ABSTRACT

DATA HIDING IN MULTIMEDIA - THEORY AND APPLICATIONS

by
Mahalingam Ramkumar

Multimedia data hiding or steganography is a means communication using *subliminal* channels. The *resource* for the subliminal communication scheme is the *distortion of the original content* that can be tolerated. This thesis addresses two main issues of steganographic communication schemes:

1. How does one maximize the distortion introduced without affecting fidelity of the content?

2. How does one *efficiently utilize* the resource (the distortion introduced) for communicating as many bits of information as possible? In other words, what is a good *signaling* strategy for the subliminal communication scheme?

Close to optimal solutions for both issues are analyzed. Many techniques for the issue for maximizing the resource, viz. the distortion introduced imperceptibly in images and video frames, are proposed. Different signaling strategies for steganographic communication are explored, and a novel signaling technique employing a floating signal constellation is proposed. Algorithms for optimal choices of the parameters of the signaling technique are presented.

Other application specific issues like the type of robustness needed are taken into consideration along with the established theoretical background to design optimal data hiding schemes. In particular, two very important applications of data hiding are addressed - data hiding for multimedia content delivery, and data hiding for watermarking (for proving ownership). A robust watermarking protocol for unambiguous resolution of ownership is proposed.

DATA HIDING IN MULTIMEDIA - THEORY AND APPLICATIONS

by
Mahalingam Ramkumar

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

Department of Electrical and Computer Engineering

January 2000

APPROVAL PAGE

DATA HIDING IN MULTIMEDIA - THEORY AND APPLICATIONS

Mahalingam Ramkumar

Ali N. Akansu, Dissertation Advisor — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Richard Haddad, Committee Member — Date
Professor of Electrical and Computer Engineering, NJIT

Dr. Necdet Uzun, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Hongya Ge, Committee Member — Date
Assistant Professor of Electrical and Computer Engineering, NJIT

Dr. Naser Memon, Committee Member — Date
Associate Professor of Computer Science, Brooklyn Polytechnic, Brooklyn, NY

BIOGRAPHICAL SKETCH

Author : Mahalingam Ramkumar

Degree : Doctor of Philosophy

Education : Doctor of Philosophy, October 1999
Department of Electrical and Computer Engineering
New Jersey Institute of Technology
Newark, NJ

Master of Science (Engineering), Jan 1997
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore, India.

Bachelor of Engineering, Jun 1987
Department of Electrical and Electronics Engineering
Government College of Engineering, University of Madras
Salem, India

Publications

1. M.Ramkumar, G.V. Anand, "An FFT-Based Technique for Fast Fractal Image Compression", *Signal Processing*, **63**[2], 1997, pp 263-268.

2. M.Ramkumar, A.N Akansu, "Capacity Estimates for Data Hiding in Compressed Images", Submitted to the *IEEE Trans. on Image Processing.*

3. M.Ramkumar, A.N Akansu, "Signaling for Multimedia Steganography", submitted to the *IEEE Trans. on Signal Processing.*

4. M. Ramkumar, A.N. Akansu, "Robust Protocols for Proving Ownership of Still Images", submitted to the *IEEE Trans. on Multimedia.*

5. M. Ramkumar, A.N. Akansu, "Optimal Design of Data Hiding Methods Robust to Lossy Compression", submitted to the *IEEE Trans. on Multimedia.*

6. M. Ramkumar, A.N. Akansu, "A Performance Study of DCT and Subband Image Codecs with Zero-Zone Quantizers", *IEEE ISCAS*, vol **2**, pp 421-424, June 1998.

7. M.Ramkumar, A.N. Akansu, "A Robust Scheme for Oblivious Detection of Watermarks/ Data Hiding in Still Images", *SPIE Multimedia Systems and Applications, Boston*, MA, **3528**, pp 474 - 481, November 1998.

8. M.Ramkumar, A.N. Akansu, " Theoretical Capacity Measures for Data Hiding in Compressed Images", *SPIE Multimedia Systems and Applications*, Boston, MA, **3528**, pp 482 - 492, November 1998.

9. M. Ramkumar, A.N. Akansu, "Information Theoretic Bounds for Data Hiding in Compressed Images", *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, California, USA, pp 267-272, December 1998.

10. M.Ramkumar, A.N. Akansu, A. Alatan, "On the Choice of Transforms for Data Hiding in Compressed Video", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, vol **VI**, pp 3049 - 3052, March 1999.

11. M.Ramkumar, G.V. Anand and A. N. Akansu, "On the Implementation of 2-Band Cyclic Filterbanks", *IEEE International Symposium on Circuits and Systems*, Orlando, Florida, vol **III**, pp 520 - 523, May 1999.

12. M.Ramkumar, A.N Akansu, "Image Watermarks and Counterfeit Attacks : Some Problems and Solutions", *Content Security and Data Hiding in Digital Media*, Newark, NJ, pp 102-112, May 1999.

13. M.Ramkumar, A.N Akansu, "Data Hiding for Internet Multimedia", *Content Security and Data Hiding in Digital Media*, Newark, NJ, pp 12-23, May 1999.

14. M.Ramkumar, A.N. Akansu, "Self-Noise Suppression Schemes for Blind Image Steganography", *SPIE Multimedia Systems and Applications (Image Security)*, vol **3845**, Boston, MA, September. 1999.

15. M.Ramkumar, A.N Akansu, "On the Design of Robust Data Hiding Systems", to be presented in the 33rd *ASILOMAR Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 1999.

16. M.Ramkumar, A.N. Akansu, "Floating Signal Constellations for Multimedia Steganography", submitted to *IEEE International Conference on Communications*, 2000.

17. M.Ramkumar, A.N. Akansu, "FFT-Based Signaling for Multimedia Steganography", submitted to *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000.

18. M.Ramkumar, A.N. Akansu, "A Robust Protocol for Proving Ownership of Still Images", Submitted to the *International Conference on Information Technology: Coding and Computing*, 2000.

TABLE OF CONTENTS

Chapter **Page**

TABLE OF CONTENTS
(Continued)

Chapter **Page**

TABLE OF CONTENTS

(Continued)

Chapter **Page**

LIST OF TABLES

Table **Page**

LIST OF FIGURES

Figure **Page**

LIST OF FIGURES

(Continued)

CHAPTER 1

INTRODUCTION

Data Hiding or *Steganography* is the art of hiding a *message signal* in a *host signal*, without any perceptual distortion of the host signal [1]. Though steganography is often confused with the relatively well-known *cryptography*, the two are but loosely related. Cryptography is about hiding the *contents* of a message [2]. Steganography, on the other hand, is about *concealing the very fact that a message is hidden.* Steganography may be considered as communication through *subliminal* channels, or *secret* communication [3, 4]. This thesis explores the theory and applications of multimedia steganography.

The proliferation of digital multimedia as opposed to conventional analog forms, is primarily a result of

- the ease with which digital data can be exchanged over the Internet, and
- the emergence of efficient multimedia data compression techniques.

The first reason is also a major cause for concern. Unlimited *perfect copies* of the original content can be made, and distributed easily. It was this concern of protecting intellectual property rights of multimedia data in digital form, that primarily triggered researchers to find ways to *watermark* multimedia data. Watermarking the content is done by embedding some data in the host signal (original content). The embedded data may be an imperceptible *signature*, which, the owner of the multimedia content should be able to extract when a dispute regarding ownership occurs.

The pioneers of digital steganography [5], perhaps had no idea of the extent of potential applications for multimedia data hiding. Data hiding can help authenticate

electronic correspondence. It can facilitate adding a 'signature' to electronic mails which could make an e-mail as valid a document as an agreement signed on stamp paper! This could facilitate carrying out accountable business deals could over the Internet.

Data hiding in multimedia [5, 6, 7, 8] could help in providing proof of origin and distribution of content. Multimedia content providers would be able to communicate with the *compliant multimedia players* through the *subliminal*, steganographic channel. This communication may control or restrict access of multimedia content, and carry out e-commerce for pay-per-use implementations. The concept of compliant multimedia players may extend to operating systems which would recognize protected multimedia files. So one may not be able to print a document or make additional copies unless authorized by the hidden data in the document. All material available on paper, may eventually be in electronic form. Downloading or distributing the documents could be controlled by the hidden data.

A typical application of data hiding for multimedia content delivery may take the form depicted in Figure 1.1. The *content providers* supply the raw multimedia data (say a full length movie) along with some hidden agents or *control data*. The job of the *distributors* would be to package the content in some suitable format (like MPEG) understandable by the players, and distribution of the multimedia either through DVDs/CDs, live digital broadcasts or by even hosting web sites for downloads. The compliant multimedia *players*, will typically be connected to the Internet.

In conventional multimedia distribution, the content provider looses all control over how the multimedia, is used / abused the moment it leaves his/her hands. The key idea behind data hiding is to re-establish control whenever the content is used. The content provider, by hiding some agents in his raw data, hopes to control access to his/her multimedia content. This can be done with the *co-operation of the players*,

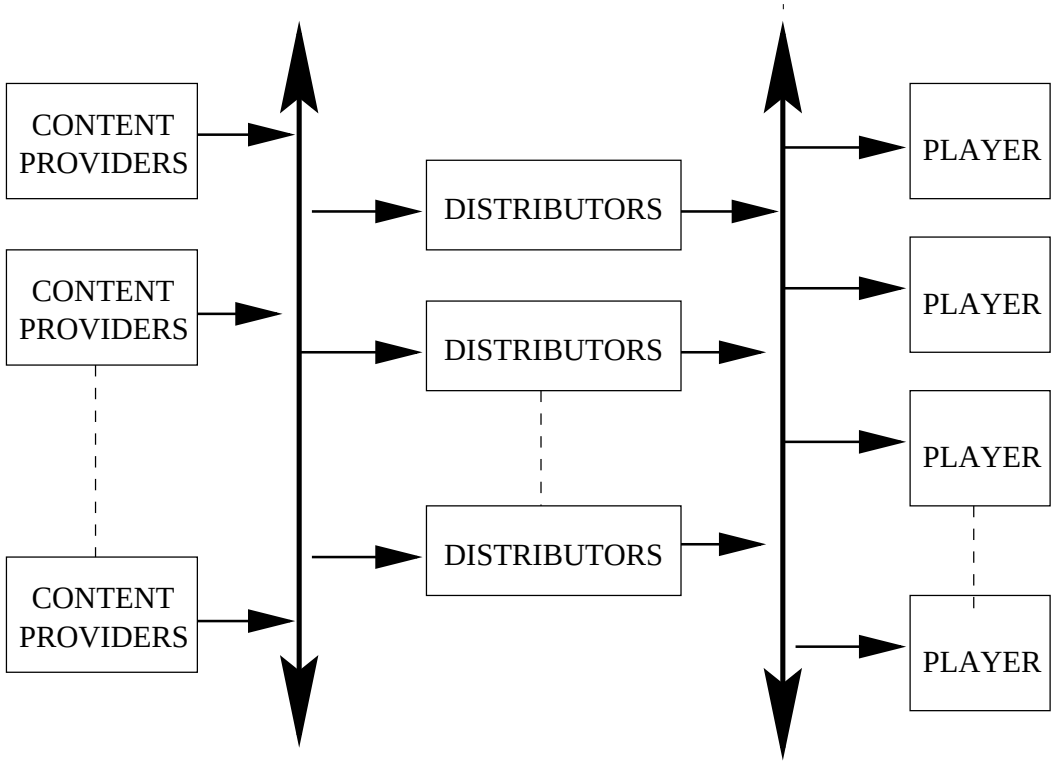

Figure 1.1 A multimedia content delivery scheme

and an *established protocol for communication* between the *content providers* and the *compliant multimedia players*.

For most data hiding applications to become a reality, some important steps have to be taken:

- Establishing protocols for *authentication of content* that is acceptable in a court of law. The protocols may be different for different types of multimedia data.
- Establishing protocols for communication over the subliminal channel (between the content providers and the compliant multimedia players).

1.1 The Steganographic Channel

Figure (1.2) depicts a block diagram of a general data hiding channel. **A** is the original multimedia data which is also referred to as the *cover* or *stego* object. The

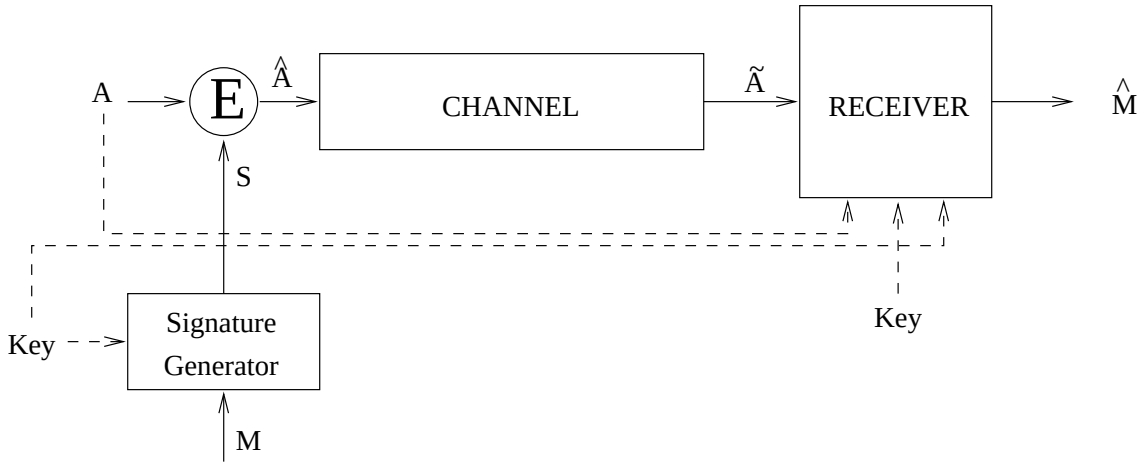

Figure 1.2 Block diagram of multimedia steganography

stego object serves as the carrier for the hidden message $\mathbf{M}$. The message $\mathbf{M}$ is converted to a signature $\mathbf{S}$ which is in a form suitable for being *embedded* in $\mathbf{A}$:

$$\mathbf{S} = \mathcal{S}(\mathbf{M}, \mathbf{K}) \tag{1.1}$$

In Eq. (1.1), $\mathcal{S}$ is the signature generator block in Figure (1.2), and the key $\mathbf{K}$ may be *private* or *public* or a combination of both. Most often the embedding operation E, takes the form of super positioning of $\mathbf{S}$ with $\mathbf{A}$ to obtain $\hat{\mathbf{A}}$. However, other forms of embedding is also possible.

$$\hat{\mathbf{A}} = \mathbf{A} + \mathbf{S} \text{ or } \hat{\mathbf{A}} = E(\mathbf{A}, \mathbf{S}). \tag{1.2}$$

The *imperceptibly modified* multimedia data $\hat{\mathbf{A}}$, is transmitted through a channel $\mathcal{C}$, and emerges at the other end as $\tilde{\mathbf{A}} = \mathcal{C}(\hat{\mathbf{A}})$. Now, the buried message signal $\mathbf{M}$ is retrieved in the receiver by a detector $\mathcal{D}$ as

$$\tilde{\mathbf{M}} = \mathcal{D}(\tilde{\mathbf{A}}, \mathbf{K}) \tag{1.3}$$

In some cases, (for *e.g*, watermarking applications), the detector $\mathcal{D}$ may require the original $\mathbf{A}$ for extracting the hidden message or signature;

$$\tilde{\mathbf{M}} = \mathcal{D}(\tilde{\mathbf{A}}, \mathbf{A}, \mathbf{K}) \tag{1.4}$$

The channel apart from other things, may include a lossy compressor at one end and decompresser at the other end. While this is the main cause of concern for most data hiding applications, the same is not true for watermarking applications. For watermarking applications, the channel may include agents with *deliberate intentions* of removing the watermark.

This thesis is a study of theory and applications of data hiding in still images and video. However, most of the suggested techniques are also applicable for data hiding in digital audio.

1.2 Organization of the Thesis

This thesis is organized as follows. Chapter 2 is a brief review of existing literature devoted to watermarking and data hiding.

In Chapter 3, the image / video steganographic channel is modeled as a communication channel [9, 10, 11, 12]. However, the embedding operation E in Figure 1.2 is assumed to be *linear addition* (in recent data hiding literature, linear data hiding methods are referred to as Type I methods). The data hiding channel is characterized as a channel with 2 sources of noise - noise due to the stego image/ frame and noise due the data compressor in the channel (the former is also referred to as *image noise, self-noise* or *host-signal noise*, and the latter as *processing noise* or *channel noise*. It is seen that the performance of the data hiding channel can be improved significantly by decomposing the data hiding channel into multiple parallel channels. The decomposition is usually performed by some unitary transform. The purpose of the decomposition, is primarily to efficiently redistribute the two sources of noise amongst different channels. Estimates of the noise sources in each subchannel from several test images and video sequences, for widely used compressors like JPEG, SPIHT [13] and MPEG [14] are obtained. This is in turn used obtain information theoretic estimates of the capacity of the linear data hiding channel for

different decompositions. It is argued why the choice of the decomposition should be motivated by the required *robustness* of the data hiding application.

Chapter 4 investigates other options for the embedding operation E in Figure 1.2. It is shown how non-linear embedding techniques can suppress the self-noise to a large extent [15, 16, 17, 18], even though the original content is not available at the receiver. The problem of data hiding is viewed as a sophisticated signaling method employing a *floating* signal constellation. Therefore, the *origin of the signal constellation* has to be *estimated* by the receiver. The signaling method is split into two steps. The first step estimates the origin of the floating constellation. The second step is the definition of the constellation itself. The step that estimates the origin is termed as the self-noise suppression (SNS) technique. Optimal methods to achieve SNS are proposed and their performance evaluated under an additive noise scenario, both by means of simulations and rigorous analysis. Some of the widely reported non-linear data hiding methods, based on quantization (referred to as Type II methods in recent literature), are shown to be special cases of the proposed floating signal constellation. Investigation of optimal choice of the parameters of the floating signal constellation, show that an extension of Type II methods is needed. The extension (Type III), takes the form of thresholding the distortion introduced by Type II methods. Type III is then shown to be a generalization of both Type I and Type II methods. Type I methods are a special case of Type III methods which are optimal only when the SNR tends to zero. Similarly, Type II methods (another special case of Type III) are optimal when the SNR approaches infinity.

Chapter 5 investigates options for the choice of the signal constellation for data hiding. An FFT based signaling method [19] with properties that make it especially useful for multimedia steganography is proposed. In the proposed technique, the signal constellation is defined by cyclic all-pass filters generated from random keys.

The intricate relationship between data hiding and data compression is explored in Chapter 6. It is seen that efficient data hiding techniques should utilize "holes" in compression techniques [20, 21]. Further, it is necessary for data hiding techniques to be robust to all known compression methods, and perhaps compression techniques which may evolve in the future. To achieve this it may be necessary to utilize "holes" common to *all* compression schemes. Such a "hole" is identified methods to exploit that "hole" are proposed. The chapter concludes with an optimal data hiding method based on the principles outlined in Chapters 4, 5 and 6. In addition other concerns like security, and computational complexity are taken into account for making appropriate trade-offs.

Chapter 7 addresses the problem of unambiguous resolution of ownership with digital watermarks. A protocol for watermarking which virtually guarantees immunity to counterfeit claims [22, 23, 24], is proposed. The techniques proposed in Chapter 6 for utilizing "holes" in compression methods addressed how data hiding can effectively survive compression. However, data hiding methods for watermarking have to be robust to intentional attacks too. Chapter 7 addresses this issue, and proposes a technique for robust watermarking.

Conclusions, and suggestions for future research are offered in Chapter 8.

CHAPTER 2

A BRIEF REVIEW OF DATA HIDING

Applications of the field of steganography date back to earlier than 1000 BC [1]. However the revival of its applications started with increasing concerns of protecting intellectual property rights of digital multimedia. Steganographic *applications* can be broadly classified into two categories [25] -

- steganography with *active wardens*, and
- steganography with *passive wardens*.

The data hiding parallels to the two categories are respectively data hiding schemes in which *intentional tampering* is not an issue (for example, captioning) and schemes which need resistance to intentional tampering (for example, watermarking). Depending on the desired properties of the data hiding scheme, we classify data hiding applications into the following three categories:

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

2.1 Watermarking

Watermarking schemes can be broadly classified into two categories. Methods that need the original (unwatermarked) image for extraction of the watermark (or *cover image escrow* methods), and methods for which the original is not necessary (*oblivious or blind detection*). Apparently, the former methods are bound to be more efficient as they have to resist only the noise due to processing (intentional and unintentional). The latter however should also overcome the host signal noise.

A watermark, added to an image or video frame should in general satisfy the following properties:

- **Robustness**. The watermark should resist both intentional and non-intentional tampering. Examples of non-intentional tampering are some common signal processing operations like lossy compression, histogram equalization, edge enhancement, low-pass filtering, gamma correction, scaling, rotation, D/A and A/D conversions, color adjustment etc.. Intentional tampering is done with the sole purpose of removing the watermark while simultaneously trying to protect the quality of the image. Many schemes / software packages for intentional tampering have been proposed, like StirMark, UnZign, and Richard Barnett's attack software.

- **Invisibility**. The watermark should be perceptually transparent. This implies that the watermark energy should be very small (there exists a possibility of having visible watermarks, but we shall not discuss them in this thesis due to their limited application).

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

- **Unambiguous**. Most importantly, the watermark should be able to resolve rightful ownership unambiguously. This may place some restrictions on the methods that can be used for watermarking.

The properties a watermark should satisfy for being acceptable in a court of law (to be able to establish the identity of the creator unambiguously) has itself been an active area of research [22, 23, 24, 26, 27, 28, 29]. Watermarking may also be used for uniquely identifying each copy distributed by the owner. For example, in the above case the creator **A** may sell many copies of his image I. While all the copies will have the same watermark to establish ownership, they might have additional buried information pertaining to the buyer of the particular copy (it may just be a serial number).This would help in incriminating the particular buyer responsible for creating unauthorized copies. If a particular buyer makes illegal copies of the image for distribution, then the copy can be traced to the buyer responsible for its circulation. But the accused buyer can still claim that the copies were circulated by the owner of the original image to frame the buyer. To avoid this situation, a cooperative buyer-seller protocol may be needed [29].

2.1.1 Watermarking for Tamper Detection

Multimedia stored in digital format can be easily modified, or forged with a wide variety of available software. Data hiding for tamper-proofing can go a long way in verifying the authenticity of the data. Tampering may be intentional or unintentional. Applications for tamper - proofing may prove important for courtroom evidence and journalistic photography.

In [30, 31], spatial domain watermarks were used. In Ref. [32] the watermark is added in the wavelet domain. This method, in addition to identifying the spatial location of the change, also indicates the type of tampering undergone.

In Ref. [33], a method suited for hardware implementation for watermarking the images by cameras is proposed. In this method, the watermark is placed takes the form of a spread spectrum sequence, added to 32×32 or 64×64 blocks. The watermark is capable of identifying the particular blocks that have been tampered with.

2.1.2 Attacks Against Watermarks

Attacks against watermarks can broadly be classified into two categories viz., counterfeit attacks and signal processing attacks. The former schemes, exploit inadequacies of the watermarking protocols to unambiguously resolve rightful ownership. They are described in greater detail in Chapter 8.

The latter are aimed at removing the signature carefully designed strategies. Some examples are StirMark, UnZign, Richard Barnett's attack software etc. The StirMark attack, for instance, simulates image distortions that commonly occur when a picture is printed, photocopied and rescanned. This also introduces imperceptible geometrical changes which results in a loss of synchronization between the watermark detector and the image. In Ref. [34, 35], Cox *et. al.* discuss the effectiveness of different attacks like affine transformations, noise reduction, compression, exploiting the watermark detector / inserter device to obtain better estimates of the watermark.

In Ref. [36] the authors compare the effectiveness of different attacks like the jitter attack, StirMark, and mosaic attacks. The jitter attack is meant for watermarking schemes that modify the least significant bits of audio / image data. In the mosaic attack, a watermarked image is chopped into a large number of small sub-images which are embedded in a suitable sequence in a web browser such that the final presentation image is very similar to the original watermarked image.

Of all the watermark attack software, StirMark is probably the most effective, and proven to effectively attack most know watermarking schemes. In Ref. [37]

Petitcolas *et. al.* suggest that StirMark should be used as a benchmark for evaluating the effectiveness of watermarking schemes.

2.2 Data hiding for Multimedia Delivery

There are numerous emerging applications in this category. What is common to all of them is that, unlike watermarking applications, 'unambiguous resolution of owner's identity' is not an issue. In addition, all these applications may depend on the existence of a common protocol for communication between the content provider and the player (or application software for viewing the image or playing the video / audio clip).

The required robustness, secrecy, and number of bits to be encoded also varies from application to application. For example, captioning applications may not need very good robustness. Commercial applications may need robustness only to the standard compression scenarios that the data is most likely to undergo. In most cases intentional tampering may not be an issue. Intentional tampering can cause more loss than gain to the end user. Captioning and hiding executables may need a large number of bits, which might, however, not be problem in video applications.

2.3 Data Hiding Techniques

2.3.1 Spatial Domain Methods

Early work in data hiding consisted mainly of modifying the least significant bits (LSB) of images to hide data. In Ref. [38] the hidden signal was restricted to modifying the two least significant bits. In Ref. [39] the author suggests adding small geometric patterns - tags - to digitized images at brightness levels that are imperceptible. Bender *et. al.* [40] proposed the "Patchwork" algorithm. The algorithm selects random pairs of pixels. It enhances the value of high valued pixel and reduces the value of low valued pixel. The contrast change in the pair is used to encode one

bit. This method however is not effective for images which do not have uniformly distributed pixel values.

Pitas *et. al.* [41, 42] introduce a method which in principle is very similar to Patchwork. But they extend the pairs of points to blocks, which results in better resistance to JPEG compression. Delp *et. al.* [43, 44], use a two dimensional watermark (which is actually obtained by reshaping one dimensional M-Sequences as a matrix.)

In Ref. [45] a more robust watermarking scheme is proposed. The robustness is achieved by forcing the signature to be low pass, so that the signature is relatively tolerant to compression. The extracted bits are mapped to a *visualizer* to display a meaningful watermark.

2.3.2 DCT and Wavelets based Data Hiding

Perhaps [46] is the first work utilizing DCT decomposition for data embedding. In this method the watermark does not tile the image completely - only some randomly selected regions are altered to embed the watermark. In this scheme, a "relationship" is encoded in blocks by swapping selected coefficients.

Cox *et. al.* [47] were the first to introduce the idea of embedding the watermark in the *perceptually significant* coefficients of an image. In their scheme, the watermark altered 1000 low frequency DCT coefficients (2-D DCT of the entire image). A Gaussian sequence is used as the signature. Detection of the signature is accomplished by correlating the Gaussian sequence with the 1000 (modified) DCT coefficients after subtraction of the corresponding DCT coefficients of the cover image.

In Ref. [48], a block based DCT is used instead of taking the DCT of the whole image. In Ref. [49, 50], 8×8 block DCT is used. However not all blocks are altered. Only blocks with high activity are altered. The watermark modifies the mid-frequency DCT coefficients.

Swanson *et. al.* [51, 52] propose an efficient watermarking scheme based on spatial masking [53] of the watermarking sequence to ensure invisibility of the watermark. The spatial mask is used to calculate the maximum allowable change for each DCT coefficient in each block.

In Ref. [54], Zeng *et. al.* raise the issue of the inability of cover image escrow watermarking schemes to resolve rightful ownership. They therefore introduce a oblivious detection scheme, in which the watermark signal is added to the 8×8 block DCT coefficients. The watermark is detected by correlating the signature with the DCT coefficients.

Fridrich proposes a hybrid watermarking scheme [55]. This hybrid scheme uses a full size 2-D DCT decomposition, and modifies the low-frequency coefficients to introduce the low-frequency watermark. In addition, a spread spectrum signal is added to the mid-frequency DCT coefficients.

Wavelet based data embedding schemes have been as widely reported as DCT based schemes. In Ref. [56] the wavelet decomposition of a signature matrix is added to the wavelet decomposition of the cover image. However, the signature coefficients are scaled by a factor depending on the contrast sensitivity of spatial frequencies [57].

In Ref. [58], the cover image is decomposed in a pyramidal fashion. The watermark is added such that it can be detected hierarchically. If the image distortion is not serious, only a few bands of the decomposition are needed to detect the watermark.

Wang *et. al* [59] introduce a blind watermarking (oblivious detection) scheme, in which embedding scheme searches for perceptually significant wavelet coefficients onto which the watermark coefficients are added. In Ref. [60], two watermarking schemes modeled after the EZW [61] compression scheme are presented and

compared. While one algorithm uses the "insignificant coefficients", the other uses "significant" coefficients.

2.3.3 RST Invariance

In Ref. [62] the authors introduce a rotation, scale and translation (RST) invariant watermarking scheme. The RST invariance is achieved as follows. Translation invariance is achieved by taking the DFT of the image and using only the DFT magnitude. The DFT magnitude is then mapped to log-polar coordinates. Translation invariance in the log polar domain corresponds to scaling and rotational invariance in the spatial domain. Thus taking the 2-D DFT of the log-polar mapping and retaining only the magnitude, results in an RST invariant domain. The signature is added to the RST invariant domain and then mapped back to the log-polar domain (using the original unmodified phase). The log polar mapping is now mapped back to the 2-D DFT magnitude coefficients of the image. Again the original phase of the image is retained and a inverse 2-D DFT performed to obtain the watermarked image.

