol. 2, pp. 427-431, Chicago, Illinois, October 1998.

[128] M. M. Yeung, and F. C. Mintzer, "Digital Watermarking for High-quality Imaging," *IEEE first workshop on the Multimedia Signal Processing,* pp. 357-362, 1997

[129] J. Zhao and E. Koch, "A digital watermarking system for multimedia copyright protection," *Proceedings ACM Multimedia 96,* pp. 443-457, Boston, MA, November 1996.

[130] W. Zeng, *Resilient video transmission and multimedia database applications,* Ph.D Dissertation, princeton Univ., June, 1997.

[131] W. Zeng, B. Liu and S. Lei, "Extraction of Multiresolution Watermark Images for Resolving Rightful Ownership," Proceedings of SPIE EI '99, Security and Watermarking of Multimedia Contents, Vol. 3657, San Jose, California, January 1999.

[132] B. Zhu, A. H. Tewfik and O. Gerek, "Low Bit Rate Near-Transparent Image Coding," *Processing of the SPIE, Wavelet Applications for Dual Use,* Vol. 2491, pp. 173-184, San Jose, California, February, 1995