2.3.4 Other Methods

Ruanaidh *et. al.* [63] propose a watermarking scheme, where only the phase of the DFT coefficients (2-D DFT) of the image are modified to embed the signature. The watermark is embedded in the phase of significant DFT coefficients. The authors claim that information in DFT phase is superior for the same reason that angle modulation is expected have better noise immunity than amplitude modulation in communications theory. In Ref. [64] Fridrich *et. al* introduce a decomposition based on random keys. In this scheme, a set of random smooth patterns are generated from a key. These patterns are then subject to Gram-Schmidt orthogonalization process to obtain a set of smooth orthogonal patterns which are used to embed the watermark in the image.

In Ref. [65], the difference between a pixel and the average value of its four adjacent pixels is modified to embed a bit. Note that this is equivalent to low pass filtering and modifying the high pass coefficients to embed a bit. The main advantage of this method is that this scheme will be relatively unaffected by histogram equalization.

In Ref. [66], Paute *et. al* combine watermarking scheme with fractal or IFS compression scheme. The signature is added by restricting the choice of 'domain blocks' depending on the bit to be enclosed. The robustness of the embedding increases as the "range' block sizes increase. However this will result in poorer quality of compression, and the resulting image may not be of acceptable quality.

In Ref. [67, 68] Voyatzis and Pitas apply nonlinear dynamical principles to watermarking images. The watermark extracted is usually a logo with very few gray levels. The logo is mapped to $N \times N$ lattice which is less than the size of the image. The lattice is mixed with the image. Extraction of the watermark is performed by repeated application of an auto-morphism to extract the logo.

2.4 Video Steganography

While data hiding in video, can be done by considering each frame as an image, efficient watermarking schemes should take into account the differences between the nature of images and video frames. As video data is much more redundant than image data, they are susceptible to a wider variety of attacks like frame averaging, frame dropping etc.. To account for the peculiar nature of the possible attacks on video frames, Swanson *et. al.* [69] present a watermarking scheme, in which the watermark is embedded in objects (8×8 blocks of frames). In smooth regions of the image use a constant watermark while the motion regions use dynamic watermarks.

In Ref. [70, 71] Hartung *et. al.* propose schemes to encode raw and compressed (MPEG compressed) video. For watermarking in the raw domain they use a spread

spectrum sequence as the watermark. For watermarking in the MPEG compressed domain, the MPEG bit-stream is separated into header, side information, motion vectors and DCT encoded signal blocks. The Huffman coded DCT coefficients are decoded, and then inverse quantization is applied. The DC coefficients are modified to add the watermark and reinserted into the MPEG bit-stream.

CHAPTER 3

LINEAR DATA HIDING

3.1 Introduction

Most of the state-of-the-art techniques for data-hiding in images utilize some decomposition for embedding the message bits. Among different orthonormal decomposition techniques, it was probably the inspiration from image compression applications that caused DCT and subband (wavelet) transforms to be more popular than the others. Another reason for the choice of DCT and wavelet based techniques is perhaps to 'match' the data hiding [72] technique with the processing the image is most likely to undergo. Currently, the most common image compression tools are the DCT based JPEG, and the wavelet based SPIHT / EZW [13] coding techniques. Adding the signature or the message signal *intelligently* (for example taking the JPEG quantization tables into account) in the DCT domain can insure robustness to JPEG. Similarly, one could design wavelet based methods robust to EZW / SPIHT compression. It is no surprise that most wavelet based methods are very robust to EZW or SPIHT compression [59], but are not very robust to JPEG. Similarly, DCT based methods are robust to JPEG, but not to EZW / SPIHT. Of course, one cannot expect robustness of these methods to other forms of compression / signal processing. Though it is true that most images are very likely to go through Wavelet / DCT based compression, the situation is different for video frames. For most video frames the major source of 'information' is the motion vectors. So it is difficult to *intelligently* devise DCT / Wavelet based methods for data hiding in video frames.

It is of great interest therefore, to devise robust data hiding methods given that no knowledge of the compression technique to be employed, is available. Now the question to be answered is, what is underlying decomposition that should be used? In this chapter, we attempt to answer that question. We provide an information

theoretic approach to estimate the achievable capacities for different orthonormal decompositions like DCT, subband, DFT, Hadamard and Hartley transforms.

Several authors, [73, 74, 75] have proposed information theoretic approaches to characterize or evaluate the performance of the data hiding channel. In [73], Smith *et al.* model the image as a Gaussian noise source of variance given by the average noise (image) power. The data hiding capacity is then calculated as the capacity of the Gaussian channel. In [74] Servetto *et al.* obtain the capacity of the data hiding channel where the source of noise is intentional jamming. However, it is assumed that the original image is available at the receiver. The work of Hernandez *et al* [75] is a more thorough model, which analyses the performance of a proposed method for data hiding. In this model, L orthogonal sequences are used for the signature. The image is broken down into channels corresponding to its projections onto each of the orthogonal signatures. The capacity of the channels are analyzed for unprocessed images and images after linear filtering operations.

3.2 Problem Statement

Let $\mathbf{I}$ be the original (cover) image, to which a message $\mathbf{S}$ (a representation for embedded information bits) is added, such that $\hat{\mathbf{I}} = \mathbf{I} + \mathbf{S}$. The modified image $\hat{\mathbf{I}}$, is *visually indistinguishable* from $\mathbf{I}$ and may typically be subjected to lossy compression, like JPEG, $\tilde{\mathbf{I}} = \mathcal{C}(\hat{\mathbf{I}})$, where $\mathcal{C}(.)$ denotes the compression / decompression operation. The embedded bits in image $\mathbf{I}$ are to be extracted from $\tilde{\mathbf{I}}$. We would like to know the maximum number of bits that can be hidden and recovered from the image with an arbitrarily low probability of error, namely, the *capacity of the data-hiding channel*, for a given compression scenario.

A block diagram of the data-hiding channel is shown in Figure 3.1. $\mathbf{S}$ is the message (signature) to be transmitted through the channel. The channel has two sources of noise; $\mathbf{I}$, the noise due to the (original) cover image, and $\mathbf{P}$, the noise

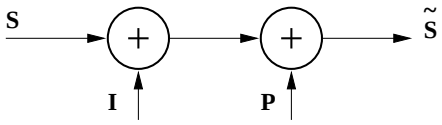

Figure 3.1 The data hiding channel

component due to processing (compression / decompression). $\tilde{\mathbf{S}}$ is the "corrupted" message. Note that for the cover image escrow methods, there is only one source of noise - due to processing. The image noise can be subtracted from the received image $\tilde{\mathbf{I}}$. One can expect such methods to have higher capacity than the oblivious detection methods.

Figure 3.2 displays the block diagram of a typical data-hiding method. The forward transform block decomposes the image $\mathbf{I}$ into its coefficients of L bands. A component of the signature / message signal is added to each band. The inverse transform block reconstructs the modified image $\hat{\mathbf{I}}$.

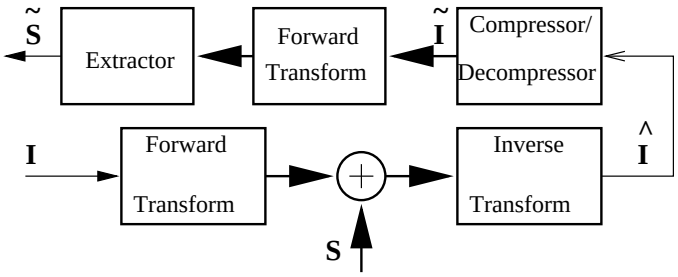

Figure 3.2 Generalized schematic of data hiding / retrieval

The image $\hat{\mathbf{I}}$ then undergoes some processing (lossy compression) to yield the image $\tilde{\mathbf{I}}$. The hidden message signal / signature is to extracted from $\tilde{\mathbf{I}}$. The image $\tilde{\mathbf{I}}$ is decomposed into L bands by the same forward transform block and each component of the signature is extracted separately. In this chapter, we assume the system of Figure 3.2 and estimate the capacity of data-hiding channel for different decompositions (different forward and inverse transform blocks).

3.3 Capacity of Additive Noise Channels

Prior to considering the data-hiding channel of Figure 3.1, we consider the simpler channel displayed in Figure 3.3(a). $\mathbf{X} \sim [f_X(x), \sigma_x^2]$ is the message signal to be transmitted, $\mathbf{Z} \sim [f_Z(z), \sigma_z^2]$ is the additive noise in the channel, and $\mathbf{Y} \sim [f_Y(y), \sigma_y^2]$ is the received signal at the output of the channel.

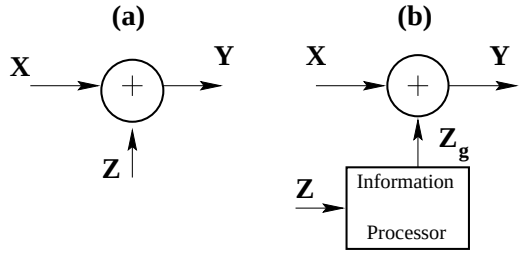

Figure 3.3 (a) A simple additive noise channel. (b) The channel of (a) modified to obtain equivalent additive Gaussian noise.

We also assume that $\mathbf{X}$ and $\mathbf{Z}$ are independent, implying that $\sigma_y^2 = \sigma_x^2 + \sigma_z^2$. Therefore, the channel capacity is given by [76]

$$\mathbf{C} = \max_{f_X(x)} I_M(\mathbf{X}, \mathbf{Y}) = \max_{f_X(x)} h(\mathbf{Y}) - h(\mathbf{Y}|\mathbf{X}) = \max_{f_X(x)} h(\mathbf{Y}) - h(\mathbf{Z}) \text{ bits.} \tag{3.1}$$

where $I_M(\mathbf{X}, \mathbf{Y})$, is the *mutual information* between $\mathbf{X}$ and $\mathbf{Y}$. For a given noise statistics $f_Z(z)$ and input variance σ_x^2, one can maximize the entropy of the output $\mathbf{Y}$,

$$h(\mathbf{Y}) = -\int f_Y(y) \log_2(f_Y(y)) \mathrm{d}y \text{ bits,} \tag{3.2}$$

by choosing a suitable distribution $f_X(x)$ for the input message $\mathbf{X}$. For a given variance σ_y^2, the maximum entropy value of $h(\mathbf{Y}) = \frac{1}{2}\log_2(2\pi \mathrm{e}\sigma_y^2)$ bits is achieved when $\mathbf{Y}$ has a normal distribution. For instance, the maximum entropy value is achievable if both pdfs $f_Z(z)$ and $f_X(x)$ are normally distributed. However, for an arbitrary distribution $f_Z(z)$, and a fixed σ_x^2, the maximum achievable entropy value is not immediately obvious. To calculate that, we pass the noise $\mathbf{Z}$ through an ideal *information processor*, (see Figure 3.3(b)) which does not alter the amount of

information in $\mathbf{Z}$, but changes its statistics to a Gaussian distribution for its output $\mathbf{Z}_g$. (The information processor can be considered as an ideal data compressor, where 'compression' is measured in terms of signal energy. The information processor translates the data to a form which has minimum energy while maintaining the information content or *entropy*). Since the output of the information processor has the same entropy as the input, the variance of the output, σ^2_{zg}, can be obtained by solving

$$h(\mathbf{Z}_g) = h(\mathbf{Z}) = \frac{1}{2}\log_2(2\pi\text{e}\sigma^2_{\text{zg}}) \text{ bits.} \tag{3.3}$$

It is well known that the Gaussian distribution has the highest entropy for a given variance [76]. Alternately, the Gaussian distribution has the least variance for a given entropy. Thus it is always true that $\sigma^2_{\text{zg}} \leq \sigma^2_z$. We call σ^2_{zg} the *entropy equivalent Gaussian variance*. The maximum value of $h(\mathbf{Y})$ is therefore obtained as

$$\max_{f_X(x)} h(\mathbf{Y}) = \max_{f_X(x)} h(\mathbf{X} + \mathbf{Z}_g) = \frac{1}{2}\log_2(2\pi\text{e}(\sigma^2_{\text{zg}} + \sigma^2_x)) \text{ bits.} \tag{3.4}$$

In order to calculate the channel capacity, we can now replace $f_Z(z)$ by $N[0, \sigma^2_{\text{zg}}]$.

$$\mathbf{C} = \max_{f_X(x)} h(\mathbf{Y}) - h(\mathbf{Z}_g) = \frac{1}{2}\log_2(1 + \frac{\sigma^2_x}{\sigma^2_{\text{zg}}}) \text{ bits.} \tag{3.5}$$

Note that if processing noise is Gaussian and independent of the image noise, the two channel noise sources in Figure 3.1 can be replaced by a single Gaussian noise source of variance $\sigma^2_{\text{ig}} + \sigma^2_p$, where σ^2_{ig} is the equivalent Gaussian variance for the image noise $\mathbf{I}$, and σ^2_p is the variance of the processing noise. If σ^2_s is the message signal energy, the capacity of the data-hiding channel can be expressed as

$$\mathbf{C}_h = \frac{1}{2}\log_2(1 + \frac{\sigma^2_s}{\sigma^2_{\text{ig}} + \sigma^2_p}) \text{ bits.} \tag{3.6}$$

As a first approach to calculate the capacity of the data-hiding channel, the image noise $\mathbf{I}$ (the original image pixels) is assumed to be uniformly distributed random variables i taking values between 0 and 255 with variance σ^2_i. Let σ^2_p be the

variance of the noise (per pixel) introduced due to processing, (*e.g.* compression). As we shall see later, the processing noise is an *estimate* of the variance of an *equivalent additive noise* which substitutes the actual non-linear processing noise sources (mainly quantization for the case of lossy compression). Since we do not know anything about the distribution of the equivalent processing noise, we assume the worst - Gaussian distribution. Finally, let σ_s^2 be the average energy per pixel allowed for the message signal. If MN is the number of pixels in an image, then the energy (or variance if zero-mean) of the message signal is calculated as

$$\sigma_s^2 = \frac{\sum_{i=1}^{MN} S_i^2}{MN}, \tag{3.7}$$

where, S_i is the message signal added to the i^{th} pixel. The (differential) entropies, $h(g)$, of a Gaussian random variable g, with variance of σ_g^2, and $h(u)$, that of a uniformly distributed random variable u with variance σ_u^2 are expressed as [76]

$$h(g) = \tfrac{1}{2}\log_2(2\pi \mathrm{e}\sigma_g^2)\,\text{bits} \quad h(u) = \tfrac{1}{2}\log_2(12\sigma_u^2)\,\text{bits}.$$

From Eq. (3.8), the *entropy equivalent Gaussian noise* (or the Gaussian random variable that has the same entropy as the uniform random variable u of variance σ_i^2), has a variance given by

$$\sigma_{\text{ig}}^2 = \frac{12}{2\pi \mathrm{e}}\sigma_i^2. \tag{3.8}$$

Although we would expect the variance of u, the pixel values, to be given by $\sigma_i^2 = \frac{255^2}{12}$ (or $\sigma_i = 73.6$), statistics from many test images (see Section 4 for the details of the test images used) show that $\sigma_i = 55$. Therefore, we assume that u has a uniform distribution with $\sigma_i = 55$. From Eq. (3.8) it is calculated that $\sigma_{\text{ig}} = 55(\frac{12}{2\pi e})^{0.5} \approx 46$. If we allow a degradation of the image after the addition of a message to a PSNR of 40 dB, then the message energy is calculated to be $\sigma_s^2 = 6.5$. Furthermore, if the image goes through JPEG compression at 50% quality, then it is measured for test images that the processing noise has a standard deviation of $\sigma_p \approx 6.7$ (the actual procedure

for estimating processing noise is described in Section 3.4.2). This would yield a capacity $\mathbf{C}_h$ value of 0.0022 bits/pixel (140 bits for a 256×256 image). Even if the message-embedded image undergoes some other processing which results in a barely recognizable image corresponding to $\sigma_p \approx 20$, the capacity $\mathbf{C}_h$ would still be 0.0019 bits per pixel (about 124 bits for a 256×256 image). Therefore, one can see that hiding the message in the image domain can be very robust. However, in most cases, we do not require such robustness. Since most data-hiding applications aim to protect and ascertain copyright or control access, it is unlikely in such a scenario that anyone would want to claim ownership or control access of an image of no commercial value (an image which has been significantly degraded in perceptual quality). Typically, it is sufficient if the message survives well-known image compression/ decompression operations with acceptable quality.

Given that we are satisfied with less robustness than the above mentioned method offers, could we do better than this? In our first approach, what we have done is very similar to the method reported in [73] (the only difference being that we have also introduced processing noise in the channel). By assuming a Gaussian channel, we assume that the image pixels have a flat spectrum. However, it is well known that the spatial frequency characteristics of a typical image is far from flat (white). Most of the image energy is concentrated in the low-frequency bands. It is therefore intuitive that a decomposition of the image into its different frequency bands might help. We expect the low frequency bands of the decomposition to very noisy due to the high energy content of the image. On the other hand, high frequency components would be very vulnerable to processing, as most compressors would discard them at low bit-rates. At mid-frequency bands, however, we could strike a compromise. A typical distribution of image and processing noise in various bands of a decomposition is shown in Figure 3.4.

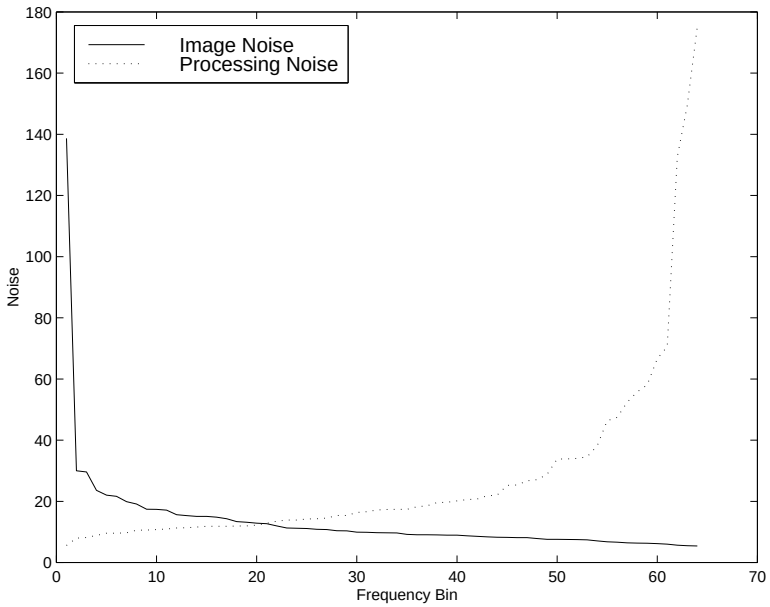

Figure 3.4 A Typical Distribution of Image and Processing Noise Among Different Bands

In Figure 3.5, the channel of Figure 3.1 is decomposed into its multiple sub-channels. The decomposition is performed by the Forward and Inverse Transform blocks of Figure 3.2. The decomposition of an image into its L sub-bands results in L parallel sub-channels with two noise sources in each sub-channel. Let $\sigma_{i_j}^2$, $j = 1 \cdots L$, be the variances of the coefficients for each sub-band (or the variances of the image noise in each sub-channel) of the decomposition. Similarly, let their corresponding equivalent Gaussian variances be $\sigma_{\mathrm{ig}_j}^2$. If $\sigma_{p_j}^2$ is the variance of the processing noise (Gaussian) in the j^{th} sub-channel, then, the total capacity of the L parallel sub-channels is given by

$$\mathbf{C}_h = \frac{MN}{2L} \sum_{j=1}^{L} \log_2(1 + \frac{v_j^2}{\sigma_{\mathrm{ig}_j}^2 + \sigma_{p_j}^2}) \,\mathrm{bits} \tag{3.9}$$

for an image of size MN pixels. In Eq. (3.9), v_j is the *visual threshold* of band j. In other words, v_j^2 is the maximum message signal energy permitted in band j based on its perceptual quality effects. Note that if the channel was a purely energy constrained channel (or if the constraint is on the total signature energy with no

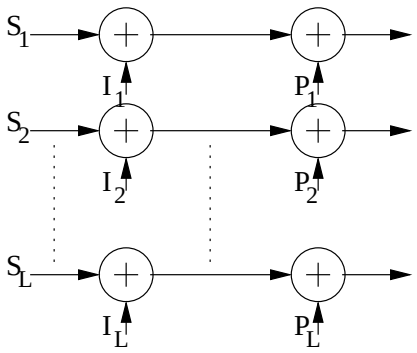

Figure 3.5 Decomposition of the data hiding channel into parallel channels

regard to how the signature energy is distributed among different bands), then the best solution would be to use the water-filling approach [76] to calculate the channel capacity. However in this case, the maximum signal energy permitted in a channel is constrained by the visual threshold of the band. Ideally, we would like to utilize all channels to the fullest extent possible.

In the following sections, we evaluate the capacity of the data-hiding channel for DCT, DFT, Hadamard, and uniform subband decomposition based embedding methods. We use well-known compression methods like JPEG and SPIHT to model the processing (compression) noise in each sub-band of the decomposition.

3.4 Modeling Channel Noise

In order to model the channel noise (the two noise sources **I** and **P** in Figure 3.1), we measure their statistics from 15 monochrome test images of size 256×256, and their JPEG and SPIHT compressed versions at various quality factors / bit rates. The 15 test are shown in Figure 3.6.

3.4.1 Modeling Image Noise

The cover images are decomposed into L sub-bands using an orthonormal transform. Let $f_{I_j}(i_j)$ be the distribution of the j^{th} sub-band with variance $\sigma_{i_j}^2$. (The image

Figure 3.6 The 15 256×256 test images used

noise $\mathbf{I}$ is split into its components in L sub-channels, which are modeled as random variables $f_{I_j}(i_j)$ with variances $\sigma^2_{i_j}$, $j = 1 \cdots L$.)

Having obtained the variances of the image noise in each sub-channel, the next step is to obtain their entropy equivalent Gaussian variances. This is achieved by plotting a histogram of the coefficients for each band, and calculating the entropy. If Δx is the width of the n bins of the histogram $g_j(m)$, $m = 1 \cdots n$, and p is the total number of coefficients in band j, the entropy $\mathcal{H}_j$ and the equivalent Gaussian variance $\sigma^2_{\mathrm{ig}_j}$ of the sub-band are obtained as

$$\mathcal{H}_j = -\sum_{i=1}^{n} \frac{g_j(i)}{p\Delta x} \log_2\left(\frac{g(i)}{p\Delta x}\right)\Delta x, \text{ bits} \quad \sigma^2_{\mathrm{ig}_j} = \frac{2^{2\mathcal{H}_j}}{2\pi \mathrm{e}}.$$

Thus, the image noise in sub-channel (band) j can be substituted by a Gaussian noise of variance $\sigma^2_{\mathrm{ig}_j}$. In our simulations, the image noise is estimated for each image individually for five different transforms.

3.4.2 Modeling Processing Noise

At the outset, one should note that Processing noise is introduced due to quantization of transform domain parameters. While one could accurately estimate the type of quantization noise that is introduced by JPEG on the DCT coefficients of the image (assuming that the quantization table is known), the same cannot be done, for instance, for the Hadamard transform coefficients of the image. The quantization of one DCT coefficient would affect many Hadamard coefficients. More importantly, for the reasons explained earlier, *viz.* we wish to make the model of the processing noise more general. The only reason we restrict ourselves to JPEG and SPIHT for processing noise sources is their widespread availability. We define processing noise as the *equivalent additive noise which accounts for the reduction in correlation between the transform coefficients of the original image and the transform coefficients of the image obtained after lossy compression.* Note that while this estimate provides us with the *variance* of the equivalent additive noise, it does not tell us anything about the nature of the noise (like its distribution). We therefore assume the worst - Gaussian distribution for the processing noise.

Let the processing noise in each sub-channel be $\sigma_{p_j}^2$, $j = 1 \cdots L$. The steps to obtain the processing noise variance are:

- Apply lossy compression / decompression (JPEG / SPIHT at various quality factors / bit rates) to n_i test images.
- Decompose the n_i test images using some transform.
- Obtain $\frac{MNn_i}{L}$ samples for each sub-band. Let i_{j_k}, $k = 1, \ldots, \frac{MNn_i}{L}$, be the coefficients of band j.
- Decompose the n_i reconstructed images using the same transform.
- Let $\tilde{i}_{j_k}$, $k = 1, \ldots, \frac{MNn_i}{L}$ be the corresponding coefficients of the images subjected to lossy compression

- Define the intra-band correlation as

$$\frac{\langle \mathbf{i}_j, \tilde{\mathbf{i}}_j \rangle}{|\mathbf{i}_j||\tilde{\mathbf{i}}_j|} = \frac{\langle \mathbf{i}_j, (\mathbf{i}_j + \mathbf{n}_j) \rangle}{|\mathbf{i}_j||\mathbf{i}_j + \mathbf{n}_j|} = \rho_j, \tag{3.10}$$

  where $\mathbf{n}_j$ is a vector of random variables, uncorrelated with $\mathbf{i}_j$.

- $\sigma_{n_j}^2 = |\mathbf{n}_j|^2$ is the variance of the *equivalent additive noise due to compression* (or $\sigma_{p_j} = \sigma_{n_j}$).

- Since $\langle \mathbf{i}_j, \mathbf{n}_j \rangle = 0$, Eq. (3.10) can be simplified to obtain

$$\sigma_{p_j}^2 = |\mathbf{n}_j|^2 = (\frac{1}{\rho_j^2} - 1)|\mathbf{i}_j|^2 \tag{3.11}$$

It can be easily seen that the processing noise in each sub-band *can not* be obtained as $\tilde{i}_{j_k} - i_{j_k}$. Consider a scenario, where DCT is used for the decomposition, and low quality JPEG for processing. Let us assume that a high frequency sub-band is completely removed due to compression ($\tilde{i}_{j_k} = 0 \, \forall \, k$ for some j). This implies that all information buried in that sub-channel (sub-band) is lost. In other words, the processing noise in that sub-channel has *infinite* variance (and *not*) the variance of $\tilde{\mathbf{i}}_\mathbf{j}$. This is because no *correlation* exists between $\tilde{i}_{j_k}$ and i_{j_k}. Note that in Eq. (3.11) when $\rho_j \to 0$, $\sigma_{p_j} \to \infty$.

Also, note that while the image noise is estimated individually for each image, the processing noise is not. There are two reasons for this:

- As the equivalent image noise is estimated by correlation, the result is likely to be more accurate if more samples are used. If we calculate processing noise for each image separately, (for 256×256 images using some 64 band decomposition), we have only 1024 coefficients in each band. However, using 15 images yields 1024×15 coefficients per band.

- The second reason is that this method of estimating the processing noise would yield unrealistic (very low) estimates of processing noise for low entropy images.

The original and compressed versions of low entropy images are bound to be very 'close', leading to high correlation in most bands. This would cause an overestimate of capacity for smooth images. To mitigate this effect we average processing noise over many images.

3.5 Visual Threshold

The value of the *visual threshold* for sub-channel j, v_j in Eq. (3.9) however, is highly subjective. Since the amount of message signal energy permitted in any sub-band is determined by the visual threshold, different models for visual thresholds would yield different estimates of achievable capacity. The visual threshold depends not only on the band, but also on the magnitude of the particular coefficient. Within the same band, a coefficient with high magnitude can be altered to a larger extent than a coefficient with small magnitude. Additionally, the visual threshold may also depend on the magnitudes of coefficients of other bands corresponding to the same block / spatial location.

However, what we desire is an estimate of the *average* energy of the message signal that can be added to a particular band. Since it is well known that the human visual system is more sensitive to the lower frequencies than the higher frequencies, the signal-to-noise-ratio (message signal to image noise) should be smaller for lower frequency sub-bands. In general lower frequency sub-bands have higher variances. Hence, a reasonable model for the visual threshold v_j could be

$$v_j^2 = K\sigma_{i_j}^{2\alpha} \tag{3.12}$$

where $0 < \alpha < 1$, and $K << \sigma_{i_j} \, \forall j$, is a constant. When $\alpha = 0$, the message signal energy is distributed equally among all sub-bands regardless of their variances. On the other hand, when $\alpha = 1$ the message signal energy is distributed in the ratio of the band variances.

From Eqs. (3.9) and (3.12), for the case of *no processing noise*, if we assume that all sub-channels have the same pdf type (such that $K\sigma_{i_j} = K_1\sigma_{\mathrm{ig}_j}$), the channel capacity can be calculated as

$$\mathbf{C}_h = \frac{MN}{2L}\sum_{j=1}^{L}\log_2(1+\frac{K_1\sigma_{\mathrm{ig}_j}^{2\alpha}}{\sigma_{\mathrm{ig}_j}^2}) \approx \frac{MN}{2L}\log_2\left(1+\sum_{j=1}^{L}\frac{K_1}{\sigma_{\mathrm{ig}_j}^{2(1-\alpha)}}\right), \tag{3.13}$$

In the above equation, the approximation is justified because $\frac{K_1\sigma_{\mathrm{ig}_j}^{2\alpha}}{\sigma_{\mathrm{ig}_j}^2} << 1 \forall j$. Note that for the case of $\alpha = 1$, the decomposition does not have any effect on the capacity. However, for $\alpha < 1$, $\mathbf{C}_h$ can be increased by choosing a suitable transform, as shown in the next section. Thus, the increase in capacity is due to the fact that one can add *relatively* more message signal energy to bands of lower variances (or high frequency bands).

However, in Eq. (3.12) there seems to be no rationale for fixing the value of α apart from actual simulations. We therefore adopt a different model for visual threshold. To derive the model, we argue that JPEG, at a reasonably good quality factor is well tuned visually in distributing the quantization errors amongst the bands, at least with respect to preserving the visual fidelity of the compressed image. More advanced methods like SPIHT tend to optimize the mean square error rather than visual fidelity (in general, the visual quality of a JPEG compressed image at a certain PSNR is much better than that of a SPIHT compressed image at the same PSNR). Let i_{j_k} be the coefficients of the original images, and $\tilde{i}_{j_k}$ the coefficients of the same images that have gone through JPEG-75 (quality factor 75) compression and decompression. Let $\sigma_{q_j}^2$ be the variance of the quantization error, $\mathbf{e}_{q_j} = \tilde{\mathbf{i}}_j - \mathbf{i}_j$, for sub-band j. If quantization error (due to JPEG-75) of variance $\sigma_{q_j}^2$ in sub-band j, results in an image that is *visually satisfactory*, we can argue that addition of message signal with energy $\sigma_{q_j}^2$ in sub-band j, would still render the image $\hat{\mathbf{I}}$ with an acceptable visual quality. However, in order to maintain the PSNR of $\hat{\mathbf{I}}$ in the range of 40-50 dB (so that the $\hat{\mathbf{I}}$ is visually indistinguishable from $\mathbf{I}$), we choose the sub-band visual

thresholds as

$$v_j^2 = K_2 \sigma_{q_j}^2 \tag{3.14}$$

where $K_2 < 1$. (The average PSNR of JPEG-75 images is only about 35 dB. Hence a choice of $K_2 = 1$ would yield images $\hat{\mathbf{I}}$ of PSNR 35 dB. This might not be an acceptable quality. For our simulations we use $K_2 = 0.25$.)

3.6 Channel Capacity vs Choice of Transform

It should be noted that both Eqs. (3.9) and (3.13), are subject to the following constraints

$$\sum_{j=1}^{L} \sigma_{i_j}^2 = L\sigma_i^2 \quad \sum_{j=1}^{L} \sigma_{\mathrm{ig}_j}^2 = L\sigma_{\mathrm{ig}}^2 \quad \mathcal{I} = \tfrac{1}{2}\log_2(2\pi \mathrm{e}\sigma_{\mathrm{ig}}^2)$$

where σ_i^2 is the variance of images, σ_{ig}^2 is the entropy equivalent Gaussian variance for σ_i^2, and $\mathcal{I}$ is the average entropy of image pixels. The first equation states that unitary transforms (the transforms used for the embedding decompositions) preserve energy. The second and third equations state that the transforms also preserve entropy. With the above constraints, it can be shown that the *minimum* channel capacity (for the case of *no processing noise* or Eq.(3.13)) is achieved for $\sigma_{\mathrm{ig}_j} = \sigma \, \forall j$, or when no decomposition (spatial embedding) is used.

Note that a transform with good energy compaction or higher Transform Coding Gain (GTC) [77] would result in more *imbalance* of the coefficient variances. This would enhance the term $\sum_{j=1}^{L} \frac{K_1}{\sigma_{\mathrm{ig}_j}^{2(1-\alpha)}}$ in Eq. (3.13), and therefore increase the capacity (when the processing noise is small). Therefore, good energy compaction transforms like DCT and subband transforms are good embedding decompositions for *low processing noise scenarios.*

However, the relationship between processing noise and the choice of transform is not immediately obvious. For example if we use JPEG at low quality factor for compression and DCT as the embedding decomposition, it is very easy to see

that the processing noise will approach infinity for many high frequency bands as they are bound to be completely eliminated. On the other hand, the high frequency coefficients of say Hadamard transform will have components in many DCT coefficients. So it is not very likely that any Hadamard transform band is completely eliminated. In fact, even if the processing the image undergoes is SPIHT, it is still more likely to affect the high frequency DCT coefficients more than the high frequency Hadamard transform coefficients. Any efficient compression method would affect the low variance (high frequency) bands of the transforms suitable for compression (or high GTC transforms).

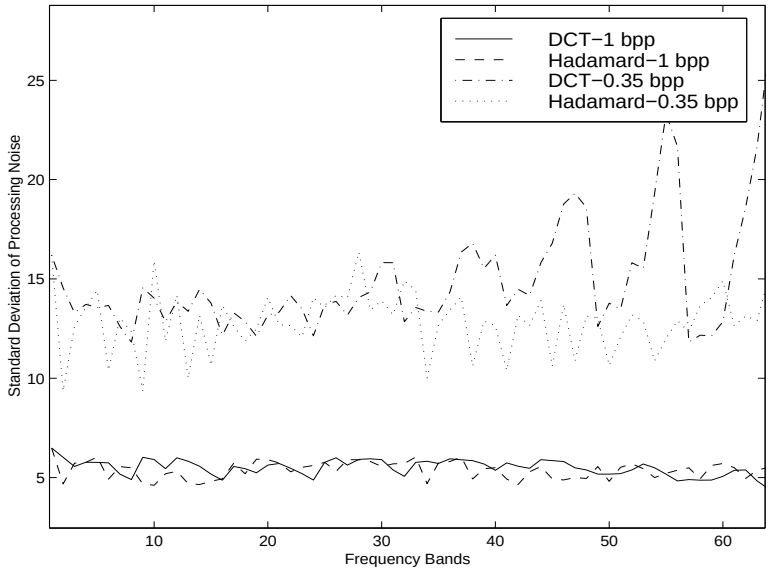

Figure 3.7 Comparison of standard deviations of processing noise for DCT and Hadamard decompositions. The source of processing noise is SPIHT compression at 1 bpp and 0.35 bpp.

To illustrate this point Figure 3.7 shows the distribution of the processing noise for DCT and Hadamard transform bands for processing noise due to SPIHT at 1 bpp and 0.35 bpp. While the processing noise for the two decompositions are comparable for SPIHT at 1 bpp, it is seen that processing noise increases drastically for high frequency DCT bands for SPIHT at 0.35 bpp. The high frequency bands of Hadamard transform, however, are relatively immune to processing noise. Similarly

low quality JPEG affects the high frequency bands of subband decomposition (using 8-tap Daubechies filter) to a much larger extent than the high frequency Hadamard bands. We already know that low frequency bands are not efficient channels due to the presence of high image noise. If the high frequency bands are also affected by processing, it leaves a small number useful of mid-frequency bands. Transforms with lower GTC have many more of this useful 'mid-frequency' bands than the high GTC transforms, at higher processing noise scenarios. Therefore, *decompositions unsuitable for compression would in general be more immune to processing noise than decompositions with high GTC*. Also, recall that in Section 2 embedding in the image domain (or using identity transform for the transform blocks in Figure 3.2), was found to be very robust to processing noise. The identity transform, which has the lowest GTC has the highest robustness to processing noise. It is relevant to point out here that the term 'robustness', is a measure of the change in *overall capacity* with a change in the processing noise (or processing scenario). More robust the decomposition, less is the reduction in capacity for a scenario of increased processing noise (or lower quality compression). One should note that the robustness of the *low frequency bands* of say the DCT decomposition will be much higher than the robustness of the single band coefficients (pixels) in the image domain. However the low frequency bands of the DCT have very little capacity due to high image noise. The reduced 'robustness' of DCT is due to the drastic reduction in the *overall capacity* due to the drastic increase of processing noise in the high frequency bands.

The next question that arises is the choice of the number of bands for the decomposition. From Eq. (3.13) we see that a decomposition will not hurt. At worst, it may cause no improvement. Therefore decomposing each sub-channel of say a 16 band decomposition further into four sub-channels can only improve the capacity of data hiding, at least when processing noise is low.

3.7 Results

The estimated capacities for different 64 band decompositions (for 256×256 images, or 65536 pixels) like DFT, DCT, subband, Hartley and Hadamard transformations, are shown in Figure 3.8. The capacities were estimated for 5 different transforms for 8 different processing scenarios and averaged over 15 images. Figures 3.9 and 3.10 show the individual capacities of 4 different images (Baboon, Barbara and Lena, Bridge).

Figure 3.11 shows the average channel capacities of each video frame of 3 video sequences (Table Tennis, Football and Garden) averaged over 90 frames per sequence. The source of processing for the video sequences is MPEG-2 compression (30 frames/ sec, 15 frames in GOP and I/P frame distance of 3), at various bit-rates. In Figure 3.11, the left column is the estimates of capacity of I-Frames and the right column for P/B-Frames.

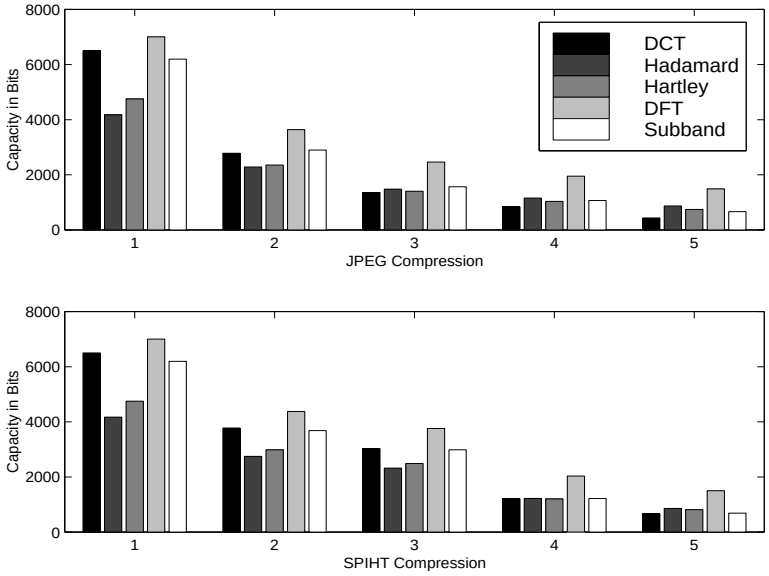

Figure 3.8 Average capacity estimates for 15 256×256 images. The indices for JPEG compression correspond to different JPEG quality factors. (1 - lossless compression, 2 - 75%, 3 - 50%, 4 - 35%, 5 - 25%) The indices for SPIHT compression correspond to different bit rates (1 - lossless , 2 - 1 bpp, 3 - 0.75 bpp, 4 - 0.5 bpp, 5 - 0.35 bpp).

For the subband decomposition we use the 8-tap Daubechies filter (though it would be better idea to use the linear phase 9-7 filters used more commonly

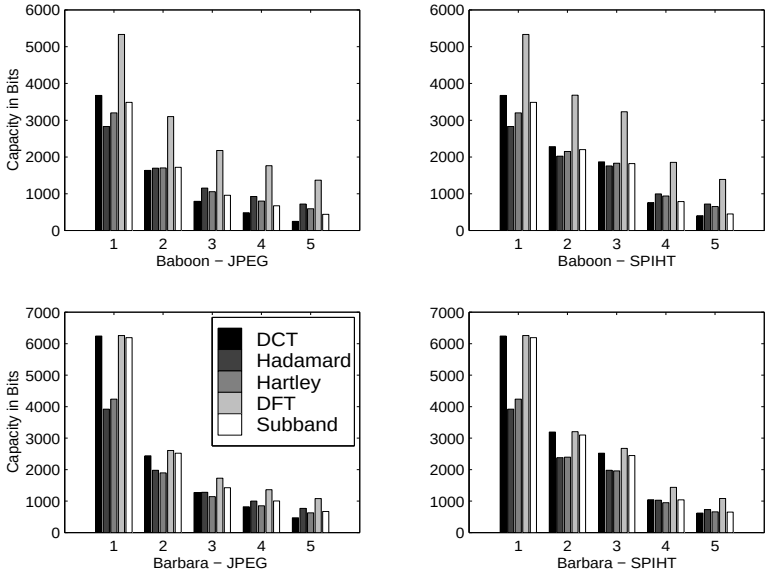

Figure 3.9 Capacity estimates for 256×256 Baboon and Barbara images. The indices for JPEG compression correspond to different JPEG quality factors. (1 - lossless compression, 2 - 75%, 3 - 50%, 4 - 35%, 5 - 25%) The indices for SPIHT compression correspond to different bit rates (1 - lossless , 2 - 1 bpp, 3 - 0.75 bpp, 4 - 0.5 bpp, 5 - 0.35 bpp).

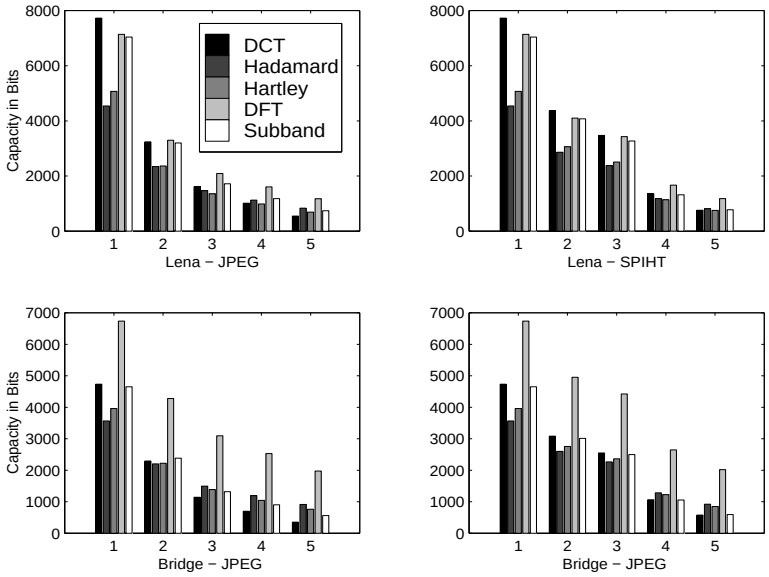

Figure 3.10 Capacity estimates for 256×256 Lena and Bridge images. The indices for JPEG compression correspond to different JPEG quality factors. (1 - lossless compression, 2 - 75%, 3 - 50%, 4 - 35%, 5 - 25%) The indices for SPIHT compression correspond to different bit rates (1 - lossless , 2 - 1 bpp, 3 - 0.75 bpp, 4 - 0.5 bpp, 5 - 0.35 bpp).

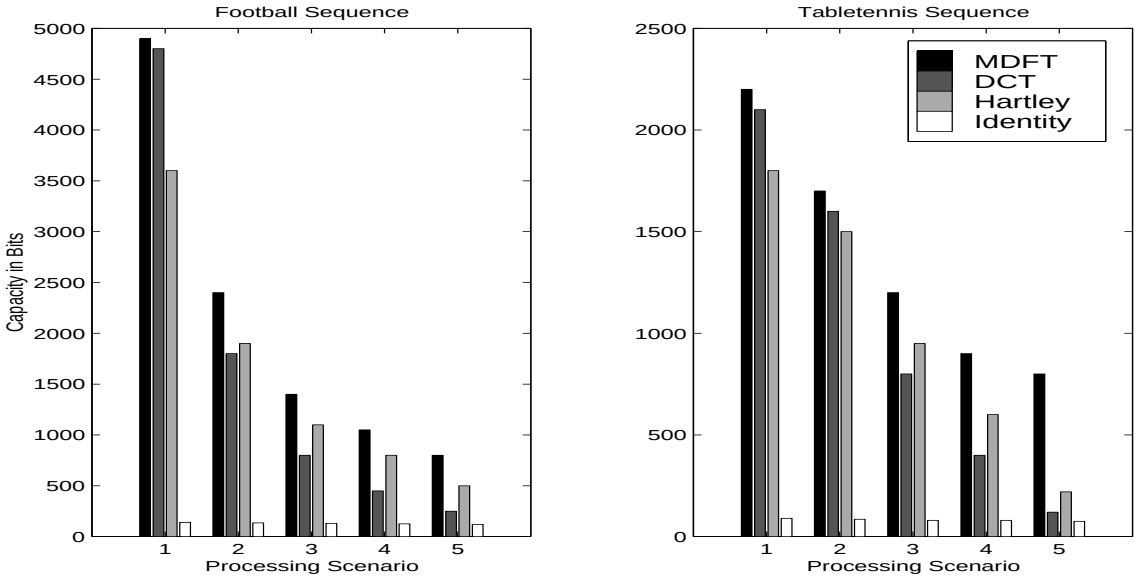

Figure 3.11 Channel capacities of different decompositions for Football and Tabletennis Sequences. The processing scenarios 1-5 correspond to lossless compression, and compression ratios of 10, 25, 50 and 100 (MPEG-2) respectively.

for subband or wavelet image compression, the *biorthogonality* of the filters would complicate the analysis). More specifically, we use uniform subband decomposition. For the DFT decomposition we use only the *magnitude* of the DFT coefficients. The phase is ignored. (In other words, the message signal added would change only the magnitude of the DFT coefficients. The phase is left intact. As no message signal information is available in the phase, the phase is ignored during detection of the message signal). The 2-D DFT of a 8×8 real matrix has 4 real, and 60 complex (out of which only 30 are unique) coefficients. Note that this causes a reduction in the number of available channels from 64 to 34, as only 34 magnitude coefficients are unique (the magnitudes of 30 complex and 4 real coefficients). In addition, this also reduces the message energy available to each channel by a factor of (approximately) half – only half the message signal energy distributed among the 60 complex coefficients is available for detection. Half the message signal energy is added just for the purpose of maintaining the symmetry properties of the DFT of a real signal. But by sacrificing some channels, (or by reducing the degrees of freedom),

we obtain smaller noise variances in each channel. As an example, consider N *iid* random variables (N degrees of freedom) with variance σ^2. If we construct $N/2$ random variables from the N original variables by averaging every two of them, the variance of the resultant $N/2$ random variables will be *iid* with variances equal to $\sigma^2/2$. Therefore, we reduce the variance of noise in the channels by reducing the degrees of freedom (from N to $N/2$).

From the plots in Figures 3.8 - 3.11, we see that capacities for all decompositions fall with increased processing noise as expected. DCT and subband decompositions are better than Hartley and Hadamard decompositions for detection of the message when processing noise is low. It is also seen that decompositions unfavorable for compression (DFT, Hartley and Hadamard) are more immune to processing noise than decompositions suitable for compression (DCT, subband).

What is surprising, is that magnitude DFT decomposition offers more capacity than better energy compaction transforms even when there is no processing noise. In this case a reduction in the entropy of the image noise is achieved by ignoring the phase of the DFT coefficients. The reduction in entropy is precisely the information content in the DFT phase. Apparently, this reduction in entropy more than offsets the reduced signal energy available for detection (again, only half the signal energy is available for detection as the added signal power is divided between 64 coefficients while only 34 of them are available for detection). Yet magnitude DFT performs better than other transforms because *DFT phase contains disproportionately more information than the DFT magnitude!*. Note that in Figures 9 and 10 the the capacity of magnitude DFT decomposition for Baboon and Bridge images is much higher than that of the high GTC transforms even for no processing noise scenario. On the other hand the capacity of magnitude DFT is comparable to or even less than high GTC transforms for smoother images like Lena and Barbara. This can be due to the following reasons:

- High GTC transforms suitable for most images are not very well suited for these high activity images.
- The disparity between information content in the phase and magnitude is even more pronounced for these high-activity images.

In addition, being a relatively low GTC transform, DFT is also robust to processing noise like Hadamard and Hartley transforms.

Another surprising result, is that we find embedding in DCT domain is slightly more resistant to Subband compression methods than JPEG. Similarly embedding in the Subband domain is slightly more resistant to JPEG than SPIHT. This may appear to contradict the idea of "matching" embedding transforms with the compression method. But one should note that the matching is useful only if we design the methods 'intelligently'. So designing a DCT based data hiding method with no idea of say, the quantization matrix used, may not be more robust to JPEG than a wavelet based data hiding method.

As an indicator of the performance of these decompositions for other possible compression methods, we look at the capacities of the decompositions when an image has to survive JPEG *or* SPIHT. We group the four different processing scenarios of JPEG and SPIHT into four pairs - (JPEG-75, SPIHT 1 bpp), (JPEG-50, SPIHT 0.75 bpp), (JPEG-35, SPIHT 0.5 bpp) and (JPEG-25, SPIHT 0.35 bpp). For example, to calculate the capacity when the message signal has to survive JPEG-50 *or* SPIHT 0.75 bpp we choose the worst processing noise in each sub-band (from the estimates of processing noise for SPIHT 0.75 bpp and JPEG-50). The capacities so obtained are plotted in Figure 3.12. Note that the estimates of the capacity still follow the same trend.

We can define a *figure of merit*, for each of the L ($\frac{L}{2}+2$ for magnitude DFT) sub-channels for the various decompositions. The figure of merit is given as the ratio of the capacity of each sub-channel to the logarithm of the power of the message signal

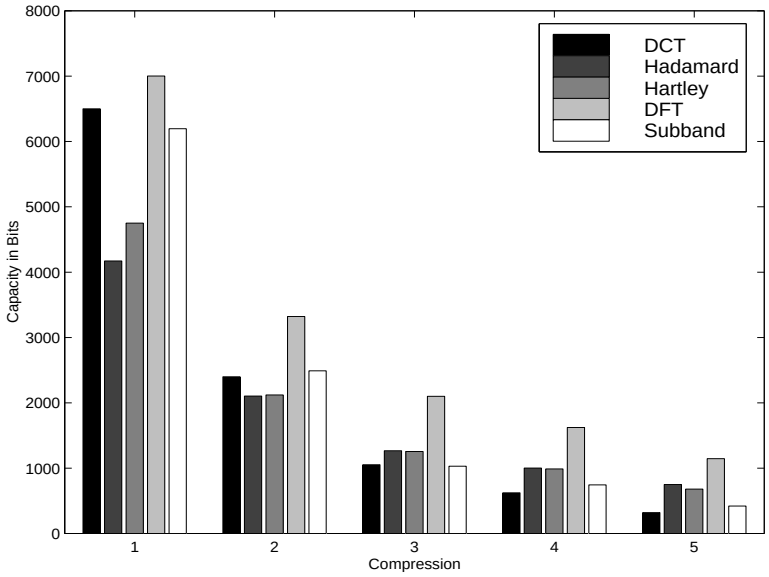

Figure 3.12 Average capacity estimates for 15 images when the message signal has to survive SPIHT *or* JPEG. The compression indices 1 - 5 correspond to 1 - lossless compression, 2 - (JPEG - 75, SPIHT 1 bpp), 3 - (JPEG - 50, SPIHT 0.75 bpp), 4 - (JPEG - 35, SPIHT 0.5 bpp), 5 - (JPEG - 25, SPIHT 0.35 bpp).

in that sub-channel. The approximate (rounded) values of the figure of merit for the channels of different decompositions (when the message has to survive SPIHT 0.5 bpp or JPEG-35), are listed in Table 3.1 for various 64-band decompositions. These figures indicate the relative performance of each sub-channel, and would therefore be useful in designing hidden communication methods to make optimal trade-offs between the visual quality of the image and the number of bits that can be embedded. As the figure of merit is normalized with respect to the message signal energy in each band, it is independent of the model used for the visual threshold. The high figures of merit for the channels of the magnitude DFT decomposition show that it would perform better than other decompositions for any message signal energy assignment method (model for visual threshold).

Figure 3.13 shows the average capacities for 15 images for 256 band decompositions. As expected, we see an increase in the estimate of the capacity. The increase is more substantial for low processing noise scenarios.

Table 3.1 Figure of merit of the bands of different decompositions when the image has to survive SPIHT 0.5 bpp. (a) magnitude DFT, (b) DCT, (c) uniform subband and (d) Hadamard.

(a)-DFT

0	27	49	69	83	0	0	0
27	53	72	70	87	0	0	0
49	72	69	38	51	0	0	0
69	70	38	18	32	0	0	0
83	87	51	32	43	0	0	0
0	69	46	33	0	0	0	0
0	71	69	46	0	0	0	0
0	54	71	69	0	0	0	0

(b)-DCT

0	8	19	29	37	42	29	23
8	17	28	34	41	28	10	28
19	28	36	40	35	15	7	22
29	34	40	40	23	8	2	22
37	41	35	23	15	2	11	2
42	28	15	8	2	0	0	0
29	10	7	2	11	0	0	6
23	28	22	22	2	0	6	14

(c)-Subband

0	9	29	37	43	41	37	33
9	18	19	26	37	43	32	18
29	19	30	37	29	23	30	16
37	26	37	28	44	43	10	8
43	37	29	44	11	19	2	7
41	43	23	43	19	39	6	9
37	32	30	10	2	6	2	12
33	18	16	8	7	9	12	11

(d)-Hadamard

0	23	11	22	5	22	10	22
23	34	30	12	38	24	34	22
11	30	31	24	22	29	28	26
22	12	24	13	28	21	27	13
5	38	22	28	11	32	17	30
22	24	29	21	32	22	33	24
10	34	28	27	17	33	24	30
22	22	26	13	30	24	30	17

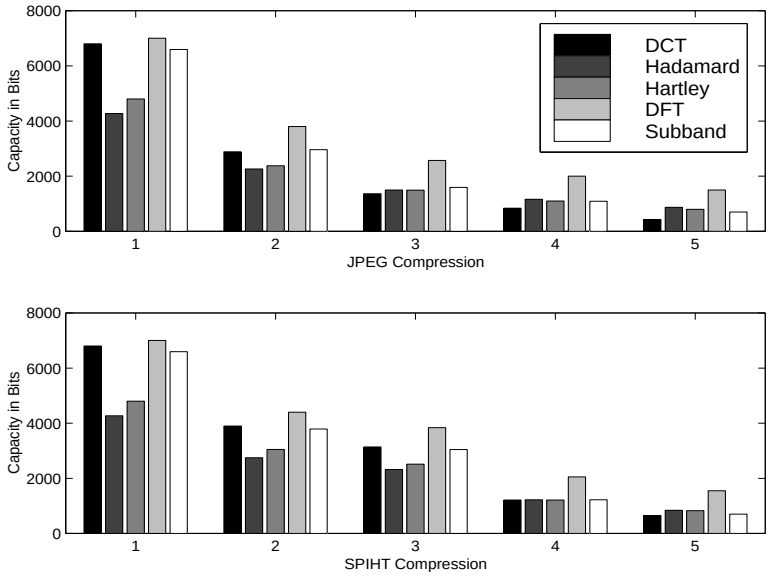

Figure 3.13 Average capacity estimates for 15 256 $\times$256 images for 256 band decomposition. The indices for JPEG compression correspond to different JPEG quality factors. (1 - lossless compression, 2 - 75%, 3 - 50%, 4 - 35%, 5 - 25%) The indices for SPIHT compression correspond to different bit rates (1 - lossless , 2 - 1 bpp, 3 - 0.75 bpp, 4 - 0.5 bpp, 5 - 0.35 bpp).

Finally, note that we evaluate processing noise by measuring the correlation between the image components before and after compression. By this, we implicitly assume that the message signal (signature) is affected to the same extent as the image coefficients themselves by the compressor / decompresser. In a practical method, this may not be true. In fact, as pointed out in Chapter 1, an *ideal* compression method would completely suppress any extra information added to the image coefficients (no data hiding would be possible with an ideal compression method). But practical compression methods can probably be tricked into believing that the embedded information is an *integral part* of the image if the embedded message signals are chosen *intelligently*. However, choosing the signature **S** *intelligently* may imply reduced degrees of freedom for its choice, translating into reduced capacity.

3.8 The Ideal Decomposition

For a moment, if we ignore the magnitude DFT decomposition, the performance of a decomposition depends roughly on its position in the GTC Scale. In Figure 3.14, a few transforms are marked in the GTC Scale. To the extreme left is the identity transform which has no energy compaction. In the extreme right is the KLT [77]. Transforms to the right would yield high capacities for low processing noise scenarios. As the processing noise increases, we should move towards the left to choose a transform. The question is, given a processing noise scenario, what would be the ideal decomposition?

Figure 3.14 The GTC Scale

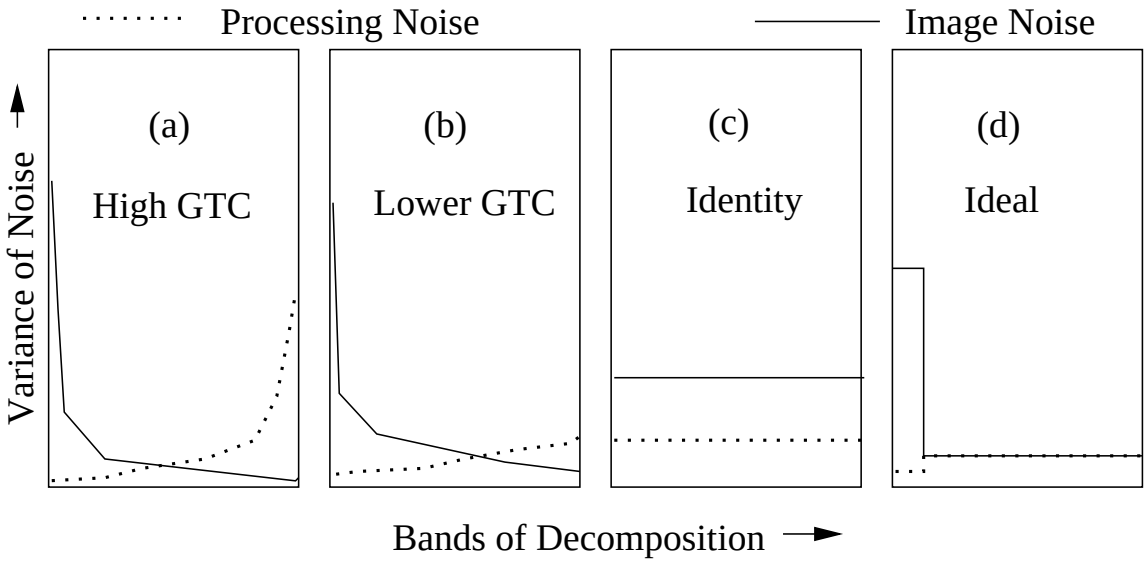

Figure 3.15 The ideal decomposition

For example, if $\alpha = 0.5$ in Eq. (3.12), the capacity of *each* sub-channel of a decomposition is given by

$$\mathbf{C}_{h_j} = \log_2(1 + \frac{K\sigma_{\mathrm{ig}_j}}{\sigma^2_{\mathrm{ig}_j} + \sigma^2_{p_j}}) \tag{3.15}$$

In order to maximize $\mathbf{C}_{h_j}$ it is enough to maximize $t = \frac{\sigma_{\mathrm{ig}_j}}{\sigma^2_{\mathrm{ig}_j}+\sigma^2_{p_j}}$. It can be easily seen, that t (and hence $\mathbf{C}_{h_j}$) is maximized when $\sigma^2_{\mathrm{ig}_j} = \sigma^2_{p_j}$. The ideal decomposition would be the one which results in image noise variances close to the processing noise variances in the maximum number of sub-bands. Typically for high GTC decompositions, (Figure 3.15 (a)) $\sigma_i >> \sigma_p$ in the low frequency bands and $\sigma_p >> \sigma_i$ in the high frequency bands. For lower GTC transforms, the discrepancy is reduced (Figure 3.15 (b)). On the other hand, for identity transform $\sigma_i >> \sigma_p$ in the single band (Figure3.15 (c)). Therefore, for the ideal decomposition, the image and processing noise variances should be distributed as shown in Figure 3.15 (d). For the ideal decomposition, the image and processing noise variances should be distributed as shown in Figure 3.15. It should also be noted, that a decomposition so obtained would perform as expected only if we are able to assume the same model for the

relationship between the coefficient variance and the visual threshold. Therefore, the search for such a decomposition may not be simple.

3.9 Factors Influencing Choice of Transform

The superiority of the magnitude DFT decomposition, among the decompositions compared, lies in an advantageous trade-off, where we reduce the degrees of freedom to reduce the entropy of the image. Simulations show that the magnitude DFT decomposition yields uniformly superior performance (over other decompositions) for both low and high processing noise scenarios.

The final choice of the decomposition should depend on the end application. While some data hiding applications, like watermarking, may need robustness to intentional tampering, some applications like captioning may not. The performance of magnitude DFT decomposition is superior to others because of its low information content. For the very same reason the magnitude of DFT coefficients can be altered significantly without affecting the visual quality of the image. This makes the DFT coefficients very vulnerable to intentional tampering. Thus, the magnitude DFT decomposition may not be suitable choice for watermarking applications. However, standard image compression methods do not seem to affect the magnitude DFT coefficients drastically. This 'hole' in standard compression methods can be put to use advantageously. So for applications where intentional tampering is not an issue, magnitude DFT may be a good choice for both low and high processing noise scenarios.

For robustness to 'commercial quality' compression methods (better than JPEG-50 or SPIHT 1 bpp), high GTC transforms like DCT and Wavelets (subband) perform better than low GTC transforms. Further, being transforms especially used for image compression applications, they would leave very little room for intentional tampering without significant degradation of the image. This property

would make them very suitable for watermarking applications. For other data hiding methods, with perhaps reduced resistance to intentional tampering but increased resistance to processing noise (lower quality compression), transforms like Hadamard or Hartley transform would probably be more useful. For example, an average video frame is likely to suffer more processing noise than an average still image. So low GTC transforms may be good choices for data hiding in video frames. Further, though lower GTC transforms are bound to have reduced resistance to intentional tampering (compared to DCT or wavelets) if the transform employed is *known*, the case is different if the transform used is not known. There exists a high degree of freedom for the choice of the low GTC embedding transforms. This enhanced degree of freedom for the choice of the embedding transform could result in very high robustness to intentional tampering. In the next section we outline a method for obtaining low GTC subband transforms, from random seeds.

3.10 Fast Transforms Generated from Random Seeds

In this section we outline 3 ways of generating low GTC subband transforms from random seeds:

- perturbation of high GTC subband filters
- random search
- generating cyclic subband filters in the DFT domain

The first and second methods generate non-cyclic subband filters of finite support, while the third method generates cyclic subband filters. The differences between subband filters with finite support and cyclic subband filters, and fast implementation of these transforms using FFT, are outlined in Appendix A.

3.10.1 Perturbation of High GTC Subband Filters

It is well known [78] that a degree k, 2-band paraunitary system $\mathbf{E}(z)$ can be obtained from $k+1$ unit norm vectors, $\mathbf{v}_1 \cdots \mathbf{v}_k$ and $\mathbf{u}$ of size 2×1, as

$$\mathbf{E}(z) = \mathbf{V}_1(z)\mathbf{V}_2(z)\cdots\mathbf{V}_k(z)\mathbf{U} \tag{3.16}$$

where

$$\mathbf{V}_i(z) = \mathbf{I} - \mathbf{v}_i\mathbf{v}_i^T + z^{-1}\mathbf{v}_i\mathbf{v}_i^T, \tag{3.17}$$

where $\mathbf{I}$ is an identity matrix of size 2×2 and

$$\mathbf{U} = \mathbf{I} - 2\mathbf{u}\mathbf{u}^T \tag{3.18}$$

In other words, for every choice of the unit norm vectors $\mathbf{v}_1 \cdots \mathbf{v}_k$ and $\mathbf{u}$, there exists a unique paraunitary system.

To generate paraunitary systems from random seeds, we could start with the unit norm vectors $\mathbf{v}_1 \cdots \mathbf{v}_k$ and $\mathbf{u}$ corresponding to some high GTC known filter (say 20 tap Daubechies filter) and perturb those vectors randomly to obtain their corresponding lower GTC filters.

3.10.2 Random Search

In this method, the key from which the filters are generated has two parts. The first part of the key is used as a seed to generate a random sequence of seeds. Each seed in turn is used to generate the unit norm vectors randomly. From the generated vectors the characteristics of the corresponding filter is obtained. The search is stopped when a 'satisfactory' filter is obtained. The second part of the key now becomes the index number of the random seed that generates a satisfactory filter. This method however, may not be acceptable for watermarking applications (we shall see in Chapter 7 that watermarking protocols should have very limited degree of freedom for choosing the signature or the decomposition).

3.10.3 Cyclic Subband Filters in the DFT Domain

The characteristics of cyclic subband filters [79] is outlined in Appendix A. If $\mathbf{h} \leftrightarrow \mathbf{H}$, and $\mathbf{h} \in \Re^N$, then $\mathbf{h}$ satisfies the conditions for a 2 - band cyclic subband filter if

$$|H(l)|^2 + |H(l + \frac{N}{2})|^2 = 2 \text{ for } l = 0, \cdots, \frac{N}{2} - 1. \tag{3.19}$$

A relatively low GTC cyclic subband filter can be generated in the DFT domain by fixing the magnitude response $|H(l)|$ for $l = 0, \ldots, \frac{N}{2} - 1$ and choosing the phase $\angle H(l)$ for $l = 0, \ldots, \frac{N}{2} - 1$ randomly.

CHAPTER 4

OPTIMAL SIGNALING FOR MULTIMEDIA STEGANOGRAPHY

Conventional communication methods employ a wide variety of signaling techniques which essentially map a bit sequence to a real valued sequence. The real valued sequence is in turn transmitted over a channel. However, communication techniques for the purpose of multimedia steganography or data hiding have to transmit the real valued sequence corresponding to the signal constellation *superimposed* on the original content (without affecting the fidelity of the original content noticeably). In Chapter 3 we explored the possibility of super-positioning the signature sequence onto the content. However, there exists other options for embedding the signature in the content.

In this chapter, we explore practical solutions for signaling methods for multimedia steganography. Data hiding is seen as a sophisticated signaling technique using a *floating* signal constellation. We propose such a signaling method and present both theoretical and simulated evaluations of its performance in an additive noise scenario. The problem of optimal choice of the parameters of the proposed technique is also explored, and solutions are presented.

4.1 Problem Statement

The process of data hiding in images consists of an embedder E, and a detector D. If $\mathbf{I}$ is the original or *cover* image, and $\mathbf{b}$ is a sequence of bits to be embedded in the image, the *stego* image $\hat{\mathbf{I}}$ (the image with the embedded data) is obtained as

$$\hat{\mathbf{I}} = E(\mathbf{I}, \mathbf{b}, \mathcal{K}) \tag{4.1}$$

where $\mathcal{K}$ is a *key*. We expect the image $\hat{\mathbf{I}}$ to undergo some modification (like lossy compression) before it reaches the receiver (detector D), where the hidden bit sequence is extracted. Let $\tilde{\mathbf{I}} = \hat{\mathbf{I}} + \mathbf{N}$ be the received image.

Depending on whether the method is escrow or oblivious, the detector takes the form

$$\tilde{\mathbf{b}} = \begin{cases} D(\tilde{I}, K, I) & \text{escrow} \\ D(\tilde{I}, K) & \text{oblivious} \end{cases} \tag{4.2}$$

In most data hiding methods, the bit sequence to be embedded, *viz.* $\mathbf{b}$, is converted to a form *suitable for embedding* in the cover image. Let $\mathbf{s} = \mathcal{S}(\mathbf{b})$. In other words, the signaling method for the steganographic communication, *viz.* $\mathcal{S}$, converts the bit sequence $\mathbf{b}$ to a *signature sequence* $\mathbf{s}$. Most often, the signature sequence $\mathbf{s}$ is embedded in some transform domain. Let $\mathcal{T}$ represent a unitary transformation employed, and $\mathbf{C} = \mathcal{T}(\mathbf{I})$. For an $M \times N$ image $\mathbf{I}$, $\mathbf{C}$ is $M \times N$ dimensional. The overall embedding and detection operations now take the following form:

$$\begin{array}{lll} \mathbf{C} = \mathcal{T}(\mathbf{I}) & \mathbf{s} = \mathcal{S}(\mathbf{b}) & \hat{\mathbf{C}} = \mathcal{E}(\mathbf{C}, \mathbf{s}) \\ \hat{\mathbf{I}} = \mathcal{T}^{-1}(\hat{\mathbf{C}}) & & \tilde{\mathbf{I}} = \hat{\mathbf{I}} + \mathbf{N} \\ \tilde{\mathbf{C}} = \mathcal{T}(\tilde{\mathbf{I}}) & \tilde{\mathbf{s}} = \mathcal{D}(\tilde{\mathbf{C}}) & \tilde{\mathbf{b}} = \mathcal{S}^{-1}(\tilde{\mathbf{s}}) \end{array} \tag{4.3}$$

From a signal processing perspective, data hiding methods can be classified into two categories, depending on the type of embedding and detecting operators. In the first category [9, 12] lies methods where the $\mathcal{E}$ adds the signature sequence *linearly* to $\mathbf{C}$, as in Chapter 3, and $\mathcal{D}$ detects $\tilde{\mathbf{s}}$ from $\tilde{\mathbf{C}}$ by *correlative processing*. For linear methods, if the original image is not available at the receiver, (or if $\mathbf{C}$ is not known), then the original image itself (or its transform coefficients $\mathbf{C}$) is noise, for the purpose of detection of the hidden bit sequence $\mathbf{b}$. Alternately, linear data hiding methods employ "conventional" signaling techniques for data hiding. In the second category $\mathcal{E}$ and $\mathcal{D}$ are *non-linear*. One of the important characteristics of the non-linear methods is their ability to suppress the noise due to the original image (or self-noise), even though the original image is not available at the receiver.

For linear data hiding methods (or Type I methods), the *purpose* of the decomposition is to obtain a *favorable distribution* of the image and processing noise in the different bands. However we shall see that even with *ideal redistribution* of the two noise sources, linear or Type I data hiding can never be optimal.

4.2 Non linear Data Hiding

The non-linear methods are capable of utilizing the robust low frequency bands even though the original image is not available at the detector. In one of our prior arts [15] the signature is introduced in 8 low frequency DCT coefficients (of each 8×8 block). The vector $\mathbf{x}$ of the low-frequency DCT coefficients is scrambled by means of an (invertible) cyclic all-pass filter $\mathcal{F}$ with pseudo random coefficients. Let $\mathbf{y} = \mathcal{F}(\mathbf{x})$. The signature is added and detected in the scrambled 'domain' $\mathbf{y}$. To embed the bit we modify the signs of many small amplitude coefficients of $\mathbf{y}$ so that the resulting sequence has more positive than negative coefficients. Coefficients with large amplitudes in the scrambled domain $\mathbf{y}$ are untouched. Altering (by flipping signs) only the small magnitude coefficients guarantees that the distortion introduced is tolerable. The modified sequence $\hat{\mathbf{y}}$ is unscrambled to obtain the modified (DCT) coefficients $\hat{\mathbf{x}} = \mathcal{F}^{-1}(\hat{\mathbf{y}})$. For detecting the buried bit, the received vector $\tilde{\mathbf{x}}$ is scrambled by the filter $\mathcal{F}$ to obtain $\tilde{\mathbf{y}}$. The excess number of positive coefficients is counted. Note that by treating both high and low magnitude coefficients of $\tilde{\mathbf{y}}$ with equal weight (only the sign of the coefficient is considered), suppression of image noise is achieved. Unlike linear detection methods using correlative processing (which would attach more significance to the high amplitude coefficients), in this case, large magnitude coefficients affect the result of the detection process in the same way as the small magnitude coefficients.

In the data hiding scheme by Wang *et. al.* [59], the significant wavelet coefficients are altered. The coefficients are modified so that they *quantize* to an even or odd value depending on the bit to be embedded. In [54] Wu *et. al.* introduce a similar scheme based on JPEG quantizers. The signature is introduced in the DCT domain. Chen *et. al* [80] provide a more formal treatment of data hiding techniques, that use the quantization index to embed bits (methods which force the quantized *indices* to take a desired value depending on the information signal to be embedded).

In fact the earliest data hiding methods [38, 39], which modified only 1 or 2 LSBs of images were also non-linear. For example, a method which modifies only 2 LSBs may be considered as a form of quantization index modulation where the step size of quantizer used is 4. In recent data hiding literature, the data hiding methods [54, 59, 80] employing quantization are referred to as Type II methods. In the next section we provide a generalization of Type II methods. The generalization is based on the observation that quantization achieves self-noise suppression because of its *periodic* nature. This implies that other periodic functions are also (probably better) candidates for this purpose.

4.3 Data Hiding as a Signaling Technique

Consider a (metric) space $\mathcal{I}$ of vectors $\mathbf{C}$ (each point in the metric space may correspond to the transform coefficients of some image). Let $\mathbf{C}$ represent the transform coefficients corresponding to the original (cover) image. To embed a bit sequence $\mathbf{b}$ of length n_b, we should be able to define a constellation with a minimum of 2^{n_b} points in $\mathcal{I}$. The problem now is the choice of a signaling set or a signal constellation, such that *any* point in $\mathcal{I}$ can be relocated to a point in the constellation corresponding to the arbitrary bit sequence to be hidden, *without perceptual distortion*. The new point to which the image (or $\mathbf{C}$) is moved is then the stego image, (or its transform coefficients $\hat{\mathbf{C}}$). If the space $\mathcal{I}$ is *tiled* by the constellation, reasonably low amounts of distortion can be achieved. On the other hand, we also need the hidden bits to survive some distortion that the stego image is expected to undergo before it reaches the detector. Therefore we need the points of the constellation to be "well separated".

4.3.1 Signaling for Data Hiding

Given a sequence of bits $\mathbf{b}$ of length K, and coefficients $\mathbf{C} \in \Re^{MN}$ (transform coefficients of $M \times N$ images), where typically $K << M \times N$, we need to *map* the bit sequence to a new "state" $\hat{\mathbf{C}}$. Let $\nu = \mathcal{T}(\mathbf{N})$, be the effect of the additive noise $\mathbf{N}$ in the channel on the transform coefficients $\hat{\mathbf{C}}$. Or, $\tilde{\mathbf{C}} = \hat{\mathbf{C}} + \nu$. However, we would like to minimize the channel noise ν. We know that most of the noise would be concentrated in the high frequency components of the image (a compression method like JPEG quantizes the high frequency coefficients very coarsely). Therefore a significant portion of the noise can be *eliminated* if the data is embedded in the transform domain, and high frequency coefficients are *ignored* (not used for data hiding). We could use a subset (low-to-medium frequencies) $\mathbf{c} \in \Re^D$ of the coefficients $\mathbf{C} \in \Re^{MN}$ for data hiding.

We can now consider any image as a point in D dimensional metric space (of D-dimensional vectors $\mathbf{c}$). Therefore, the over-all embedding and detecting sequences now take the form

$$\begin{array}{ll} \mathbf{s} = \mathcal{S}(\mathbf{b}) & \hat{\mathbf{c}} = \mathcal{E}(\mathbf{s}, \mathbf{c}) \quad \text{Embedding} \\ \tilde{\mathbf{s}} = \mathcal{D}(\tilde{\mathbf{c}}) & \tilde{\mathbf{b}} = \mathcal{S}^{-1}(\tilde{\mathbf{s}}) \quad \text{Detection} \end{array} . \tag{4.4}$$

The over all signaling method has now been split into two parts - a part ($\mathcal{E}$ and $\mathcal{D}$) which depends on $\mathbf{c}$, and the part $\mathcal{S}$ and $\mathcal{S}^{-1}$ which are independent of $\mathbf{c}$. Moreover, $\mathbf{s}$ represents a point in a signal constellation with *known origin*. We shall see that $\mathcal{E}$ and $\mathcal{D}$ can be implemented as simple *periodic functions*, and of course, a wealth of knowledge exists for the choice of the conventional signaling part $\mathcal{S}$.

4.3.2 Self-Noise Suppression

Figure 4.1 is an illustration of the function of $\mathcal{E}$ and $\mathcal{D}$. In the figure, for purposes of illustration we have $D = 2$ (typically, for images D may be of the order of tens of thousands). A bit sequence $\mathbf{b}$ is mapped by $\mathcal{S}$ to a point $\mathbf{s}$ in the *bold rectangular region near the origin*. The filled box represents the position of $\mathbf{s}$ in D-dimensional

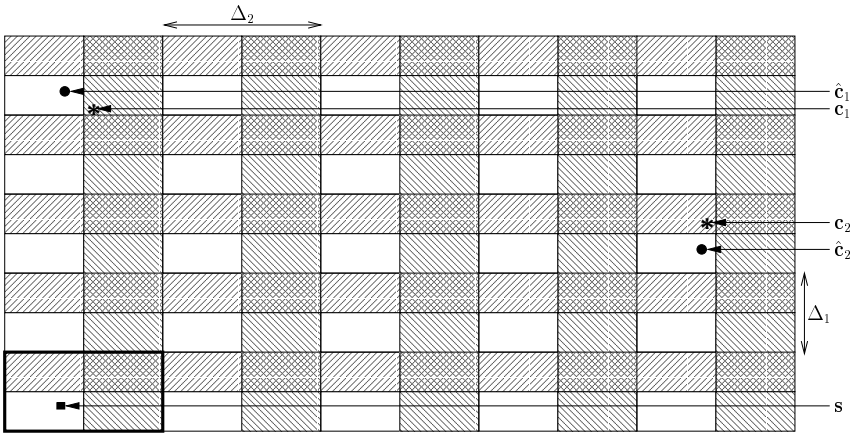

Figure 4.1 The SNS operators $\mathcal{E}$ and $\mathcal{D}$

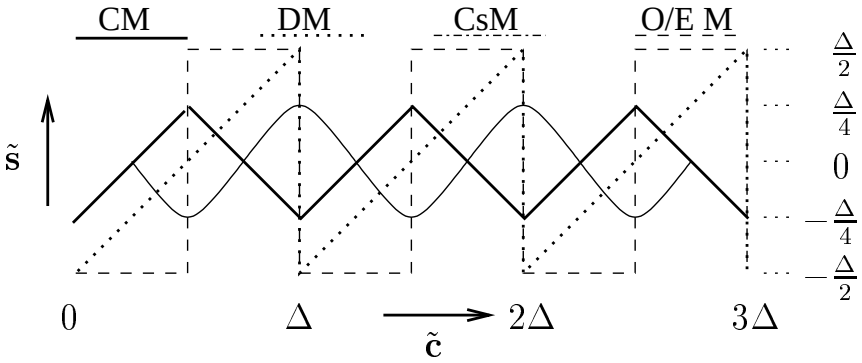

Figure 4.2 Periodic functions for SNS

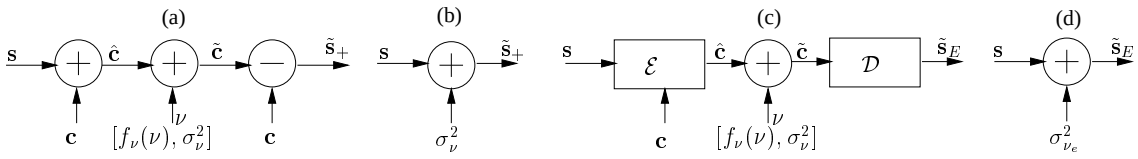

Figure 4.3 (a) Linear cover image escrow data hiding. (b) Equivalent additive noise channel. (c) Non-linear oblivious detection data hiding. (d) Equivalent additive noise channel.

space. The filled circles represent the position of $\mathbf{c}_1$ and $\mathbf{c}_2$ (transform coefficients of 2 images). $\mathcal{E}$ maps $\mathbf{c}_1$ to the point $\hat{\mathbf{c}}_1$ and $\mathbf{c}_2$ to $\hat{\mathbf{c}}_2$. $\mathcal{D}$, on the other hand, would map both $\hat{\mathbf{c}}_1$ and $\hat{\mathbf{c}}_2$ to $\mathbf{s}$. We call the pair ($\mathcal{E}$, $\mathcal{D}$) as the *self-noise suppression* (SNS) method. As explained earlier, for *linear oblivious* data hiding techniques, for the purpose of detection of the hidden bits in an image, the image itself is noise. The SNS operators "suppress" the original image component in $\tilde{\mathbf{c}}$ and extract the component $\tilde{\mathbf{s}}$ which is needed for obtaining $\mathbf{b}$. The SNS method, which obtains the origin of the signal constellation, is characterized by step sizes $\Delta_i, i = 1 \cdots D$ corresponding to each of the D dimensions. The SNS method uses *periodic functions* in each of the D dimensions to translate the point $\mathbf{s}$ in the constellation with known reference (the origin), to points like $\hat{\mathbf{c}}_1$ or $\hat{\mathbf{c}}_2$ depending on the position of the original coefficients ($\mathbf{c}_1$ or $\mathbf{c}_2$) such that the distortion introduced ($d(\mathbf{c}_1, \hat{\mathbf{c}}_1)$ or $d(\mathbf{c}_2, \hat{\mathbf{c}}_2)$) is minimal.

4.3.3 Correlation and Equivalent Noise

Before we explore specific SNS techniques, consider the linear *cover image escrow* data hiding method of Figure 4.3 (a). Let $\nu \sim [f_\nu(\nu), \sigma_\nu^2]$ be additive noise in the channel.

$$\begin{array}{ll} \hat{\mathbf{c}} = \mathbf{c} + \mathbf{s} & \tilde{\mathbf{c}} = \hat{\mathbf{c}} + \nu \\ \tilde{\mathbf{s}}_+ = \tilde{\mathbf{c}} - \mathbf{c} & \tilde{\mathbf{s}}_+ = \mathbf{s} + \nu \end{array} \tag{4.5}$$

Let the signature be a binary sequence ($s(k) = \pm t_k, k = 1 \cdots D$). For simplicity we further assume that $t_k = t \,\forall\, k$. This is equivalent to the scenario in Figure 4.3 (b), of transmitting $\mathbf{s}$ over a channel with additive noise variance σ_ν^2.

In such a scenario, the normalized inner product of the $\mathbf{s}$ and $\tilde{\mathbf{s}}_+$ (for sufficiently large D) can be written as

$$\rho_+ = \frac{\mathbf{s}^T \tilde{\mathbf{s}}_+}{|\mathbf{s}||\tilde{\mathbf{s}}_+|} = \frac{\int_{-\infty}^{\infty} t(t+\nu) f_\nu(\nu) \mathrm{d}\nu}{\sqrt{\int_{-\infty}^{\infty} t^2 (t+\nu)^2 f_\nu(\nu) \mathrm{d}\nu}}. \tag{4.6}$$

If the pdf $f_\nu(\nu)$ is *even*, then it can be easily seen that

$$\rho_+^2 = \frac{t^2}{t^2+\sigma_\nu^2} \quad \text{or} \quad \sigma_\nu^2 = \frac{t^2(1-\rho_+^2)}{\rho_+^2} \tag{4.7}$$

Now consider the Type II data hiding scenario in Figure 4.3 (c). To differentiate between the the recovered signature sequences $\tilde{\mathbf{s}}$ in Type I and Type II methods, we use different subscripts - $+$ and E. Let ρ_E be the normalized inner-product of $\mathbf{s}$ and $\tilde{\mathbf{s}}_E$. We could represent Figure 4.3 (c) by Figure 4.3 (d) where, similar to Eq. (4.7),

$$\rho_E = \frac{\mathbf{s}^T \tilde{\mathbf{s}}_E}{|\mathbf{s}||\tilde{\mathbf{s}}_E|} \quad \sigma_{\nu_e}^2 = \frac{t^2(1-\rho_E^2)}{\rho_E^2}. \tag{4.8}$$

Even though the additive noise in the channel is the same as the previous (linear escrow technique of (a)) case, typically, $\sigma_{\nu_e}^2 > \sigma_\nu^2$ (or $\rho_E < \rho_+$). We may consider $\sigma_{\nu_e}^2$ as the variance of the *equivalent* additive noise. The difference $\sigma_{\nu_e}^2 - \sigma_\nu^2$ may then be considered as the *penalty* paid for having to "guess" the origin of the signal constellation. We shall see later that for the proposed SNS technique, analytical evaluation of ρ_E is possible (similar to Eq. (4.6)). From the value of ρ_E, the equivalent additive noise variance ($\sigma_{\nu_e}^2$) can be evaluated.

4.3.4 Periodic Functions for SNS

As mentioned in the previous section, what we need is a periodic function for tiling the space of $\mathbf{c}$ with a constellation defined by a conventional signaling scheme (with known origin). Classic Type II methods, which embed a zero or one by forcing

the quantization index to be odd or even in effect, use a periodic function of square waves (O/E M) in Figure 4.2. The figure also shows other possible periodic functions. Dither Modulation (DM), proposed by Chen *et. al* in [80] may can be considered as using the saw-tooth periodic function in Figure 4.2. In Ref. [81], we introduced a *continuous* periodic function (CM) for self-noise suppression. Another possibility is a Sine / Cosine periodic function (CsM).

4.3.4.1 Dither Modulation

In this method

$$\begin{aligned} \hat{\mathbf{c}} = \mathcal{E}(\mathbf{c}, \mathbf{s}) &= Q(\mathbf{c} + \mathbf{s}) - \mathbf{s} \\ \tilde{\mathbf{s}} = \mathcal{D}(\tilde{\mathbf{c}}) &= Q(\tilde{\mathbf{c}}) - \tilde{\mathbf{c}} \end{aligned} \tag{4.9}$$

where, Q represents a uniform quantizer with step size Δ.

Figure 4.4 illustrates the simulated performance of this SNS technique for uniformly distributed and binary sequences $s(k)$. The simulations were obtained for Gaussian sequences $\mathbf{c}$ ($\sigma_c = 200$) of length 4096 for $\Delta = 30$. The normalized correlation ρ was obtained by averaging over many realizations of additive Gaussian noise ν.

Note that embedding any signature sequence $\mathbf{s}$ (even a sequence of zeroes!) results in a mean square distortion of $\frac{\Delta^2}{12}$. The SNR in the x-axis therefore represents the ratio of the power of the distortion introduced to embed the signature, viz, $\frac{\Delta^2}{12}$, to the variance of the additive noise σ_ν^2 - SNR $= 10 \log_{10} \frac{\Delta^2}{12\sigma_\nu^2}$. It is clear from Figure 4.4 that the best performance is obtained for binary $\pm\frac{\Delta}{4}$ sequences. This is due to the fact that as long as $-\Delta/4 \leq s(k) \leq \Delta/4$, corresponding points in neighboring quantization cells are *maximally separated*.

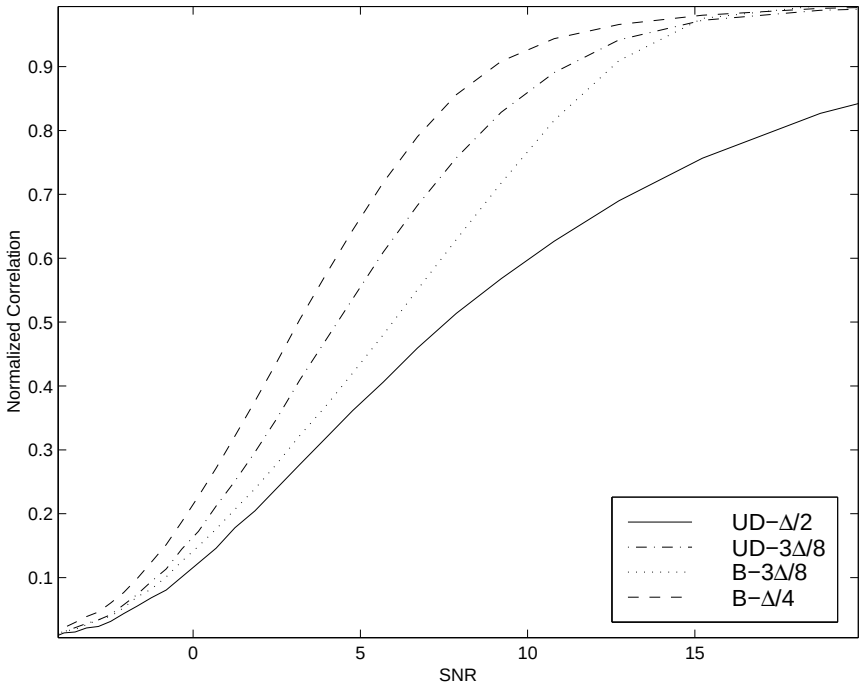

Figure 4.4 Performance of dither modulation for uniformly distributed and binary signature sequences

4.3.4.2 Continuous Periodic SNS The algorithm for $\mathcal{D}(\tilde{\mathbf{c}})$ of the CM-SNS is as follows:

$$\begin{aligned} q(k) &= \text{rem}(\frac{|\tilde{c}(k)|}{\Delta}), \quad k = 1 \cdots D \\ \tilde{s}(k) &= (q(k) \geq \frac{\Delta}{2}) \quad ? \quad (\frac{3\Delta}{4} - q(k)) \; : \; (q(k) - \frac{\Delta}{4}) \end{aligned}$$

In the above equation $x = (\text{Condition}) \quad ? \quad x_1 \quad : \quad x_2$ stands for "If Condition is true $x = x_1$, else, $x = x_2$", in the spirit of the C language. The operation rem(.) stands for "reminder".

Let $\mathbf{p} = \mathcal{D}(\mathbf{c})$. To introduce the signature $\mathbf{s}$, we need to modify $\mathbf{c}$ to obtain $\hat{\mathbf{c}}$ such that $\mathbf{s} = \mathcal{D}(\hat{\mathbf{c}})$. To achieve this, the distortion $e(k)$ introduced in coefficient $c(k), k = 1 \cdots D$ is equal to $|e(k)| = |\hat{c}(k) - c(k)| = |s(k) - p(k)|$. The algorithm for embedding the sequence $\mathbf{s}$ in $\mathbf{c}$ is as follows

$$e(k) = s(k) - p(k)$$

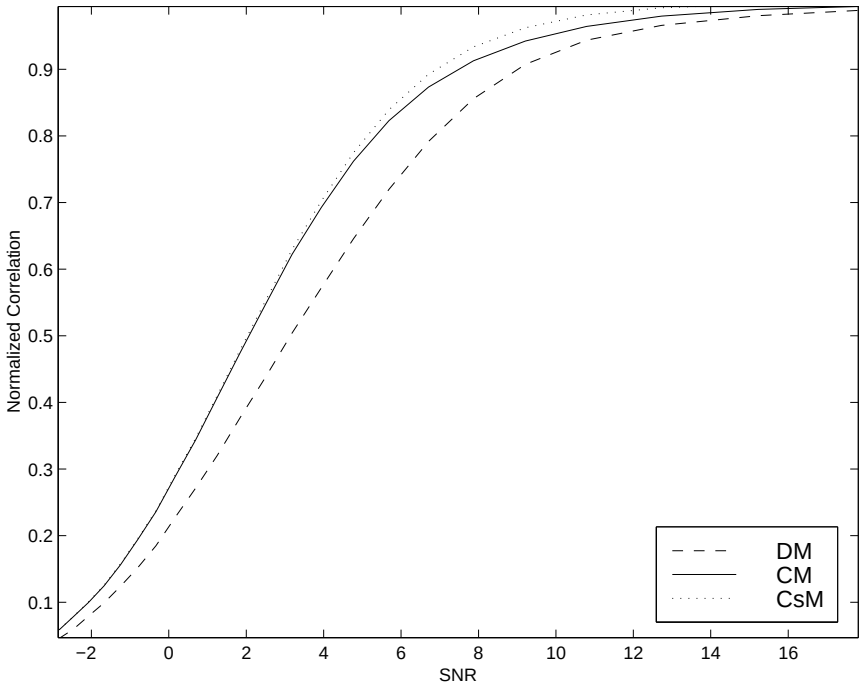

Figure 4.5 Comparison of DM (QIM), CM-SNS and cosine modulated SNS techniques

$$\begin{aligned} e(k) &= (\text{rem}\left(\frac{c(k)}{\Delta}\right) > \frac{\Delta}{2}) \quad ? \quad -e(k) \quad : \quad e(k) \\ \hat{c}(k) &= (c(k) \geq 0) \quad ? \quad c(k) + e(k) \quad : \quad c(k) - e(k) \end{aligned}$$

Figure 4.5 compares the performance of the CM-SNS technique with that of the dither modulation (DM) technique for $s(k) = \pm\frac{\Delta}{4}$. The better performance of the proposed technique (CM) is not surprising, considering the periodic function used by CM is continuous, as opposed to the DM method. For instance, for the DM method (employing signature sequences $\pm\frac{\Delta}{4}$) noise greater than $\frac{\Delta}{4}$ can change an originally $\frac{\Delta}{4}$ signal to $-\frac{\Delta}{4}$, due to the discontinuity. Figure 4.5 also illustrates the performance of another continuous periodic function - a cosine function (CsM) which performs even better than CM (especially for high SNRs). For the cosine periodic SNS technique, the detector can be represented as

$$\mathbf{s} = \mathcal{D}(\hat{\mathbf{c}}) = \frac{\Delta}{4}\cos(\frac{\hat{\mathbf{c}}}{2\pi\Delta}). \tag{4.10}$$

However, due to reasons of analytical tractability, we restrict ourselves to CM-SNS. Additionally, note that at low SNRs, the difference between CsM and CM-SNS is negligible. Typically, data hiding applications operate at low SNR levels (the ratio of permitted distortion to additive noise in the channel).

4.3.5 Analysis of CM-SNS

We shall now analytically evaluate the equivalent noise for the CM-SNS scheme, when the additive noise in the channel is ν (Figure 4.3 (c)). Let $\nu \sim [f_\nu(\nu), \sigma_\nu^2]$, and $s(k) \pm \frac{\Delta}{4}$. The expected value of the normalized correlation between $\mathbf{s}$ and $\tilde{\mathbf{s}}$, similar to Eq. (4.6), can be obtained as

$$\rho_n = \frac{2\sum_{i=0}^{\infty}\int_{\frac{i\Delta}{2}}^{\frac{(i+1)\Delta}{2}}(-1)^i(\frac{(2i+1)\Delta}{4}-\nu)f_\nu(\nu)\mathrm{d}\nu}{\sqrt{2\sum_{i=0}^{\infty}\int_{\frac{i\Delta}{2}}^{\frac{(i+1)\Delta}{2}}(\frac{(2i+1)\Delta}{4}-\nu)^2 f_\nu(\nu)\mathrm{d}\nu}} \tag{4.11}$$

The main difference between Eqs. (4.6) and (4.11) is that in the latter, the integrals are split into segments of length $\frac{\Delta}{2}$ to account for the periodicity. For Gaussian $f_\nu(\nu)$, each term (both of the numerator and denominator) of the above integral can be solved and expressed in terms of the Gaussian error function, $\mathrm{erf}(t) = \frac{2}{\pi}\int_0^t \mathrm{e}^{\frac{-y^2}{2}}\mathrm{d}y$ The variance of the equivalent additive noise $\sigma_{\nu_e}^2$ can then be obtained as

$$\sigma_{\nu_e}^2 = \frac{\frac{\Delta^2}{12}(1-\rho_n^2)}{\rho_n^2} \tag{4.12}$$

Note that even though the signature ($\pm\frac{\Delta}{4}$ binary sequence) energy is $\frac{\Delta^2}{16}$ in Eq. (4.12) we use the *energy of the distortion introduced for embedding the signature* ,viz. $\frac{\Delta^2}{12}$, instead. Its bears repeating, that the "signal" for data hiding is the *distortion introduced in the content.* In the rest of this chapter, the term SNR represents the ratio of the energy of the "signal" (which is the distortion introduced), to the energy of noise in the channel.

Figure 4.6 is a plot of the normalized correlation *vs* the standard deviation of additive Gaussian noise for various values of the quantizer step size Δ, obtained from

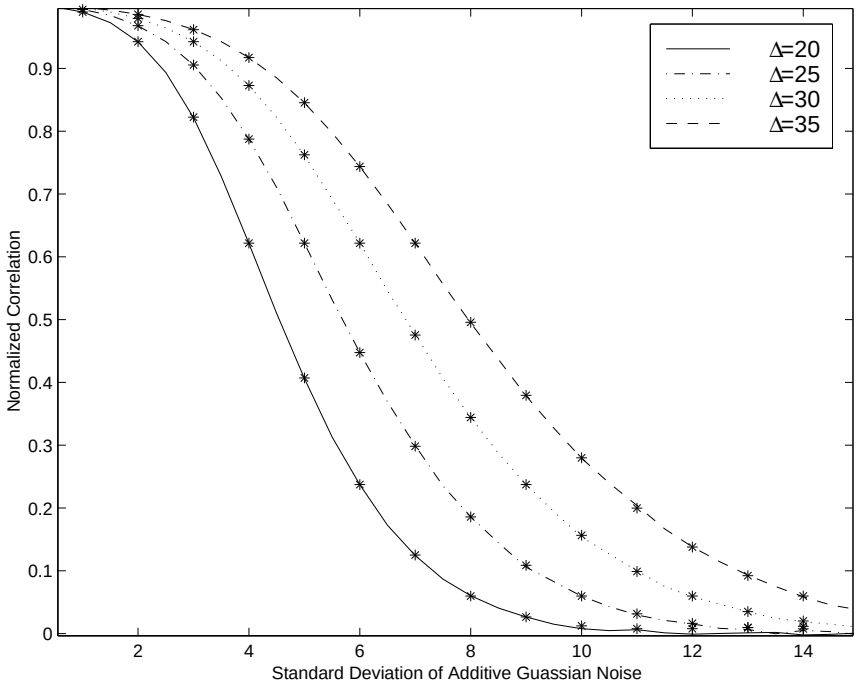

Figure 4.6 Effect of additive Gaussian noise. The lines represent values obtained from simulations. The $*$s represent the values calculated from Eq. (4.11).

simulations. The stars $(*)$ represent the corresponding values calculated by solving Eq. (4.11). The excellent agreement between simulation and the values obtained from analysis confirm the validity of Eq. (4.11). Figure 4.7 is a plot of $\sigma_{\nu_e}^2$ *vs* σ_ν^2 for various values of Δ. Note that the equivalent noise variance $\sigma_{\nu_e}^2$ can be considerably greater than σ_ν^2, the variance of the additive noise in the channel.

As mentioned earlier, the choice of Δ dictates the distortion introduced by the embedding function $\mathcal{D}$. The distortion introduced for embedding a $\pm\frac{\Delta}{4}$ sequence, is uniformly distributed between $\pm\frac{\Delta}{2}$. Therefore, as mentioned earlier, the a variance of the distortion introduced is $\frac{\Delta^2}{12}$. If the permitted distortion has a variance γ^2, then we need to choose $\Delta = \sqrt{(12\gamma^2)}$ This implies that Δ is chosen *without any consideration* of the expected noise variance σ_ν^2! Obviously, this can not be an optimal solution. This problem can be overcome by introducing *thresholding* in the SNS method.

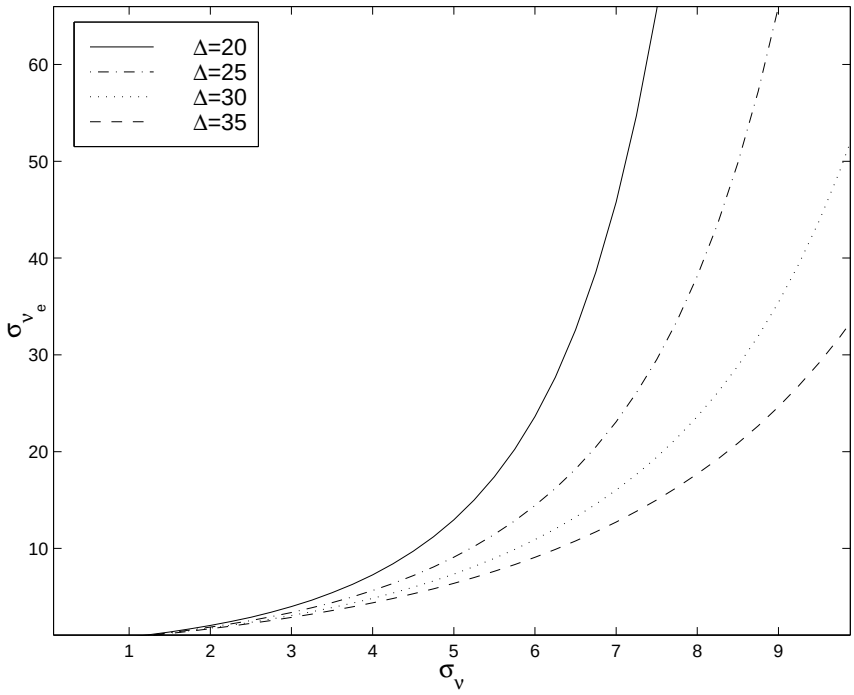

Figure 4.7 Plot of $\sigma^2_{\nu_e}$ *vs* σ_ν for $\Delta = 20, 25, 30,$ and 35

4.4 CM-SNS with Thresholding

Let γ^2 be the variance of permitted distortion due to data embedding. Let $\Delta_0^2 = 12\gamma^2$. The question we are faced with now is that given γ and some additive noise σ_ν^2, what is the optimal choice of Δ for the SNS method?

We define a modified embedding function $\mathcal{E}_t$ with the same detecting function $\mathcal{D}$. Let $\mathbf{p} = \mathcal{D}(\mathbf{c})$. In the modified embedding method, the distortion $|e(k)|$ introduced in coefficient $c(k)$, is *hard limited* to $-\frac{\beta}{2} < e(k) < \frac{\beta}{2}$, where $\beta < \Delta_0 < \Delta$. The algorithm for embedding the sequence $\mathbf{s}$ in $\mathbf{c}$ is therefore

$$
\begin{aligned}
e(k) &= s(k) - p(k) \\
e(k) &= (e(k) > \frac{\beta}{2}) \quad ? \quad \text{sign}(e(k))\frac{\beta}{2} \quad : \quad e(k) \\
e(k) &= (\text{rem}\left(\frac{c(k)}{\Delta}\right) > \frac{\Delta}{2}) \quad ? \quad -e(k) \quad : \quad e(k) \\
\hat{c}(k) &= (c(k) \geq 0) \quad ? \quad c(k) + e(k) \quad : \quad c(k) - e(k)
\end{aligned}
$$

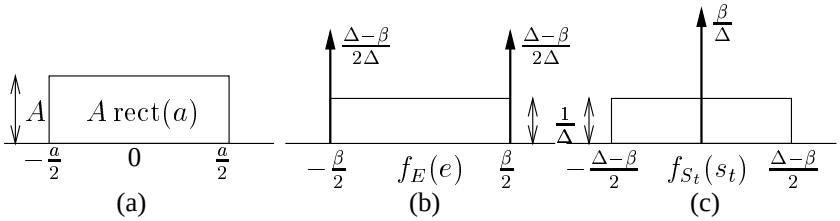

Figure 4.8 (a) The rectangular function. (b) and (c) Probability distributions of $f_E(e)$ - distortion introduced by the modified embedding function, and $f_{S_t}(s_t)$ - noise introduced due to modified embedding function.

The distortion $\mathbf{e}$ introduced by the modified embedding function $\mathcal{E}_t$ has a probability distribution and variance given by

$$\begin{aligned} f_E(e) &= \frac{1}{\Delta}\text{rect}(\beta) + \frac{\Delta-\beta}{2\Delta}\left(\delta(e-\frac{\beta}{2}) + \delta(e+\frac{\beta}{2})\right) \\ \sigma_e^2 &= \frac{\beta^2}{12\Delta}(3\Delta - 2\beta) \end{aligned} \tag{4.13}$$

Therefore, we can choose $\Delta > \Delta_0$, and $\beta < \Delta_0$, such that the distortion introduced is equal to $\gamma^2 = \Delta_0^2/12$ if

$$\gamma^2 = \Delta_0^2/12 = \frac{\beta^2}{12\Delta}(3\Delta - 2\beta) \tag{4.14}$$

Note that, with the modified embedding function, if $\hat{\mathbf{c}} = \mathcal{E}_t(\mathbf{c}, \mathbf{s})$, then $\mathcal{D}(\hat{\mathbf{c}}) \neq \mathbf{s}$. The difference $\mathbf{s}_t = \mathbf{s} - \mathcal{D}(\hat{\mathbf{c}})$ has a probability distribution and variance given by

$$\begin{aligned} f_{S_t}(s_t) &= \frac{\beta}{\Delta}\delta(s_t) + \frac{1}{\Delta}\text{rect}(\Delta - \beta) \\ \sigma_{s_t}^2 &= \frac{(\Delta-\beta)^3}{12\Delta} \end{aligned} \tag{4.15}$$

Alternately, we could assume that a distortion of variance $\Delta^2/12$ (corresponding to $\mathbf{s}$) was introduced in $\mathbf{c}$ by the embedding scheme, *along* with a noise of variance $\sigma_{s_t}^2$, given by Eq. (4.15).

Once again, the *equivalent additive noise* due to thresholding can be obtained by a measure of correlation. Let

$$\rho_t = \frac{\frac{\beta}{\Delta}\frac{\Delta}{4} + \frac{2}{\Delta}\int_0^{\frac{\Delta-\beta}{2}}(\frac{\Delta}{4} - x)\text{dx}}{\sqrt{\frac{\beta}{\Delta}\frac{\Delta^2}{16} + \frac{2}{\Delta}\int_0^{\frac{\Delta-\beta}{2}}(\frac{\Delta}{4} - x)^2\text{dx}}} = \frac{\sqrt{3}\beta(2\Delta - \beta)}{\Delta\sqrt{6\beta^3 - \frac{4\beta^3}{\Delta} + \Delta^3}}.$$

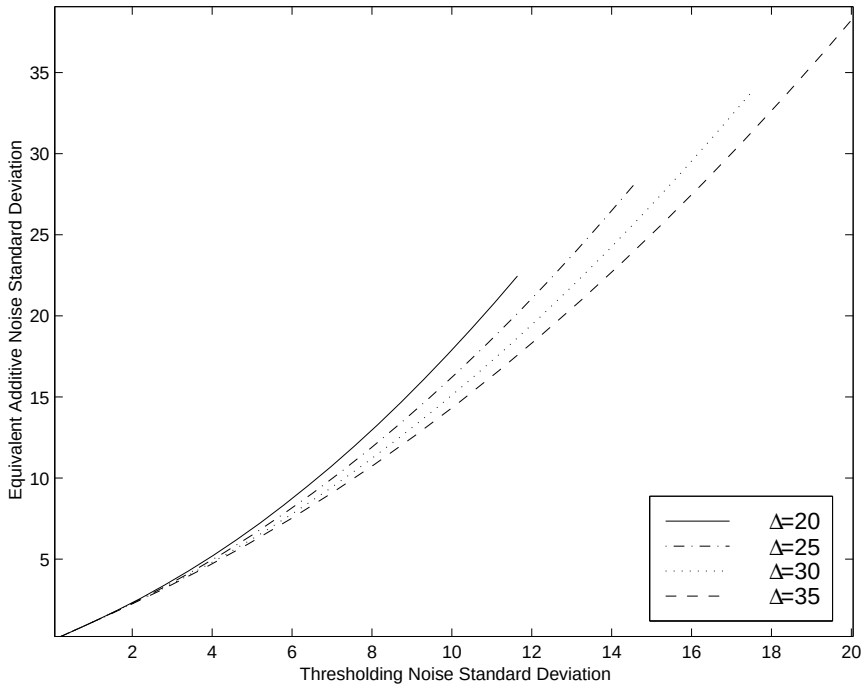

Figure 4.9 Plot of standard deviation of thresholding noise (σ_{s_t}) *vs* standard deviation of *equivalent noise* due to thresholding, ($\sigma_{s_{t_e}}$)

The equivalent additive noise, is therefore

$$\sigma^2_{s_{t_e}} = \frac{\Delta^2(1-\rho_t^2)}{12\rho_t^2} \tag{4.16}$$

The plot of σ_{s_t} *vs* $\sigma_{s_{t_e}}$ for different values of Δ is shown in Figure 4.9.

4.4.1 Combined Effect of Channel Noise and Thresholding Noise

Let the additive noise in the channel is Gaussian with variance σ_ν^2. The thresholding noise has a probability distribution given by Eq. (4.15). The probability distribution of the total noise, $\mathbf{z} = \nu + \mathbf{s}_t$, *viz* $f_Z(z)$ is obtained as

$$f_Z(z) = \int_{-\infty}^{\infty} f_\nu(x) f_{S_t}(z-x)\mathrm{d}x. \tag{4.17}$$

If $f_\nu(\nu)$ is Gaussian,

$$f_Z(z) = \frac{\beta}{\Delta} f_\nu(z) + \frac{1}{2\Delta}\left\{\mathrm{erf}\left(\frac{z+\frac{\Delta-\beta}{2}}{\sqrt{2}\sigma_\nu}\right) - \mathrm{erf}\left(\frac{z-\frac{\Delta-\beta}{2}}{\sqrt{2}\sigma_\nu}\right)\right\} \tag{4.18}$$

The normalized correlation ρ_{nt}, and hence the equivalent additive noise can then be obtained by solving

$$\rho_{nt} = \frac{2\sum_{i=0}^{\infty}\int_{\frac{i\Delta}{2}}^{\frac{(i+1)\Delta}{2}}(-1)^i(\frac{(2i+1)\Delta}{4}-z)f_Z(z)\mathrm{d}z}{\sqrt{2\sum_{i=0}^{\infty}\int_{\frac{i\Delta}{2}}^{\frac{(i+1)\Delta}{2}}(\frac{(2i+1)\Delta}{4}-z)^2 f_Z(z)\mathrm{d}z}} \tag{4.19}$$

Once again, the solution for the above integral can be obtained in terms of the Gaussian error function, and the equivalent noise variance σ_{nt}^2 is obtained from

$$\sigma_{nt}^2 = \frac{\Delta^2(1-\rho_{nt}^2)}{12\rho_{nt}^2}. \tag{4.20}$$

Figure 4.10 is a plot of the normalized correlation ρ_{nt} versus the SNR for values of $k = \Delta/\Delta_0$ ranging from 1 to 1.6. The $k = 1$ case corresponds to no thresholding (or $\Delta_0 = \Delta = \beta$). For all four plots, $\Delta_0 = 30$. This implies that the distortion introduced to embed the signature is the same for all the four cases. The plots have been obtained from simulations. The $*$'s represent the corresponding values obtained from calculating the normalized correlation from Eq. (4.19).

Note that as the channel noise increases, we need to increase the size of Δ for the optimal SNS scheme. This can be explained as follows. Let the value of an arbitrary coefficient of $\tilde{\mathbf{c}}$ be, say, 350. Further, it is known that the coefficient could not have undergone drastic modification in the channel (for example, we know that the content could have only undergone lossy compression of reasonably good quality). We can now say with a high degree of certainty that the corresponding coefficient in the original content had a value between $350-\delta$ and $350+\delta$ ($\delta = 30$, for example). For Type I methods, the self-noise of the image is directly related to the variance of the image coefficients. It should be appreciated however, that 'noise' is actually a measure of the *lack of information*. In other words, the entropy of the self-noise is equal to the entropy of the original coefficient, *given the received coefficient*. Mathematically, $\mathcal{H}$, the entropy of the self-noise, is given by

$$\mathcal{H} = h(\mathbf{c} \mid \tilde{\mathbf{c}}) \tag{4.21}$$

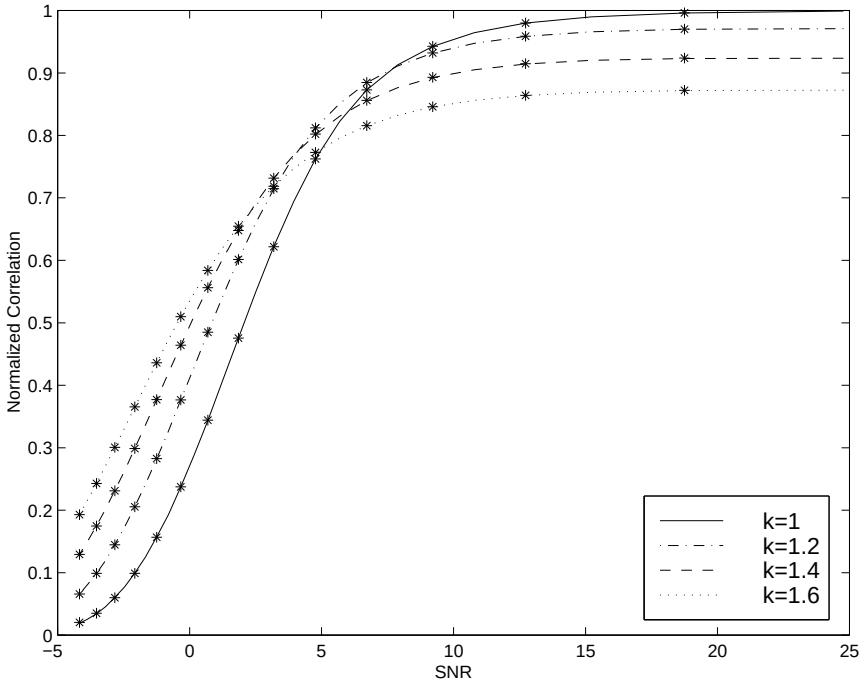

Figure 4.10 Plot of correlation *vs* SNR for $k = 1, 1.2, 1.4$ and 1.6

bits, where $h(.)$ denotes the entropy [76]. Self noise suppression schemes utilize the fact that the self-noise entropy $\mathcal{H}$ is substantially smaller than $h(\mathbf{c})$. Type II methods therefore employ some sort of *prediction* of $\mathbf{c}$ from the received signal $\tilde{\mathbf{c}}$. The period Δ can be considered as a *degree of confidence* or *tightness* of the prediction. Obviously, if the channel noise is low Δ can be small. On the other hand, if channel noise is high we need to choose larger values of Δ. However, in traditional Type II SNS methods, the choice of Δ was decided solely by the permitted distortion. The introduction of thresholding to Type II, goes a long way in overcoming that limitation.

We shall refer to the modified SNS scheme (SNS with thresholding) as a Type III method. It is interesting to note that as $\Delta \to \beta$, Type III becomes Type II. What is more interesting is that as $\Delta \to \infty$ (and β is finite), Type III systems become Type I! As Δ approaches ∞ every coefficient of $\mathbf{c}$ will be "perturbed" by $\pm\beta$. This is exactly the same as *adding* a binary ($\pm\beta$) sequence to $\mathbf{c}$!

Table 4.1 Optimal values of $k = \frac{\Delta}{\Delta_0}$ for different SNRs (SNR $= 10\log_{10}(\frac{\gamma^2}{\sigma_\nu^2})$)

SNR	k	SNR	k	SNR	k
0.00	1.87	3.01	2.57	4.77	3.14
6.02	3.59	6.99	4.04	7.78	4.40
8.45	4.78	9.03	5.11	9.54	5.41
10.00	5.71	13.01	8.10	14.77	9.95
16.02	11.51	16.99	12.85	17.78	14.10
18.45	15.20	19.03	16.25	19.54	17.30
20.00	18.20	20.97	20.40	21.76	22.30

Also note that from the trend in Figure 4.10 neither Type I nor Type II can perform as well as Type III methods. For high SNRs the "optimal" Type III method is "close" to Type II. However, as the SNR reduces, the "optimal" Type III method approaches Type I. The steps to obtain the optimal parameters for the Type III CM-SNS, for a given permitted distortion γ^2 and additive noise variance σ_ν^2, can be summarized as follows:

- Obtain $\Delta_0^2 = 16\gamma^2$.
- Let $k > 1$ such that $\Delta = k\Delta_0$.
- Evaluate β under the constraint of Eq. (4.14).
- Choose k to maximize ρ_{nt} (Eq. (4.19)).

Table 4.1 shows the optimal values of $k = \frac{\Delta}{\Delta_0}$, where $\Delta_0^2 = 12\gamma^2$, for different signal to noise ratios (SNR=$10\log_{10}(\frac{\gamma^2}{\sigma_\nu^2})$. Figure 4.11 depicts the maximum value of ρ and the corresponding theoretical capacities (assuming that the conventional signaling part that follows the SNS approaches theoretical capacity) for different SNRs, for escrow, Type III and Type II systems. The relationship between ρ and capacity is obtained by using Eq. (4.12) to obtain the variance $\sigma_{\nu_e}^2$ of the equivalent additive noise, and then using the Gaussian capacity equation [76]

$$C = \frac{1}{2}\log_2\left(1 + \frac{\gamma^2}{\sigma_{\nu_e}^2}\right). \tag{4.22}$$

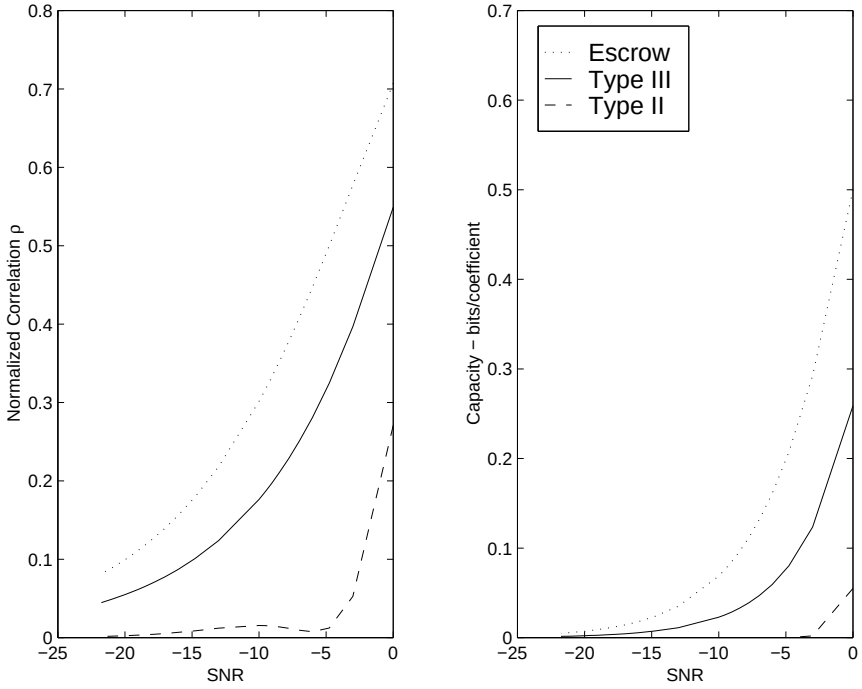

Figure 4.11 The maximum value of normalized correlation ρ (left) and corresponding capacities (right) achievable by escrow, Type III and Type II methods

Note that Type III methods can significantly outperform Type II methods (for typical SNRs of interest), and achieve about half the capacity of escrow methods.

4.4.2 Sub-optimality of Type III Methods

Even though Type III methods outperform Type I (oblivious) and Type II methods by a considerable margin, they are still not the best possible solution. To see why, consider the power constrained communication scheme modeled as

$$\tilde{\mathbf{c}} = \mathbf{c} + \mathbf{w} + \nu, \tag{4.23}$$

where $\mathbf{c}, \mathbf{w}, \nu \in \Re^N$, and $c(i) \sim \mathcal{N}[0, \sigma^2], w(i) \sim \mathcal{N}[0, \gamma^2]$ and $\nu(i) \sim \mathcal{N}[0, \sigma_\nu^2]$ $\forall$ i are i.i.d. Further $\mathbf{c}$, $\mathbf{w}$, and ν are independent. In the above model $\mathbf{w}$ is power constrained (variance γ^2), and ν is the noise in the channel. $\tilde{\mathbf{c}}$ is the signal received at the receiver, which does not have access to $\mathbf{c}$. This problem is exactly similar to the oblivious data hiding, where $\mathbf{c}$ is the content, ν is the additive noise in the

channel, and $\mathbf{w} = \hat{\mathbf{c}} - \mathbf{c}$. It is obvious, that if $\mathbf{c}$ *is available at the receiver*, one could theoretically achieve a capacity of

$$C_0 = \frac{1}{2}\log_2\left(1 + \frac{\gamma^2}{\sigma_\nu^2}\right) \tag{4.24}$$

bits per coefficient. Costa [82], however, argued that one could achieve capacity C_0 *even if* $\mathbf{c}$ *is not available at the decoder*. Unfortunately, this would require the use of codebooks of size $2^{N(C_0+L)}$ where

$$L = \frac{1}{2}\log_2\left(1 + \frac{\gamma^2(\gamma^2 + \sigma^2 + \sigma_\nu^2)}{\sigma_\nu^2(\sigma_\nu^2 + \gamma^2)}\right). \tag{4.25}$$

On the other hand, the maximum codebook size used for Type III methods (employed by the signaling scheme $(\mathcal{S}, \mathcal{S}^{-1})$ which will be described in the next chapter) is 2^{NC_0}. To get a clearer picture of the difference in complexity between the two approaches, let us consider a specific case of data hiding in 256×256 images. Some reasonable choices of $N = 8192$ (8192 transform coefficients used for data embedding), $\sigma^2 = 12000$ (variance of the low frequency coefficients used for data hiding), $\gamma^2 = 32$ (distortion of the host signal), and $\sigma_\nu^2 = 320$ imply $L \approx 40C_0$. In other words, the Type III method can achieve capacities of approximately $\frac{C_0}{2}$ (as shown in Figure 4.11) while their complexity is 2^{40} times less than methods which can approach capacity C_0. However, this does not rule out the possibility, that there may exist other suboptimal alternatives which can do better than Type III methods while maintaining reasonable signaling complexity. However, as we have already seen, other periodic functions for $(\mathcal{E}, \mathcal{D})$ may perform better than the triangular function proposed and analyzed in this paper.

Once the optimal values of Δ and β have been chosen, for a given additive noise variance σ_ν^2 and given distortion tolerance γ^2, the next step is to choose the optimal "conventional" signaling method for the equivalent noise $\sigma_{\nu_e}^2$ (or correlation ρ_{nt}). In the next chapter, we explore options for the choice of the conventional signaling method.

CHAPTER 5

FFT-BASED SIGNALING

5.1 Conventional Signaling

The conventional signaling part, viz. the pair $(\mathcal{S}, \mathcal{S}^{-1})$, addresses the problem of mapping a K length bit sequence $\mathbf{b}$ to a possibly real valued sequence $\mathbf{s}$ of length D, where $D >> K$. As a simple approach we have

$$\mathbf{s} = [\mathbf{s}_1\, \mathbf{s}_2\, \cdots \mathbf{s}_K], \tag{5.1}$$

where $\mathbf{s}_i = \text{sign}(b(i))\theta, i = 1 \cdots K$, and θ is random vector (obtained from a random seed or the private key $\mathcal{K}$), of length $\frac{D}{K}$. On the other hand, we could generate 2^K sequences $\mathbf{s}_i, i = 1 \cdots 2^K$ of length D, such that the sequences $\mathbf{s}_k$ are *maximally separable*. Geometrically, the sequences $\mathbf{s}_k$ can be represented by a set of 2^K points in a D-dimensional hypersphere. In other words, the minimum distance between any two of 2^K points should be as high as possible, under the given constraint of the hypersphere radius. The binary sequence $[b_1\, b_2\, \cdots\, b_K]$ can be interpreted as a decimal number between 0 to $2^K - 1$. To transmit a particular sequence of bits, whose decimal equivalent is say d, we choose $\mathbf{s} = \mathbf{s}_d$.

Detection of the hidden bit sequence, or equivalently the number d can be accomplished as $\tilde{d} = \arg\max_{i=0\cdots 2^K-1} \langle \tilde{\mathbf{s}}, \mathbf{s}_i \rangle$.

While it is assured that the latter scheme, will approach the *channel capacity* closer than the former, in practice, implementation of the second scheme may be prohibitively expensive, especially for large K and/or D. A reasonable compromise might be to choose an alphabet size between 2 of the former (bit-by-bit signaling) technique , and 2^K of the latter. For example, if the alphabet size is chosen as $2^{\frac{K}{k}}$, then a single member of the alphabet is detected from each of the k sequences of length $\frac{D}{k}$.

An FFT-based signaling method proposed in the next section offers an efficient way to increase the alphabet size used for signaling, while keeping the computational complexity at manageable levels. Furthermore, the maximally separable signal constellation itself is generated from random seeds.

5.2 FFT Based Signaling

In the FFT-based signaling technique, the maximally separable sequences are constrained to be orthogonal. Let $\mathbf{s}_k \in \Re^{L_k}, L_k = 2^{p_k-1}$. Maximally separable signature sequences $\mathbf{s}_k^l, l = 1 \cdots 2^{p_k}$, corresponding to p_k bits, are obtained as L_k orthogonal sequences and their negatives. *Random signature spaces* are generated from a seed. This is achieved by constraining the signatures to be *cyclic all-pass sequences*.

5.2.1 Cyclic All-Pass Sequences

Let $\mathbf{h} \in \Re^N$ and $\mathbf{H} = \mathcal{F}(\mathbf{h})$ where, $\mathcal{F}(.)$ stands for the Discrete Fourier Transform (DFT). Further, let $\mathbf{h}$ be such that

$$\mid H(n) \mid = 1 \text{ for } n = 0, 1, \cdots, N-1 \tag{5.2}$$

Hence

$$(\mathbf{H}.\mathbf{H}^*) = [1, 1, \cdots, 1]. \tag{5.3}$$

Taking the IDFT of both sides of Eq. (5.3) we get

$$\mathcal{F}^{-1}(\mathbf{H}.\mathbf{H}^*) = [1, 0, 0, \cdots, 0]. \tag{5.4}$$

As $\mathcal{F}^{-1}(\mathbf{H}.\mathbf{H}^*)$ is the *circular autocorrelation* of the vector $\mathbf{h}$, it follows that all circular shifts of $\mathbf{h}$ are mutually orthogonal [79]. As the phases ϕ_n, $n = 0, 1, \ldots, N-1$ of the elements of $\mathbf{H}$ can be arbitrary, we have infinitely many choices for the vector $\mathbf{h}$ with mutually orthogonal circular shifts. For real $\mathbf{h}$ we have $\frac{N}{2} - 1$ phase values

which can be arbitrarily chosen. Thus a pseudo-random all pass sequence of length N can be generated from a pseudo-random (uniformly distributed between π and $-\pi$) sequence of length $\frac{N}{2}-1$. If

$$\begin{aligned} \phi_k &= \begin{cases} 0 \text{ or } \pi & k=0, k=\frac{N}{2} \\ \theta_k & k=0\cdots\frac{N}{2}-1 \\ -\theta_{N-k} & k=\frac{N}{2}+1\cdots N-1 \end{cases} \\ H(k) &= \cos(\phi_k)+\mathrm{i}\sin(\phi_k), k=0\cdots N-1, \end{aligned} \tag{5.5}$$

where $\theta_k, k=1\cdots\frac{N}{2}-1$ are randomly distributed between π and $-\pi$, $\mathrm{i}=\sqrt{-1}$, then $\mathbf{h}=\mathcal{F}^{-1}(\mathbf{H})$, is a cyclic all-pass sequence.

Alternately, a pseudo-random binary sequence is generated from a seed. Then, the *unique* all-pass sequence "closest" (in the mean-square sense) to the binary sequence is obtained (this guarantees that the signature energy will not concentrated in few coefficients).

Let $\mathbf{f}=[f(0)\ f(1)\ \cdots\ f(N-1)]$ be a random binary sequence. We need to find the all-pass sequence that is closest to $\mathbf{f}$. In other words, we need to find the vector $\mathbf{h}=[h(0)\ h(1)\ \cdots h(N-1)]^T$ that minimizes the error ε defined as

$$\varepsilon=\sum_{n=0}^{N-1}\mid h(n)-f(n)\mid^2, \tag{5.6}$$

subject to the constraint that $\mathbf{h}$ is a cyclic all-pass sequence. Since the DFT of a (cyclic) all-pass sequence can be written as $\mathbf{H}=[e^{j\phi_0}\ e^{j\phi_1}\ \cdots\ e^{j\phi_{N-1}}]$, let

$$h(n)=\textstyle\sum_{k=0}^{N-1}e^{j(\frac{2\pi kn}{N}+\phi_k)} \quad f(n)=\sum_{k=0}^{N-1}a_k e^{j(\frac{2\pi kn}{N}+\theta_k)}$$

for $n=0\cdots N-1$. It can be easily shown (see Appendix) that the error ε is given by

$$\varepsilon=N\Big[N-2\sum_{k=0}^{N-1}a_k cos(\phi_k-\theta_k)+\sum_{k=0}^{N-1}a_k^2\Big]. \tag{5.7}$$

The error is minimized if we choose $\phi_k=\theta_k$ for $k=0,1,\cdots,N-1$. In other words, we choose $\mathbf{H}$ to have the same phase as $\mathbf{F}$, while the magnitude of all coefficients of $\mathbf{H}$ are set to unity.

5.2.2 Signal Constellation

The procedure employed for generating the maximally separable sequences is as follows.

1. From a random seed, generate a binary (± 1) sequence $\mathbf{e}_k$ of length $L = 2^{p_k - 1}$.
2. Obtain the length-L_k DFT $\mathbf{E}_k$ of the binary sequence.
3. Obtain $\mathbf{S}_k$ from $\mathbf{E}_k$ such that $|S_k(l)| = 1, l = 1 \cdots L_k$ and $\angle S_k(l) = \angle E_k(l), l = 1 \cdots L_k$.
4. Take the length-L_k IDFT of $\mathbf{S}_k$ to obtain $\mathbf{s}_k$. $\mathbf{s}_k$ is a *cyclic all-pass* function. All $L_k = 2^{p-1}$ cyclic shifts of $\mathbf{s}_k$ are orthogonal.
5. $\mathbf{s}_k$ and the other $L_k - 1$ cyclic shifts of $\mathbf{s}_k$, and their negatives are the 2^{p_k} maximally separable sequences.

Note that the inner product of the sequence $\mathbf{s}_k$ of length L_k with each of the $2L_k = 2^{p_k}$ maximally separable sequences can be obtained by one length-L_k cyclic correlation efficiently implemented using the FFT. The index of the maximum absolute value of the cyclic correlation coefficients gives then detected sequence of p bits. Let $0 \leq d_k \leq 2^{p_k} - 1$ be the decimal representation of $\mathbf{s}_k^d$.

$$\mathbf{s}_k^d = \begin{cases} \alpha \mathcal{C}(\mathbf{s}_k, d_k) & \text{if } d_k < 2^{p-1} \\ -\alpha \mathcal{C}(\mathbf{s}_k, d_k - 2^{p-1}) & \text{if } d_k \geq 2^{p-1} \end{cases} \tag{5.8}$$

where $\mathcal{C}(\mathbf{x}, q)$ stands for cyclic shift of the vector $\mathbf{x}$ by q (counter-clockwise) positions, and α is a scaling factor that depends on Δ of the SNS technique. For detection,

$$\mathbf{R}_k = \mathcal{F}(\mathbf{s}_k)\mathcal{F}(\tilde{\mathbf{s}}_k) \quad \mathbf{r}_k = \mathcal{F}^{-1}(\mathbf{R}_k) \tag{5.9}$$

where $\mathcal{F}$ denotes the DFT, and,

$$\tilde{d}_k \quad = \quad \begin{cases} \arg\max_{i=0 \cdots L_k - 1} |r_k(i)| & \text{if } r_k(i) > 0 \\ \arg\max_{i=0 \cdots L_k - 1} |r_k(i)| + L_k & \text{if } r_k(i) \leq 0. \end{cases}$$

An easier way of generating cyclic all-pass sequences $\mathbf{s}_k$ would be to generate them in the DFT domain by choosing unit magnitudes for DFT coefficients, but choosing the phases randomly. However, we need binary sequences of length $\frac{\Delta}{4}$ for the optimality of the self-noise suppression method employed to find the origin of the floating signal constellation. Steps 1-4 ensure that the generated signature sequences $\mathbf{s}_k$ is an all-pass sequence *closest in the mean- square sense* to the binary random sequence $\mathbf{e}_k$.

The choice of the length L_k of each segment (which in-turn decides the alphabet size) will depend mainly on the correlation ρ_{n_t} for the particular choice of Δ and β. Typically, lower the value of ρ_{n_t}, higher will be the value of L_k. Obviously, other factors like computational complexity may also influence the choice of L_k.

As the segment lengths are restricted to be powers of 2 for efficient implementation of the FFT, smooth trade-offs between bit-rate and the probability of error can only be achieved by redundant signaling. In the next section we propose a suitable and practical redundant signaling technique for improving the over-all efficiency of the signaling method.

5.2.3 Redundant Signaling

For the proposed FFT-based signaling technique, we propose a combination of Reed-Solomon coding [83] and introduction of parity for error correction. A sequence of d-bit symbols D_1 to D_n is encoded using Reed-Solomon encoding over $\mathcal{GF}(2^d)$, with block size of $2^d - 1$ (if $n < 2^d - 1$, the "shortened" code can be easily implemented by zero-padding $D_1 \cdots D_n$ to length $2^d - 1$, and considering the non-existent symbols as "erasures" at the decoder). The RS encoded sequence of d-bit symbols is then "appended" with q-parity bits to produce a p-bit symbol sequence, where $p = d + q$.

Signaling with parity can be done efficiently for the FFT-based technique. To introduce one parity bit (or reduce the valid points in the constellation by a factor of 2) we choose only odd values D between 0 and 2^{p-1} and only even values between

2^{p-1} and 2^p. This would correspond to choosing the largest from the *even-indexed* coefficients of $\mathbf{r}_k$ in Eq. (5.9). If $L_k = 2^{p-1}$ is the length of $\mathbf{r}_k$, the even indexed coefficients $\mathbf{r}_{e_k}$ of $\mathbf{r}_k$ can be obtained as (proof in Appendix)

$$\begin{aligned} R_{2_k}(l) &= R_k(l) + R_k(l + L_K/2), l = 0 \cdots \frac{L_k}{2} - 1 \\ \mathbf{r}_{e_k} &= \mathcal{F}^{-1}_{L_k/2}(0.5\mathbf{R}_{2_k}). \end{aligned} \tag{5.10}$$

In the above equation, $\mathcal{F}^{-1}_{L_k/2}(.)$ is a $\frac{L_k}{2}$- point IDFT (the factor 0.5 is irrelevant as our intention is only to pick the coefficient with the highest magnitude). For introducing q parity bits, (in the segment L_k representing p bits, where $p = q + d$) valid points in the constellation are given by

$$D = \begin{cases} m2^q - 1 & D < L_k - 1 \\ m2^q & L_k \leq D < 2L_k \end{cases} \quad m = 0, 1, \ldots, \frac{L_k}{2^q} \tag{5.11}$$

In this case, only coefficients of $\mathbf{r}_k$, with indices which are multiples of 2^q are needed. For $l = 0 \cdots \frac{L_k}{2^q} - 1$,

$$R_{q_k}(l) = \textstyle\sum_{i=0}^{2^q-1} R_k(l + i\frac{L_k}{2^q}) \quad \mathbf{r}_{q_k} = \mathcal{F}^{-1}_{L_k/2^q}(\mathbf{R}_{q_k}).$$

Signaling with parity is especially useful for very low SNR data hiding (if ρ_{n_t} in Eq. (4.19) is very small - which results in large p or L_k).

For example, let $\mathbf{c} \in \Re^{8192}$. For a low-noise scenario we may use segment lengths of $L_k = 64$ for each $p = 7$ bit symbol ($L_k = 2^{p-1}$). Under such a scenario, we may use for example two blocks of RS code (127,111) over $\mathcal{GF}(2^7 = 128)$, which can correct up to 8 errors in each block of length 127 (number of source bits = 2blocks $\times$ 111symbols per block $\times$ 7bits per symbol = 1554). However, if the SNR is low, and we use say segment sizes of $L_k = 1024$ ($p = 11$). If we do not employ parity bits, we need to use an RS code, say (2047, 2045). The maximum block size possible is however, $8192/1024 = 8$. We need a shortened code. We may start with a source of 6 11-bit symbols (66 bits), zero-padded to length 2045, and then perform (2047,2045) RS encoding, which can correct 1 error out of the 8 transmitted symbols.

Obviously this is computationally expensive. An alternative is to use $L_k = 512$ and $p = 10$, and also have say $q = 5$ parity bits. We may now start with 14 5-bit source symbols (70 bits), and zero-pad it to a length 29 symbol block. This is followed by a computationally simple RS encoding (31,29). The first 16 5-bit symbols obtained after RS encoding are then made into 10-bit symbols by introducing 5 parity bits (which is done efficiently in the FFT-based method). For detection, the parity bits are stripped first to obtain a 16 symbol sequence of 5 bit symbols. This may be zero-padded to length 31 and RS decoded.

For data hiding applications where computational complexity of detection is not a serious limitation, or if channel noise is low (implying small p), signaling with parity would be sub-optimal. However, if p is large, and $q = 0$ (or $d = p$), then RS encoding / decoding may become prohibitively expensive.

CHAPTER 6

OPTIMAL DESIGN OF DATA HIDING METHODS

In this chapter, we explore the intricacies of the duality of data hiding and data compression to help develop optimal data hiding techniques for images, that can reasonably resist lossy compression. The problem of efficient data hiding is split into two sub-problems First is to maximize the *resource* - which is the *permitted distortion* of images. The second is the *efficient use of the resource* by means of sophisticated *signaling techniques* presented in the earlier chapters.

6.1 Introduction

Growing concerns over protection of intellectual property rights of digital multimedia, has resulted in an explosive growth of the field of data hiding, or multimedia steganography. Applications of data hiding can be classified in many ways. One classification of data hiding may be based on the *key* required to extract the hidden data. For example "hidden" captions in multimedia data may be accessed through a *public key* (though there is no reason to "hide" something that can be read by anybody, using data hiding for embedding captions assures that the caption stays with the data irrespective of format conversions). On the other hand *private key* steganography is the basis for applications like *invisible watermarking* and *secret communications*. Another classification may be based on the robustness requirements of the data hiding application. For instance, applications like watermarking typically require robustness to *intentional tampering*. On the other hand, some applications may need robustness only to *unintentional* attacks (attacks not especially directed at removing the hidden data) like lossy compression. Yet another classification may be depending on the *restrictions* to be placed on data-hiding. For example, *invisible watermarking* is expected to resolve rightful ownership of the multimedia

content, unambiguously, in a court of law. For this purpose many restrictions may have to be imposed [26, 23, 22, 29] on data hiding for watermarking. On the other hand virtually no restrictions are placed on applications like *secret communications*, (communication between two private parties through a *subliminal channel* facilitated by data hiding).

We focus on data hiding applications and methods for images and video. We also restrict ourselves to applications that only require robustness to *lossy compression*. In the next section, we suggest possible applications [21] where only robustness to lossy compression is an issue, especially for *secure multimedia delivery*. We then investigate the inverse relationship between efficiency of lossy compression and efficiency of data hiding. In fact, data hiding would be impossible if lossy compressors were *ideal*. Therefore efficient data hiding should utilize *holes* in the compression methods. We explain and illustrate why, while it is very easy to develop efficient data hiding techniques if the type of compression the multimedia data is likely to undergo is known in advance, it may be very difficult to design techniques robust to *any* type of compression [21]. In Section IV we point out a hole common to *all* known compression schemes, and suggest methods to utilize that hole for efficient data hiding.

6.2 Data Hiding For Secure Multimedia Delivery

Data Hiding is expected to be a boon for multimedia content providers. Content providers can expect to communicate with *compliant multimedia players* through the subliminal channel provided by data hiding. This communication could control access, provide customized delivery, and provide solutions for pay-per-view implementations [7]. A compliant multimedia player would honor an agreed upon protocol for extracting (and abiding by) the hidden control information.

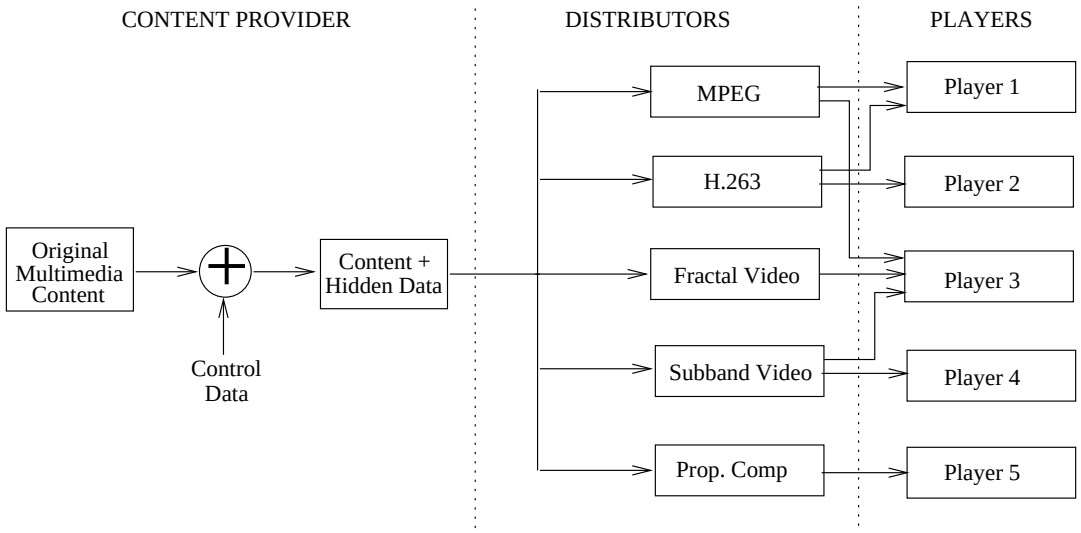

Figure 6.1 Block diagram of a multimedia distribution system. Though the generic multimedia players may support only a limited number of compression formats, all the players follow the same protocol for extracting the hidden control information. Player 3 supports 3 different formats while Player 5 supports only the proprietary compression format.

Figure 6.1 is a block diagram of a possible multimedia delivery system. Content providers (the creators of multimedia content) can hide pertinent control information for the multimedia players and make it available for distribution. The distributors may compress the content using some standard or proprietary compression method before it reaches the end users (or their multimedia players). The content may be distributed by several distributors in different formats, understandable by different players. However, as long as all such players follow an established protocol for extracting the hidden information, *and the hidden data is able to survive all lossy compression schemes the distributors may employ*, the content providers can indirectly *control* compliant players through the hidden information. Hiding the information in the raw multimedia data ensures that the hidden data stays embedded forever in the content.

Unless the hidden data is extracted with a "reasonable degree of certainty", the compliant multimedia players may refuse to play the content. Thus intentional tampering for the purpose of removing the hidden information only serves to make that particular copy of the content unusable. On the other hand, the motivation to make it robust to *all* compression methods is to facilitate more efficient distribution of the content. Failure of the hidden data to survive a "good" compression method, makes that compression method unusable for distributing that content.

6.3 Compression and Data Hiding

Multimedia compression tries to convey the information of a multimedia content as efficiently as possible - with the fewest number of bits. Data hiding on the other hand tries to sneak in additional bits of information into the content. As the "additional information" does nothing to improve the quality of the content, an ideal compressor would completely suppress the hidden information.

Let $\mathcal{I}$ represent the space of $M \times N$ images of b bits per pixel (2^{MNb} possible images). Alternately, every point in $\mathcal{I}$ is an $M \times N$ image. As the image is represented by fewer bits in the compressed domain, many original image points are mapped by the compressor to one image point after (lossy) compression and decompression. As an example, in Figure 6.2, all points in the range R are mapped to a single point D.

Consider an image A (represented by $+$) in the region R. Let us say we want to hide one bit of information in the image A that would survive compression. The space $\mathcal{I}$ is *completely* tiled by two regions that represent 0 or 1. For example, if the image A is located in a region representing 0, it could be left intact if the bit to be hidden is 0. To hide a bit 1 however, A has to be moved to a point B (represented by $*$) which simultaneously belongs to region 1 and lies *outside the range* R, so that after compression (and decompression), the image is mapped to a different point B_1.

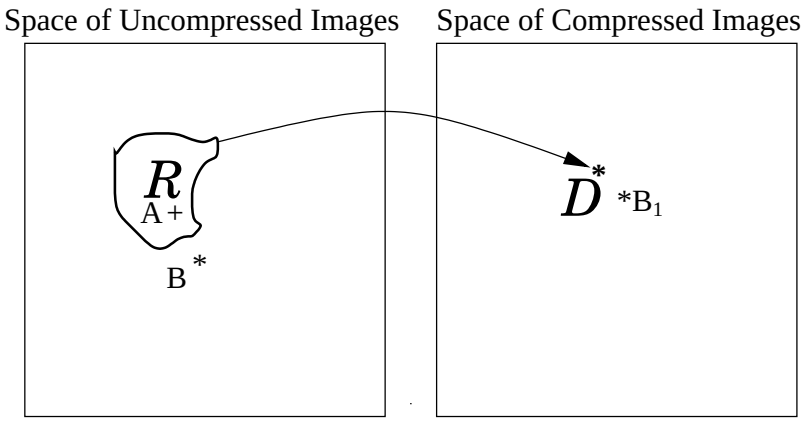

Figure 6.2 A lossy compression - decompression sequence maps all points in the range R to a single point in the domain D

To hide n_b bits in an image which can survive compression, the image has to be distorted such that after decompression the image is mapped to any of 2^{n_b} possible points. In other words, the space of images has to be tiled by 2^{n_b} regions.

Now it is easy to see that no data hiding would be possible with an *ideal* compressor. If δ_t is the visual distortion permitted (δ_t *may not* be a measure of the mean square error), then there exists a finite number of points to which the original image may be "moved". However, an ideal compressor with the same threshold $\delta_c = \delta_t$ would map all such points to a *single point* in the space of decompressed images! So unless we employ different standards (a measure of δ) for the quality of the image after data hiding and that for the decompressed image, (or unless $\delta_c > \delta_t$), *no data hiding would be possible with ideal compressors.* However, practical compression techniques are not ideal. Therefore, efficient design of data hiding should utilize the holes in compression techniques.

6.3.1 Data Hiding With Known Compression

When the compression method the image is likely to undergo is known in advance, it is easier to design efficient data hiding methods. For example, let us assume that it is

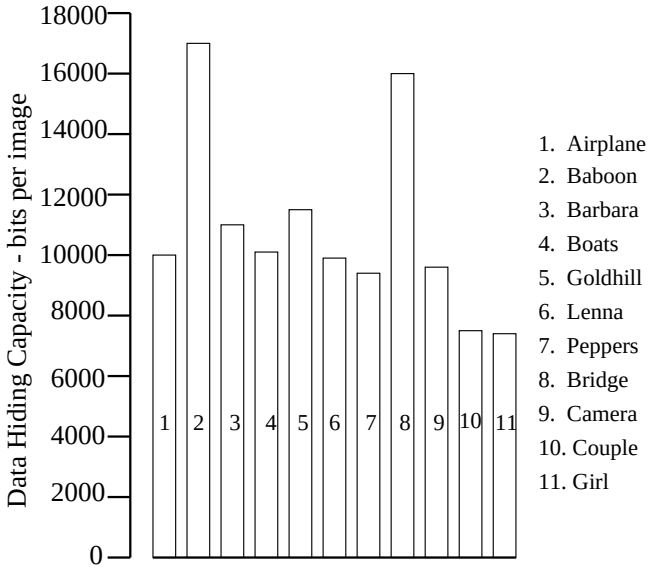

Figure 6.3 Data hiding capacities (number of DCT coefficients that quantize to a non-zero value with quantization matrix **Q**) of 11 256×256 test images

Table 6.1 The DCT quantization matrix **Q**

16	11	10	16	24	40	51	61
12	12	14	19	26	58	60	55
14	13	16	24	40	57	69	56
14	17	22	29	51	87	80	62
18	22	37	56	68	109	103	77
24	35	55	64	81	104	113	92
49	64	78	87	103	121	120	101
72	92	95	98	112	100	103	99

known in advance that the images will only undergo DCT based JPEG compression with the *default quantization matrix*. Let us also assume that image is not expected to undergo compression more severe than quality factor 50%. The best data hiding method, for such a situation would be the following [20]:

- Obtain the 8×8 2-D DCT of the image blocks of an $M \times N$ image.
- Let $Q(m, n), m = 1 \cdots 8, n = 1 \cdots 8$ be the quantization matrix for JPEG at 50 % quality. The matrix is tabulated in Table 6.1.
- Fix a particular scan order for the $\frac{M}{8} \times \frac{N}{8}$ image blocks.
- Fix a scan order for the 8×8 coefficients of each block.
- Let K be the total number of coefficients (among the $M \times N$ DCT coefficients) that quantize to a *non-zero* value when the quantization matrix $\mathbf{Q}$ is used. We shall hide one bit in each of those non-zero coefficients. (A significant amount of compression is achieved by JPEG compression due to efficient run-length coding of the coefficients that quantize to zero. So changing coefficients that quantize to zero would affect the compression ratio of the image with embedded data). Let $\mathbf{c}$ be the vector of the non-zero coefficients.
- Let $\mathbf{b}_s$ be a bit sequence of length K to be hidden in the image.
- For $i = 1 \cdots K$, if $\mathbf{b}_s(i) = 0$ then force the coefficient $\mathbf{c}(i)$ to quantize to an odd number. Otherwise force it to quantize to an even number. If the values are forced to the mid points of the quantizers, then the hidden data would survive JPEG compression of any quality as long as it is better than 50 % (if they are not forced to the *midpoints* of the quantizer steps, the hidden data will survive JPEG-50 but may not survive any higher quality compression, like JPEG-75!).

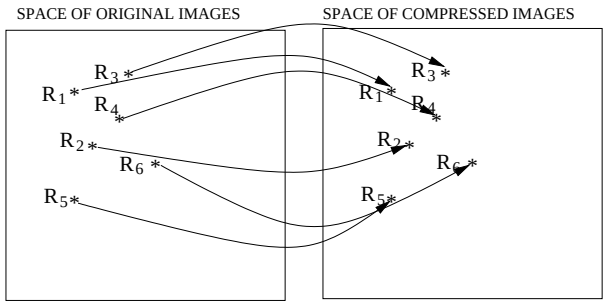

Figure 6.4 Known compression scheme

- For extracting the hidden information, the DCT of the image blocks (of the received image) are obtained. The DCT coefficients are quantized using the quantization matrix **Q**. All coefficients quantizing to zero are ignored. All other coefficients are arranged in the prescribed order. If the quantized result is odd, the hidden bit is a zero. Otherwise the hidden bit is a 1.

Figure 6.3 depicts the achievable data hiding capacities for 11 standard test images using this simple data hiding technique. However, the hidden data is very unlikely to survive other forms of lossy compression, or even if DCT based JPEG is used with a different quantization matrix.

6.3.2 Simultaneous Robustness to Multiple Compression Techniques

Consider the space $\mathcal{I}$ of original images. When the compression method is known, (as in the previous section), we make use of the fact that points (or "states") R_1 to R_n are mapped to the same points R_1 to R_n in the space of decompressed images. Therefore, the number of valid "states" of the compression method that lie within an envelope of "unnoticeable visual distortion" is a direct measure of the number of bits that can be hidden in an image (in the example above, it is the number of valid JPEG-50 compressed images within the envelope of "unnoticeable visual distortion").

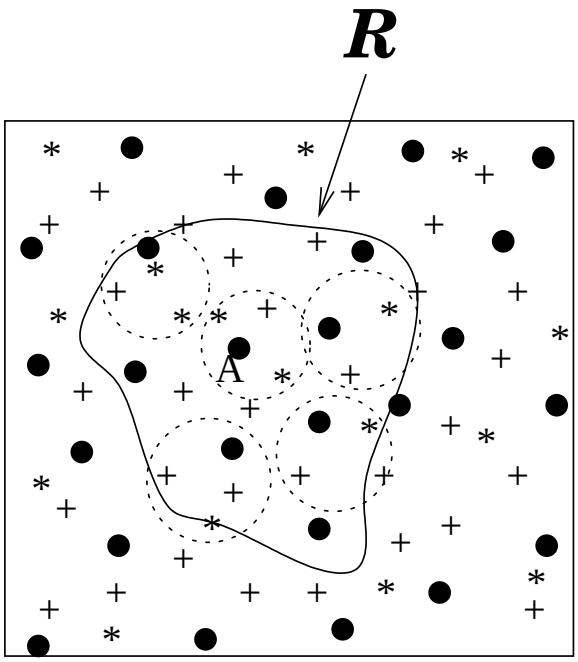

Figure 6.5 Data hiding with robustness to different compression schemes

The problem becomes more complicated if the hidden data has to survive *multiple* compression methods. To see how the requirement of robustness to different compression schemes (simultaneously) can drastically reduce the data hiding capacity, consider 3 compression schemes $\mathcal{C}_1$, $\mathcal{C}_2$, and $\mathcal{C}_3$. In Figure 6.5 the '+'s denote points in $\mathcal{I}$ which are permissible $\mathcal{C}_1$-compressed (and decompressed) images. Similarly filled 'o's and '*'s stand for $\mathcal{C}_2$ and $\mathcal{C}_3$ compressed images. Let A be the original image $\mathbf{R}$ an envelope of the possible points A could be moved to, without noticeable visual distortion. If the data hiding scheme has to survive only one of the 3 compression schemes, one can see that there are roughly 9 points to which the image can be moved in each case. However, if the hidden data has to survive any compression scheme, then the number of possible states (2^p, where p is the number of bits that can be hidden) is limited to the number of non-intersecting regions (marked by dotted circles) where at least one of the valid points of different compression schemes can be found.

6.3.3 Robustness to Unknown Compression Methods

However if the exact effect of compression is not known (the valid states are not know *a priori*), the job of designing efficient data hiding methods warrants a totally different approach. As one has no idea of the "valid" compression points (or valid compressed images for that particular compression method), the centers of the non-intersecting regions have to be considerably well separated to ensure that at least one valid compression point of all compression methods lies in each hyper-sphere. However, the following questions arise:

- Large distance between the centers of the hyper-spheres implies that it may be necessary to introduce a significant amount of distortion to move the image to a desired "state". Is it possible to do that without affecting the visual fidelity of the image?

- Assuming that it is possible to to introduce a significant amount of distortion without affecting the visual fidelity to move the image A to a new point $\hat{A}$, why should a good compressor map two *visually identical* images A and $\hat{A}$ to different points in the compressed domain?

The answer to the second question is the following.

- All known compression methods try to minimize the *mean square error* between the original and the compressed image. In fact, the new generation of compression methods (like EZW, SPIHT and IFS (fractal) image compression) even more so than the DCT based JPEG. This is a hole common to all compression methods and can be used effectively for data hiding, if satisfactory answers to the first question exist.

In the next section we explore solutions to the first question.

6.4 Utilizing the Hole in Compression Techniques

As stated in the previous section, if the images can be modified considerably in the mean square sense without affecting the visual fidelity of the image, then one could achieve large separation between "states" corresponding to different bit sequences, and thus achieve robust data hiding.

One solution to this problem (of trying to introduce as much distortion as possible without affecting the visual fidelity) is to use good models of "visual thresholds" (for example, see Ref. [53, 84]) to embed the hidden bits. Many data hiding methods [85] that utilize these models have been proposed. However, a main draw back of these methods is that well defined visual threshold models (say in the DCT or wavelet domain) also suggest the compression techniques means to improve their performance. Thus when one uses these models to add significant amount of signature energy to certain coefficients of the image, a better compression technique which may evolve in the future may also make use of these visual thresholds to perhaps *quantize those coefficients more coarsely*. In other words utilizing these visual threshold models indirectly amounts to utilizing holes that can be easily "plugged" in the future. One of the main advantages of data hiding is that the hidden data stays with the content *forever*. As compression techniques improve in the future, content *distribution* becomes more efficient. But if the hidden data is not able to survive those compression methods, the content loses its value. Therefore, more useful data hiding techniques should utilize holes which are very difficult to plug.

Figure 6.6 depicts the original 256×256 Goldhill image, its histogram reshaped version, and image after StirMark [37] (StirMark is a watermark attack software that introduces imperceptible geometric distortions in the image). Though the second and third images are very close to the original in visual fidelity, their PSNRs are 20 and 19 dB respectively! It is clear that significant amounts of distortion (in the MSE sense)

Original Image | After Histogram Modification | After StirMark

Figure 6.6 Left : original Goldhill image. Center: Goldhill image obtained by modifying the histogram. Though both images look similar, and are of good visual quality, the difference between the two images in terms of PSNR is 20 dB. Right: Image obtained after StirMark.The difference between the two images in terms of PSNR is 19 dB.

can be tolerated as long as the introduced distortion *only modifies the histogram or introduces small geometric distortions, or perhaps, both.* So if we are able to embed the hidden data by introducing geometric distortions / histogram modification, a large separation between different "states" can be obtained.

However, things may not be as simple as it seems at first glance. Let $\mathcal{H}(I)$ be a function of the histogram of the pixels of an image I. If we try to embed data by *specifying* $\mathcal{H}(I)$ [86], the hidden data will *not* be robust to compression. Even small modifications in the MSE (like what may typically be introduced by lossy compression) can change the histogram significantly. Similarly, if $\mathcal{G}(I)$ is a function of some geometric features of the image I, and $d(.,.)$ is some metric, $d(\mathcal{G}(I), \mathcal{G}(I_1))$ may be large even if $d(I, I_1)$ is small. Just as introduction of small geometric distortions can cause a significant change in the MSE, introduction of small distortions in the MSE may cause significant changes to $\mathcal{G}(.)$. This is the reason that the watermarking technique proposed by Rongen et. al [87] is robust to StirMark, but not very robust to JPEG compression. To achieve robustness to compression, the well separated

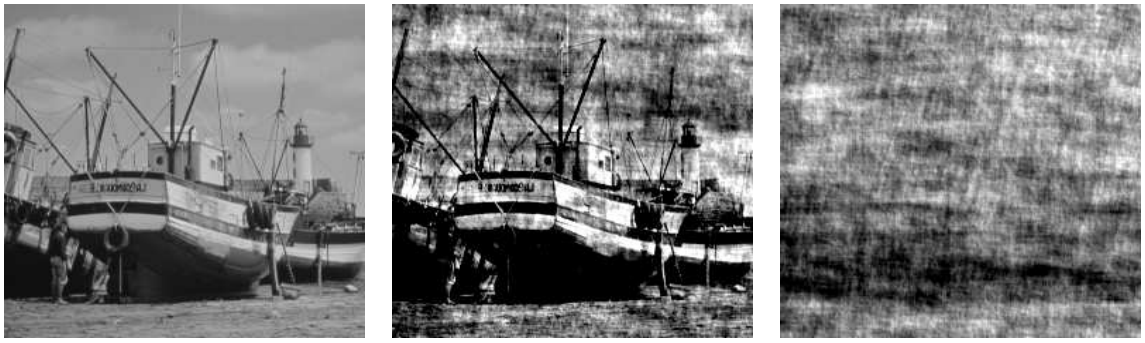

Figure 6.7 Left : original Boats image. Center: Boats image obtained by retaining the DFT phases of the original image and choosing random magnitudes (PSNR 14.1 dB). Right: image obtained by retaining DFT magnitudes of the original and choosing random DFT phases (PSNR 15.6 dB).

"states" (corresponding to the bit sequence to be embedded) have to be *specified first*. Then geometric distortions and / or histogram modifications have to be introduced to move the image close to the specified state. However, there may not be a simple or even methodical way to do this. But if such a method can be found and implemented with reasonable degree of computational complexity,[1] it promises to be an excellent solution to the problem of robust data hiding.

A practical solution to introduce a large amount of distortion in the image without affecting its visual fidelity, is to modify the DFT magnitudes. Figure 6.7 (left) shows the original 256×256 Boats image. The second image (center, 14.1 dB PSNR) was derived by retaining the DFT phases of the original image and choosing random magnitudes. In spite of the very low PSNR of the image, we see that a significant amount of "information" about the original image is preserved. The third image (right, 15.6 dB PSNR) was derived by retaining the magnitudes of the DFT coefficients of the original image but choosing the DFT phases randomly. Even though the PSNR of the third image is 1.5 dB better than that of the second, the

[1]Computational complexity of the data embedding algorithm is not a serious limitation for the applications proposed in Section II. Data embedding is done only once.

resulting image conveys almost no information about the original. This illustrates the well known fact that the human visual system (HVS) is much more sensitive to DFT phase than DFT magnitudes [88].

Thus if the data embedding is done in the magnitude DFT domain (the "states" are specified by their magnitude DFT coefficients - embedding the data changes the magnitudes of the DFT coefficients of the original image, but leaves the phase intact) a significant amount of distortion (in the MSE sense) can be introduced without affecting the visual fidelity of the image. In addition, unlike the use of well defined visual threshold models, this is not a hole that is capable of being easily "plugged" in the future (compression techniques that utilize the DFT and quantize the magnitudes coarsely and the phases finely have been proposed earlier, but have not been effective [89, 90]).

Introducing the distortion to the magnitude DFT coefficients (for embedding information bits) can be achieved as follows. Let I be the original $M \times N$ image. Let

$$I \overset{\mathcal{F}}{\Longleftrightarrow} \mathbf{I} \tag{6.1}$$

where $\overset{\mathcal{F}}{\Longleftrightarrow}$ stands for 2-D DFT pairs. $\mathbf{I}$ has 4 real coefficients and $MN-4$ complex coefficients. Only half ($D_0 = \frac{MN-4}{2}$) coefficients however, have unique magnitudes. Let $\mathbf{C} \in \Re^{D_0}$ be a vector of the unique magnitudes of the complex DFT coefficients of $\mathbf{I}$. Every image can be represented as a point in D_0-dimensional space. The D_0 magnitude DFT coefficients serve as the carriers for the subliminal communication. However, as high frequency DFT coefficients may not be able to survive lossy compression, we shall use only a subset $\mathbf{c} \in \Re^D$ of $\mathbf{C}$ for data hiding.

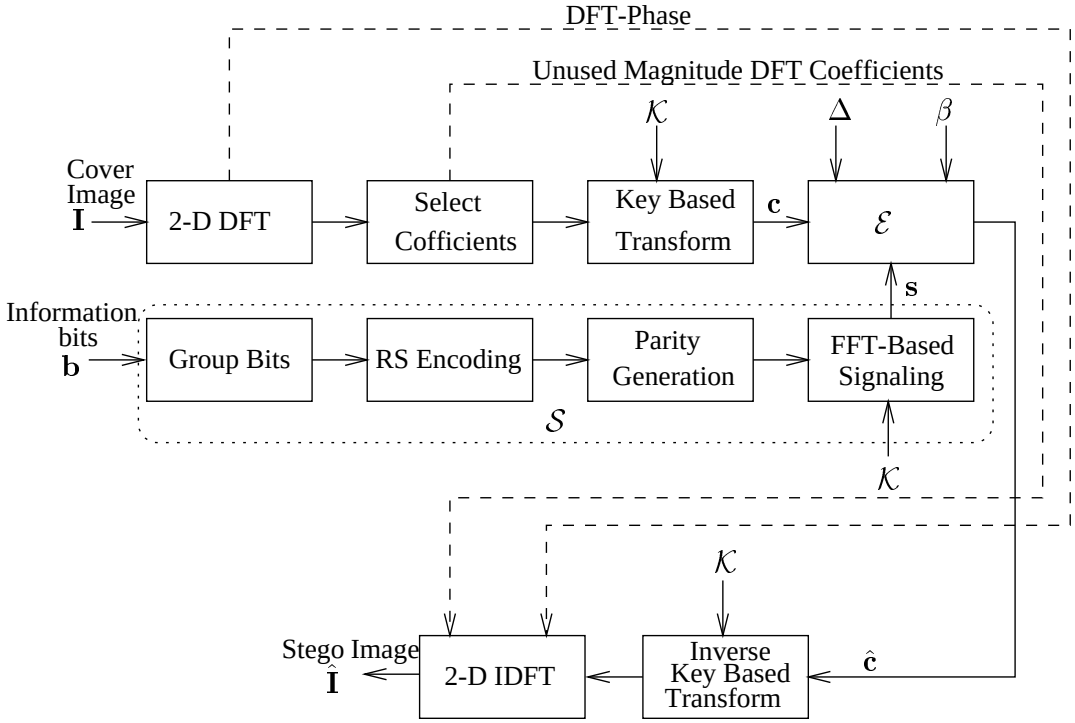

Figure 6.8 Block diagram of data embedding

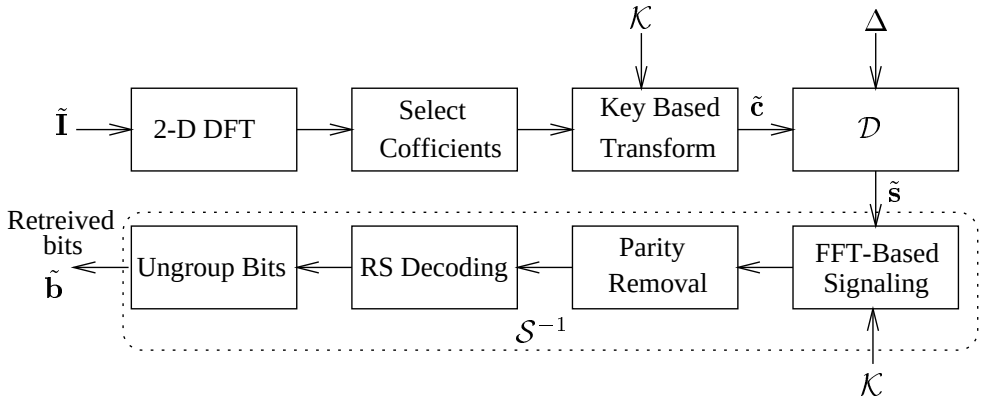

Figure 6.9 Block diagram of data detection

6.5 The Data Hiding Scheme

Figures 6.8 and 6.9 show the block diagrams of data embedding and the data detection schemes. The figures are self explanatory, except for the additional "Key Based Transform" blocks. A truly secure data hiding scheme, should be difficult to crack even if every step of the algorithm for data hiding is public. In this case, the only 'secret' should be the key $\mathcal{K}$ (though it is possible to have Δ as part of the key, as its choice is demanded by design criteria, one would not have very much freedom in choosing Δ). So if the transform employed (DFT) and the value of Δ is public, then the signature can be easily 'read', especially if binary signatures are used. While erasing hidden data may not be a very serious issue for multimedia delivery, *modifying* it may have disastrous consequences. The security can be vastly improved by using a key based transform before data embedding (and therefore before detecting). In the proposed scheme, we use a simple key based transform based on cyclic all-pass filters.

Let $\mathbf{h} \overset{\mathcal{F}}{\longleftrightarrow} \mathbf{H}$ where $\mathbf{h} \in \Re^N$ is cyclic all-pass (or $|H(k)| = 1 \forall k$). As all cyclic shifts of $\mathbf{h}$ are orthogonal, they form a basis for $\Re^N$. The basis functions are generated from the key as in Eq. (5.5). A transform employing the $\mathbf{h}$ and all its cyclic shifts as its basis can be easily implemented by cyclic correlation. If $\mathbf{x} \in \Re^N$ is a vector of coefficients, the corresponding transform coefficients $\mathbf{X}$ can be obtained as

$$\mathbf{X} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{x}).\mathcal{F}(\mathbf{h})) \tag{6.2}$$

and the inverse transform can be obtained as

$$\mathbf{x} = \mathcal{F}^{-1}(\mathcal{F}(\mathbf{X}).\mathcal{F}(\mathbf{h})^*) \tag{6.3}$$

Figure 6.10 shows the performance of the data hiding scheme for several test images undergoing JPEG (at various quality factors), and SPIHT compression (at different bit-rates). From applying JPEG at quality factors of 75, 65, 55 and 50

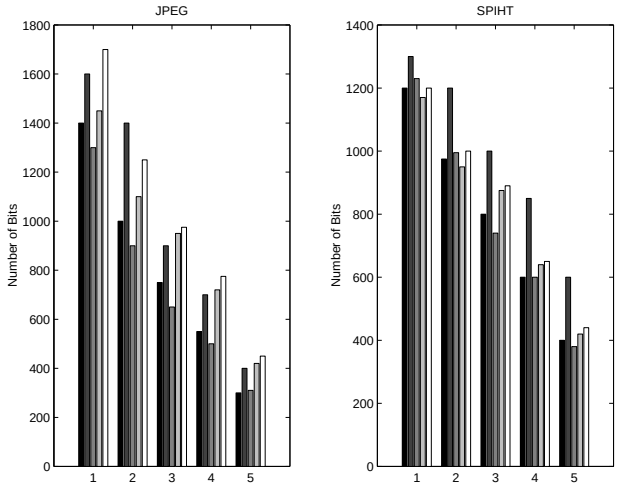

Figure 6.10 Plots of achieved data hiding capacities for JPEG (Left) and SPIHT (Right) compression for 5 256×256 test images (Lena, Barbara, Boats, Goldhill and Girl). JPEG compression scenarios 1 - 5 correspond to quality factors 75, 65, 55, 50 and 40 respectively. SPIHT compression scenarios 1 - 5 correspond to 1.35, 1.25, 1.15, 1.10 and 1.0 bpp respectively.

respectively, it was found that the resulting images on an average were compressed to 1.35, 1.25, 1.15 1.10 and 1.0 bpp respectively. So in the figure, the X-axis for both plots (JPEG and SPIHT) is an indication of the bit-rate of the compression method employed.

For all cases, we used 8192 low frequency magnitude DFT coefficients. By subjecting various images to bitrate-N compression ($N = 1 \cdots 5$, the x-axis) schemes, the average noise variances σ_ν were estimated. The permitted distortion γ, was chosen depending on the overall "activity" of the image. The measure of activity used was the MSE of the image after SPIHT compression at 1-bpp. The estimates of γ and σ_ν was used to obtain optimal values of Δ and β for each scenario.

CHAPTER 7

A ROBUST PROTOCOL FOR PROVING OWNERSHIP OF STILL IMAGES

In this chapter, we explore the problem of proving ownership or origin of digital images through watermarking. The need for *watermarking* arises out of the unsuitability of present copyright laws for claiming ownership of digital content. Watermarking schemes, however, are threatened by *counterfeit attacks*, which primarily use the *freedom* available in *choice of signature* or choice of the *watermarking method*. A *restrictive protocol* for watermarking could go a long way in rendering counterfeit attacks extremely difficult. We suggest a comprehensive protocol that makes it possible for the true owner to claim ownership unambiguously, while making it practically impossible for a pirate to do so. A robust watermarking method, compliant to the protocol, is also proposed.

7.1 Introduction

Digital watermarking is a means of protecting multimedia data from intellectual piracy. It is achieved by imperceptibly modifying the original data to insert a "signature". The signature is extracted when necessary to show proof of ownership. In this chapter, we restrict ourselves to watermarking digital images.

Let I be the original (cover) image. A watermark embedding function $\mathcal{E}$ inserts a watermark S in the image I to generate the watermarked image $\hat{I} = \mathcal{E}(I, S)$. The existence of the watermark S in an image $\tilde{I}$ is checked by a detector function $\mathcal{D}$. Watermark detectors can be broadly classified into two categories. *Cover image escrow* detectors need the original image I to check for the presence of the signature S in $\tilde{I}$. On the other hand, *oblivious detection* methods *do not* require the original

image. We shall term the output of the detector function,

$$s_d = \begin{cases} \mathcal{D}(\tilde{I}, S, I) & \text{cover image escrow} \\ \mathcal{D}(\tilde{I}, S) & \text{oblivious detection} \end{cases} \tag{7.1}$$

as the *detection statistic*. The detection statistic is an indication of the *degree of certainty* with which the signature S is detected in the image $\tilde{I}$.

Establishing ownership of creations like books or blueprints, have traditionally been done by obtaining copyright on that content, perhaps from the copyright office. However, the nature of digital content makes traditional copyright mechanisms unsuitable for establishing ownership. Figure 7.1 depicts two typical scenarios, where existing copyright mechanisms may be unsuitable for securing copyright of say, digital images. In scenario I, A and B represent two distinct but identical photographs created by Alice and Bob respectively (both photographs may have been shot from the same place at different instances of time). Alice is responsible for circulating copies of her art as A_1. Meanwhile, Bob circulates his creation as B_2. Both Alice and Bob register their contents A and B with the Copyright Office [1]. If both A and B (and hence A_1s and B_2s) look identical, Bob can claim that A and all A_1s are violations of his copyright while Alice can claim that B and B_2 are violations of her copyright. Obviously, this is not a desirable situation. In a second scenario, the photograph was created by Alice who is not *interested* in obtaining a copyright. Bob may have received a copy of A (which Alice may have made freely available on her web-site), for which he promptly obtains a copyright, and then circulates it as A_2. While it may still be acceptable for Bob to claim ownership of all A_2s (circulated by Bob) it is definitely unethical to let the copyright law enable Bob to claim ownership of the original A created by Alice. The key issue here (which cannot be determined by traditional copyright mechanisms) is to determine copies which *originate* from a *particular source*. *Watermarking the source* can effectively address this problem.

[1] to register a work of *visual art* a completed application form, a non-refundable filing fee of $30 and a non-returnable deposit of the material to be registered are to be mailed to the Copyright Office. See `http://www.loc.gov/copyright/reg.html` for more details

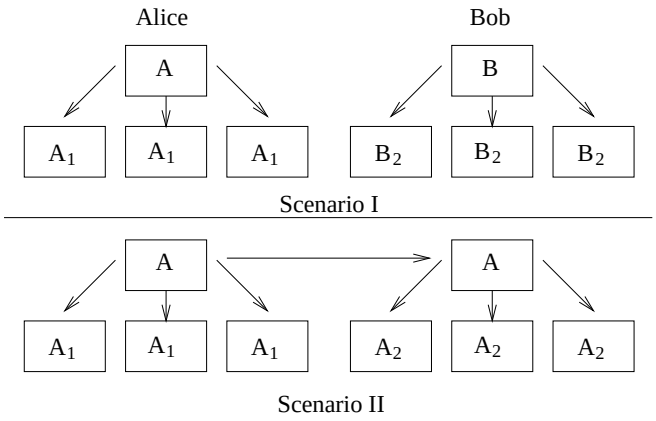

Figure 7.1 Scenarios where existing copyright laws may be Inadequate for resolving ownership. Top - scenario I: A and B are two similar photographs created by different individuals. Bottom - scenario II : creator Of A does not want to obtain copyright.

7.2 Counterfeit Attacks on Watermarks

One of the primary problems to be addressed by watermarking methods is their ability to make a counter-claim *practically impossible*. A counter-claim arises from situations where a pirate can use the *inadequacies of watermarking protocols* to "demonstrate" the presence of a his / her "watermark" (fake watermark or signature) in the actual original content. Time stamping [91, 2] has been proposed as an enhancement to the security provided by watermarking to overcome the problems associated with counter-claims in watermarks. In addition to watermarking, the creator can obtain a time stamp from a time stamping service (TSS). If the time stamp is obtained before the content is released to the public, (before the *pirate* can obtain a time stamp on the content) nobody else can claim legitimate ownership of the content. However, time-stamping have the disadvantage of requiring *ongoing involvement* of a third party. Moreover, there are some situations for which it does not provide acceptable solutions:

- Time-stamping, does not protect people who do not want to obtain time stamp and/or watermark their content, like Alice in Scenario II. If Bob is able to show a counterfeit signature in A created by Alice, and if Alice has not obtained a

time-stamp, then Bob will be able to claim ownership of content created by Alice. Clearly, time stamping does not help in situations like this.

- Time stamping simply is not a solution for time sensitive applications. The creator may not want to wait till he/she obtains a time stamp from the TSS. Obviously, time stamping cannot be used for securing live broadcasts as well.

However, we shall demonstrate, that with a suitable protocol, which would lay some (very reasonable) restrictions on watermarking algorithms, the above mentioned problems can be effectively addressed.

7.2.1 Freedom in Choice

Let Alice be the creator of the original image I. She embeds her signature S_A in I to obtain the watermarked image $\hat{I}_A = \mathcal{E}_A(I, S_A)$. The presence of her signature S_A in $\hat{I}_A$ or any image $\tilde{I}_A$ *derived* from $\hat{I}_A$ ($\tilde{I}_A = \hat{I}_A + N$) can be demonstrated with a reasonably good degree of certainty, by obtaining a sufficiently high detection statistic

$$s_{d_A} = \mathcal{D}_A(\tilde{I}_A, S_A, < I >) \tag{7.2}$$

In the above equation $< I >$ denotes that I may or may not be used by the detector. The job of Bob, an aspiring pirate, is to demonstrate the presence of his (arbitrary) signature S_B in Alice's original image I. In other words

$$s_{d_B} = \mathcal{D}_B(I, S_B, < I_1 >) \tag{7.3}$$

where I_1 may be Bob's *fake original* image. Note that Bob is at liberty to *choose his own watermarking scheme* $(\mathcal{E}_B, \mathcal{D}_B)$. If Bob has freedom in choosing his signature S_B, he could fix some $(\mathcal{E}_B, \mathcal{D}_B)$, and "construct" a signature S_B that yields a high detection statistic s_{d_B}. Note that even though Bob does not possess a copy of I (which is never released to the public by Alice), he does have $\hat{I}_A$, which is "very

close" to I. If Bob does not have freedom in choosing his signature (say, if the signature is assigned to him by a Watermarking Authority), he could still try to *tweak* the watermarking scheme $(\mathcal{E}_B, \mathcal{D}_B)$ to obtain a high detection statistic. It is obvious therefore, that a good protocol for watermarking should lay some restrictions *both on the choice of signature and choice of the embedding and detecting functions* (or the watermarking algorithm).

7.2.2 Detection Statistic

The detection statistic s_d, is a measure of degree of certainty with which the signature is detected. Typically, the signature S takes the form of a Gaussian or binary pseudo random sequence $\mathbf{s}$ (say of length N) generated from a *key* $\mathcal{K}$. The watermark embedding and detection operations can therefore be written as

$$\hat{I} = \mathcal{E}(I, \mathbf{s}) \quad \tilde{\mathbf{s}} = \mathcal{D}(\tilde{I}, < I >) \quad s_d = \frac{\mathbf{s}^T \tilde{\mathbf{s}}}{|\mathbf{s}||\tilde{\mathbf{s}}|} \tag{7.4}$$

In other words, the detection statistic is a measure of (normalized) *inner product* of the embedded and the detected signature sequence.

The inner product of randomly generated signature sequences will also be random. More specifically, for large N, the distribution of the inner product will be Gaussian $\mathcal{N}[0, \frac{1}{N}]$. If the creator (or pirate) has *absolutely no freedom* in choosing the signature, and if the detection statistic s_d obtained is say 6 times the standard deviation (if $s_d = 6\frac{1}{\sqrt{N}}$), then we could say that the signature is detected with a probability of error of less than $Q(6) \approx 1 \times 10^{-9}$. This is due to the fact that on an average only 1 out of 1×10^9 signatures chosen randomly can yield such a high correlation.

Any judge would be more than willing to rule in favor of detection of the signature, say if the probability of him/her making a wrong decision is one in a million. In this case, $s_d = 5\frac{1}{\sqrt{N}}$ is more than acceptable. However, if the pirate can find a *loop hole* in the watermarking protocol that enables him / her to *search* for

a suitable signature, then he / she has to search for one million signatures (on an average) before he / she can obtain one that yields satisfactory detection statistic.

One way to overcome this problem is to insist that the detection statistic be of the order of say $9\frac{1}{\sqrt{N}}$. This would imply that the pirate has to search for about 1×10^{19} signatures before he can obtain one which yields satisfactory correlation. If a pirate can search for say 1×10^8 signatures in a second then he/she would still need over 300 years to come up with a satisfactory signature! However, this restriction may make it considerably simpler for the pirate to *remove* the watermark by carefully planned attacks. After such attacks, the real owner may not be able to extract the signature with such a high degree of confidence (obtain high detection statistic).

7.2.3 Fake Originals

Even if the watermarking scheme *and* the choice of signature is fixed, it may still be possible for a pirate to engineer a counterfeit attack, *if the detection scheme is cover image escrow.* This would permit the pirate to create a *fake original* (cover) image, for which there are no restrictions! This problem can be solved to a certain extent if the detection method is oblivious. But some geometric attacks on images like StirMark [2] may be extremely difficult to overcome unless it is permitted to use the original image to *undo* the geometric distortions. Under this condition, the pirate may gain some freedom in *choosing* an algorithm for undoing the geometric distortions. A good watermarking protocol should also therefore, fix the algorithm to be used. However, the pirate still has *freedom in choosing the fake original which will be used by the fixed algorithm for undoing geometric distortions.* In other words, the pirate (Bob) can *engineer* a (fake) original which when used in conjunction with the fixed algorithm, can "undo the distortion" in Alice's original image I (Bob would claim that the fake original he has created to be the original, and Alice's image I,

[2] Free software available for download from `http://www.cl.cam.ac.uk`

to be an image derived from his original) to show the presence of his watermark. Again, to engineer the attack, he has the image $\hat{I}$ which is "very close" to I.

7.3 Watermarking Algorithms

To gain a better idea of the effect of counterfeit attacks on watermarking algorithms, we need to take a closer look at the model of the watermarking scheme used. Usually, the watermark is inserted in some transform domain. Let $\mathbf{C} = \mathcal{T}(I)$. More generally, only a subset of $\mathbf{c} \in \Re^N$ of $\mathbf{C}$ may be modified to embed the watermark. Let $\mathbf{C} = \mathbf{c} \cup \bar{\mathbf{c}}$, where $\mathbf{c} \cap \bar{\mathbf{c}} = \Phi$. The overall embedding and detecting operation may be expressed as

$$\begin{array}{ll} \mathbf{C} = \mathcal{T}(I) & \mathbf{C} = \mathbf{c} \cup \bar{\mathbf{c}} \\ \hat{\mathbf{c}} = \mathcal{E}(\mathbf{c}, \mathbf{s}) & \hat{\mathbf{C}} = \hat{\mathbf{c}} \cup \bar{\mathbf{c}} \\ \hat{I} = \mathcal{T}^{-1}(\hat{\mathbf{C}}) & \tilde{I} = \hat{I} + \mathbf{N} \\ \tilde{\mathbf{C}} = \mathcal{T}(\tilde{I}) & \tilde{\mathbf{C}} = \tilde{\mathbf{c}} \cup \bar{\tilde{\mathbf{c}}} \\ \tilde{\mathbf{s}} = \mathcal{D}(\tilde{\mathbf{c}}) & s_d = \frac{\mathbf{s}^T \tilde{\mathbf{s}}}{|\mathbf{s}||\tilde{\mathbf{s}}|} \end{array} \tag{7.5}$$

The watermarking algorithms that fit into the general model of Eq. (7.5) can further be classified into 3 types, Types I, II and III as in Chapter 4, depending on the embedding and detecting operators $(\mathcal{E}, \mathcal{D})$. For Type I methods, $(\mathcal{E}, \mathcal{D})$, take the form of linear addition. Mathematically, $\hat{\mathbf{c}} = \mathbf{c} + \mathbf{s}$. Type I methods can further be classified as escrow methods, where $\mathcal{D}(\tilde{\mathbf{c}}) \equiv \tilde{\mathbf{c}} - \mathbf{c}$ (for example, the method in [47]), and oblivious methods (for example, [28]), where $\mathcal{D}(\tilde{\mathbf{c}}) \equiv \tilde{\mathbf{c}}$ (no operation). Type II and Type III methods on the other hand utilize periodic functions for embedding / detecting. We also saw that optimal methods should be Type III, using continuous periodic functions. The embedder $\mathcal{E}$, characterized by a period Δ and threshold β is as follows:

$$\begin{aligned} \mathbf{p} &= \mathcal{D}(\mathbf{c}) \\ e(k) &= s(k) - p(k) \\ e(k) &= (|e(k)| > \frac{\beta}{2}) \quad ? \quad \text{sign}(e(k))\frac{\beta}{2} \quad : \quad e(k) \end{aligned} \tag{7.6}$$

$$
\begin{aligned}
e(k) &= (\text{rem}\left(\frac{c(k)}{\Delta}\right) > \frac{\Delta}{2}) \quad ? \quad -e(k) \quad : \quad e(k) \\
\hat{c}(k) &= (c(k) \geq 0) \quad ? \quad c(k) + e(k) \quad : \quad c(k) - e(k)
\end{aligned}
$$

The algorithm for $\mathcal{D}(\tilde{\mathbf{c}})$ is as follows:

$$
\begin{aligned}
q(k) &= \text{rem}(\frac{|\tilde{c}(k)|}{\Delta_k}), \quad k = 1 \cdots D \\
\tilde{s}(k) &= (q(k) \geq \frac{\Delta}{2}) \quad ? \quad (\frac{3\Delta}{4} - q(k)) \; : \; (q(k) - \frac{\Delta}{4})
\end{aligned} \tag{7.7}
$$

For high SNR's the optimal method will be a Type III which is close to Type II (β close to Δ). On the other hand, for low SNRs the optimal Type III method will be closer to Type I (large Δ and small β). As we expect the watermarks to undergo significant attacks, we would like to design the watermarking scheme for low SNRs. As an example, if one-eighth of the coefficients of some unitary transform of the image are used for watermarking, and if the permitted distortion of the image after addition of the watermark is restricted to have a peak SNR of 42 dB, then $\gamma^2 \approx 33$, implying $\Delta_0 \approx 20$. The expected attacks (σ_ν^2) is typically expected to be much larger than γ^2. So a reasonable choice may be $k = 5$ (or $\Delta = 100$) and $\beta = 12$. As the decoder does not need to know the value of β, the value of β may be chosen depending on the nature of the image. Small values of β may be chosen for very smooth images, and larger values for highly textured images. A better approach might be to choose a high value of β and obtain the watermarked image $\hat{I}_1$. The distortion introduced due to watermarking, viz. $\hat{I}_1 - I$ may then be thresholded using a reasonable visual threshold model to obtain the final watermarked image $\hat{I}$.

7.4 Aids to Overcoming Attacks on Watermarks

Conventional watermarking methods rely on the assumption that if the image is altered significantly in the mean-square error (MSE) sense, then the quality of the resultant image would be so poor that it would not warrant a ownership claim.

Therefore, most attacks on watermarks would rely on changing the image significantly in the MSE sense, without visually distorting the image. There are many ways to accomplish this - for example, scaling of pixel intensities, modifying the histogram, introducing small geometric distortions etc..

Figure 7.2 (left) shows the original Goldhill image. Figure 7.2 (center) shows the modified Goldhill image obtained by reshaping the histogram. Though both images are very similar and are of good visual quality, the difference in terms of PSNR between the two images is 20 dB! Figure 7.2 (right) shows the Goldhill image after application of StirMark which introduces imperceptible geometric distortions to the image. Application of StirMark on 15 256×256 test images yielded resulting images, of reasonably good perceptive quality, though the difference in PSNR between the original and the modified image was around 19 to 20 dB on an average.

One way to survive geometric attacks like StirMark would be to cause the watermarking method to *introduce* geometric distortions [87]. Let $\mathcal{G}(I)$ be a function of some geometric features of the image I. The watermark is can be introduced by *specifying* $\mathcal{G}(\hat{I})$. However, we cannot expect such methods to be robust to compression. Just as small geometric distortions can modify the MSE significantly, small changes in MSE (such as those that might be introduced due to lossy compression) can change $\mathcal{G}(I)$ significantly. In this light it is not surprising that the watermarking method proposed by Rongen et. al [87] is robust to StirMark, but not robust to compression. Similarly methods that specify the histogram [86] too, are not very resistant to compression. One could still use conventional watermarking methods effectively if the primary ways by which the fake original can be moved away from the original in the MSE sense can be identified, and suitable algorithms to *undo* the changes can be employed. For example, against attacks that modify the histogram, we could permit reshaping the histogram of the image in question to match the histogram of the original image before detecting the signature. Similarly

Original Image | After Histogram Modification | After StirMark

Figure 7.2 Left : original Goldhill image. Center: Goldhill image obtained by modifying the histogram. Though both images look similar, and are of good visual quality, the difference between the two images in terms of PSNR is 20 dB. Right: Image obtained after StirMark.The difference between the two images in terms of PSNR is 19 dB.

a good algorithm for detecting *salient points* of the original image and those of the image in question may be used to re-warp the image so that the salient points match, before the signature is detected. Similar algorithms could also be used to overcome pixel scaling attacks. However, only "permitted" algorithms may be used for reshaping the histogram / identifying the salient points to re-warp the image, or for rescaling the pixel values. As mentioned in Section II C, permitting freedom in choice of these algorithms would provide the pirate with additional degrees of freedom to engineer counterfeit attacks.

7.5 Restrictions on Choice of Signature

The type of restrictions for choice of signature, proposed in watermarking literature, can be classified into 3 types -

1. issued by a Watermarking Authority (Scheme I).

2. derived from a meaningful string [28] (Scheme II).

3. derived from the cover image [26] (Scheme III).

Scheme I has a major disadvantage of needing a Watermarking Authority in possession of all "secrets". The disadvantage of the Scheme II is the following; if the method of *obtaining the signature from the meaningful string* is fixed (as it should be - otherwise the whole purpose is defeated), then it may be possible for pirates to "guess" the meaningful string used by Alice, thus reducing security). In addition, both Schemes I and II suffer from the *fake original* problems illustrated in Section 3.4.

In Ref. [26], Craver *et. al* suggested a novel idea (Scheme III) to solve the fake original problem, which at one stroke solves the fake original problem and the need for an agency to issue signatures. They suggested that the signature be obtained *from the original image itself.* The original image is hashed by a fixed hash function. The output is used as a seed for a *fixed random sequence generator* to generate the signature. Tying up the signature to the original image in an inextricable way goes a long way in restricting the freedom available for the pirate to engineer counterfeit attacks. The signature is obtained as $\mathbf{s}_A = \mathcal{H}(I)$. More importantly, $\mathcal{H}(I) \neq \mathcal{H}(\hat{I})$. The watermarking scheme is cover image escrow described by Eq. (7.8).

$$\begin{array}{ll} \mathbf{s}_A = \mathcal{H}(I) & \hat{\mathbf{c}} = \mathbf{c} + \mathbf{s}_A \\ \tilde{\mathbf{s}}_A = \tilde{\mathbf{c}} - \mathbf{c} & s_d = \frac{\mathbf{s}_A^T \tilde{\mathbf{s}}_A}{|\mathbf{s}_A||\tilde{\mathbf{s}}_A|} \end{array} \tag{7.8}$$

However, Scheme III too is not entirely foolproof. At least, to be foolproof Alice should obtain very high detection statistics in Bob's image, which may not be possible in some cases. The attack for this method rests on the fact that Bob can still search for a *combination* of a fake original and its corresponding signature. Bob, who has in his possession $\hat{I}$ (or equivalently $\hat{\mathbf{c}}$), could change $\hat{I}$ significantly, in the mean-square-error sense while maintaining the "visual similarity" between the original $\hat{I}$ and the resulting (modified) image $\hat{I}_m$.

Let I_d be the difference image

$$I_d = \hat{I}_m - \hat{I} \quad \mathbf{c}_d = \hat{\mathbf{c}}_m - \mathbf{c} \tag{7.9}$$

Even though the algorithms for undoing geometric distortions / histogram modifications / pixel rescaling would not permit Bob to move very far away from $\hat{I}$, he should be able to introduce distortions such that the total power of I_d ($\mathbf{c}_d$) is much larger than that of the signature S_A ($\mathbf{s}_A$) added by Alice. Mathematically,

$$\sum_{i=1}^{N} c_d^2(k) >> \sum_{i=1}^{N} (c(i) - \hat{c}(i))^2 \tag{7.10}$$

Therefore

$$\mathbf{c}_d = \hat{\mathbf{c}}_m - \hat{\mathbf{c}} \approx \hat{\mathbf{c}}_m - \mathbf{c} \tag{7.11}$$

The next step for Bob is to derive his "original" (fake original) image from $\hat{I}_m$. Before we see how he can do that, note that the hash function $\mathcal{H}$ maps different images to (possibly) different seeds. For example if all the images in the world were of size 256×256 and restricted to 8 bits per pixel, there are still $2^{256\times256\times8}$ possible images. Though $\mathcal{H}$ would map the space of images to a (comparatively) very restricted 'space' of seeds, the space of seeds should still be large enough to ensure that the probability that different signatures are correlated is very small. Two 'obviously' different images having the same signature is not likely to create a problem. The problem only arises when images are 'similar'. So it is important that the (fixed) hash function generates different seeds especially when the images are 'similar'. So the hash function would be required to "respond" to the LSBs of image more than to the MSBs. This works to Bob's advantage.

Bob could probably generate enough (different) signature sequences from the image $\hat{I}_m$ (or $\hat{\mathbf{c}}_m$) just by tweaking 1-2 LSBs of the image pixels. But when he does that the resulting image is still very close to $\hat{I}_m$. So he would correlate every signature sequence obtained from modified versions of $\hat{I}_m$ with the fixed $\mathbf{c}_d$. Whenever

a particular "tweaking" of the bits results in a signature sequence with satisfactory correlation with $\mathbf{c}_d$, he stops. He calls the resultant image $I_m \approx \hat{I}_m$ as his "original" image. If S_B (or $\mathbf{s}_b$) is the signature generated from I_m, and $\mathbf{s}_b$ has a reasonable correlation with $\hat{\mathbf{c}} - \mathbf{c}_m$, then it can also be expected to have high correlation with $\mathbf{c} - \mathbf{c}_m$. So Bob can demonstrate the presence of his signature in I! Note that making $I_m - \hat{I}$ large swamps out the difference between I and $\hat{I}$. Let

$$\rho_e = \frac{(\hat{\mathbf{c}} - \mathbf{c}_m)^T(\mathbf{c} - \mathbf{c}_m)}{|(\hat{\mathbf{c}} - \mathbf{c}_m)||(\mathbf{c} - \mathbf{c}_m)|} \tag{7.12}$$

It can be easily seen that to generate a random signature sequence which yields a detection statistic s_d with $(\mathbf{c} - \mathbf{c}_m)$, Bob has to obtain a detection statistic of $\frac{s_d}{\rho_e}$ with $(\hat{\mathbf{c}} - \mathbf{c}_m)$. So, larger the MSE between $\mathbf{c}$ and $\mathbf{c}_m$, closer the value of ρ_e in Eq. 7.12 to unity. As the signature energy is typically very small, it would be very easy for Bob to introduce a distortion of energy more than 10 times that of the energy of the signature introduced by Alice (this would imply $\rho_e > 0.95$). After a series of carefully planned attacks on Alice's watermark in I_m, Alice, may not be able to detect her signature in I_m with a high degree of certainty. Lets assume that Alice, using a sophisticated watermarking method manages to detect her signature in I_m with $P_e \approx 3 \times 10^{-7}$, (or $s_d = 5\frac{1}{\sqrt{N}}$). To obtain a comparable detection statistic of his signature in I, Bob has to *search* roughly 3.3×10^6 sequences if $\rho_e = 1$. For $\rho_e = 0.95$ and $\rho_e = 0.90$, Bob has to search 1.5×10^7 and 7.2×10^7 signatures respectively (on an average), before obtaining a suitable signature. This is certainly computationally feasible.

7.6 Improving Scheme III

Ideally, we would like to reduce the detection threshold, to enable the content creators to claim ownership even under substantial attacks by a pirate. As mentioned earlier, if counterfeit claims do not exist, any judge would decide in favor of detection of the

watermark, even if the probability of error is 1 in 10000. The problem with Scheme III is that the complexity of the attack would also be of the same order of the accepted threshold for signature detection. With the following small modification to Scheme III we shall see that we can substantially increase the complexity of engineering a counterfeit attack:

- The watermark should be detected *without subtracting* the original image. But the original image is still necessary because the seed is obtained from the original image as in Scheme III.

- The signature should yield a high detection statistic with the image in which the signature is to be detected.

- The signature should yield a *low* detection statistic (less than a threshold δ) with the original image.

To engineer a signature, Bob again starts with I_m, obtained as earlier, and $\hat{I}$ which is a good approximation of Alice's original I. Let $\hat{s}_i$ be the detection statistic obtained as the inner-product of randomly generated signature sequences with the coefficients $\mathcal{D}(\hat{\mathbf{c}})$ of the the image $\hat{I}$ (we shall assume that that $\mathcal{E}, \mathcal{D}$ are Type III with $\Delta \approx 100$ and $\beta \approx 12$). Let s_i be the statistics of the inner-product of randomly generated signature sequences with the coefficients $\mathcal{D}(\mathbf{c})$ corresponding to the true original I. In order to show his signature in the image I with the same degree of certainty as in Method III ($P_e < 10^{-7}$), the signature should be chosen such that $s_i > 5\frac{1}{\sqrt{N}}$. However, in addition, the same signature should also yield a *low detection statistic* with Bob's (fake) "original" image I_m. Let s_{i_m} be the statistic obtained as the inner-product of randomly generated signature sequences with the coefficients $\mathcal{D}(\mathbf{c}_m)$. Obviously, the detection statistics s_i and s_{i_m} are not independent. As I and I_m are still more "similar" than "not similar", we would expect a random sequence that yields a high statistic s_i to also yield a high statistic s_{i_m}. This makes it extremely

difficult for Bob to engineer a signature. Let

$$\rho_o = \frac{\mathcal{D}(\hat{\mathbf{c}})^T \mathcal{D}(\mathbf{c}_m)}{|\mathcal{D}(\hat{\mathbf{c}})||\mathcal{D}(\mathbf{c}_m)|} \quad \rho_h = \frac{\mathcal{D}(\mathbf{c})^T \mathcal{D}(\hat{\mathbf{c}})}{|\mathcal{D}(\mathbf{c})||\mathcal{D}(\hat{\mathbf{c}})|} \tag{7.13}$$

Eq. (7.13) states that $\rho_o \times 100\%$ of the *subspaces* of $\mathcal{D}(\hat{\mathbf{c}})$ and $\mathcal{D}(\mathbf{c}_m)$ overlap. The projection of the engineered signature $\mathbf{s}_B = \mathcal{H}(I_m)$ in the subspace shared by $\mathcal{D}(\hat{\mathbf{c}})$ and $\mathcal{D}(\mathbf{c}_m)$ does not help Bob. Bob should therefore search for signatures lying in the disjoint subspace $((1-\rho_o) \times 100\%)$. At the same time the signature should also lie in the subspace common to $\mathcal{D}(\mathbf{c})$ and $\mathcal{D}(\hat{\mathbf{c}})$ (recall that Bob does not have the original I with him). Therefore, to obtain $s_i > 5\frac{1}{\sqrt{N}}$, and a *small* s_{i_m}, or to obtain $s_i - s_{i_m} > 5\frac{1}{\sqrt{N}}$, is equivalent (in terms of complexity of search) obtaining $\hat{s}_i > \frac{5}{(1-\rho_o)\rho_h\sqrt{N}}$. Like ρ_e in Eq. (7.12), the values of ρ_o and ρ_h would depend on

1. The MSE distortion Bob can introduce to move I_m "away" from $\hat{I}$ (and I), and

2. The MSE between I and $\hat{I}$, and

Let $s_{d_B} = (s_i - s_{i_m})_B$, and $s_{d_A} = (s_{i_m} - s_i)_A$. The suffixes A and B stand for Alice's and Bob's signatures respectively. For instance s_{d_A} is obtained by checking for the presence of Alice's signature $\mathbf{s}_A$ in I_m and I. To win a counterclaim, Bob has to obtain $s_{d_B} > s_{d_A}$. For the suggested Type III watermarking methods, our simulations on many test images show that it may be extremely difficult to obtain ρ_o less than 0.5. Therefore, even if ρ_h is close to unity, Bob needs to search over 10^{23} signatures before he can be reasonably sure that he can obtain a detection statistic in Alice's original image higher than Alice can obtain in his fake original.

7.7 Protocol for Robust Watermarking

We suggest the following list of restrictions to be placed on watermarking methods, in order to make them resolve rightful ownership unambiguously. The overall protocol for watermark embedding and detection are shown in Figure 7.3. All the unshaded

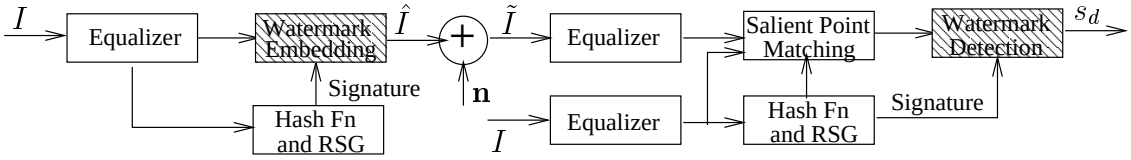

Figure 7.3 Watermark embedding and detection protocol

blocks in the figure are fixed (or regulated from time to time by the Watermarking Authority). Only the watermark embedding function and detecting function will depend on the particular watermarking algorithm.

1. A prescribed algorithm for equalizing histogram. The signature is added to the original image after equalizing its histogram. The histogram of the image in question is equalized (using the same equalizer) before performing detection of the signature.

2. A prescribed algorithm for determining significant points and re-warping the image if necessary.

3. A prescribed algorithm for determining scale factors of pixel values and re-scaling.

4. Fixed hash function $\mathcal{H}$ to be used. The hash function could be made computationally intensive to further discourage engineering of digital signatures. The hash function operates on the (histogram equalized) original image I to produce the seed $\mathcal{H}_I$.

5. The seed $\mathcal{H}_I$ is input to a *fixed random sequence generator* $\mathcal{G}$ to generate the signature sequence S_I.

$$S_N^d = \mathcal{G}(\mathcal{H}_I, N, d) \tag{7.14}$$

is the complete set of sequences that could be generated by $\mathcal{G}$. For a fixed I, the only parameters that can be changed are N - the length of the sequence, and d - the probability distribution. Probably d could take two options - Gaussian and Uniform. Another useful option for d might be to generate a list of integers from $1 \cdots N$ in a random order. This may be used for reordering the image coefficients if the algorithm calls for it. No restriction is placed on the length N.

6. Any decomposition of the original image can be used. If decompositions are generated from random sequences only one from the set of possible sequences S_N^d can be used. If the watermarking algorithm calls for a random sequence (say for re-ordering of coefficients), at any stage of the watermark embedding / extraction process, only random sequences S_N^d are permitted.

7. Signature to be extracted from the image without subtracting the original image.

8. High detection statistic of the signature with the image in which the existence of the signature is checked, *and* low detection statistic between the signature and the original image. Equivalently, the detection statistic may be considered as the difference between the detection statistics obtained from the image in question and the original.

The proposal does not limit itself only to methods in which the signature is detected by correlative processing. For example, in [16] some low frequency DCT coefficients are scrambled by a random cyclic all-pass filter. The detection statistic is obtained by counting the difference between positive and negative coefficients. The only restriction the proposal places on the method above is how the seed is obtained and the corresponding random sequence to be used to generate the all-pass filter coefficients. To our knowledge any existing oblivious detection watermarking method

(with the exception of methods [87, 86] that introduce geometric distortions or modify the histogram to introduce the watermark) can be modified to meet the requirements of the proposed protocol.

7.8 An Example Watermarking Scheme

This section outlines a possible watermarking scheme. The main purpose of this section, is to illustrate with an example, the influences of the proposed protocol in choosing parameters for the watermarking scheme. However, the section also briefly addresses other issues for increasing the security and robustness of the watermarking scheme. The block diagram of the scheme (embedding and detecting) is shown in Figure 7.4. This block diagram may be considered as closer look at the shaded blocks in Figure 7.3.

Perhaps, high GTC (Transform Coding Gain) [77] transforms like DCT or wavelet transforms are the best suited for watermarking applications. As high GTC transforms provide the most compact representation of the image, attacking DCT / wavelet coefficients for the purpose of watermark removal will most likely destroy the image. From the complete set of DCT / wavelet coefficients we choose a low to medium frequency subset for watermarking purposes. The selected coefficients undergo a key based transform (employing all-pass filters, similar to the data hiding scheme suggested in Chapter 6) to obtain the coefficients $\mathbf{c}$ to be used for embedding the signature. The signature sequence $\mathbf{s}$ to be embedded in $\mathbf{c}$ may be obtained as a pseudo-random binary sequence using the prescribed random sequence generator (RSG) trigerred by the key $\mathcal{K}$ (which in turn is derived from hashing the original image). The coefficients obtained after embedding, viz. $\hat{\mathbf{c}}$ then undergo the inverse Key-based transform to obtain the modified DCT / wavelet coefficients, which together with the unmodified coefficients are inverted to obtain the watermarked image or the stego-image.

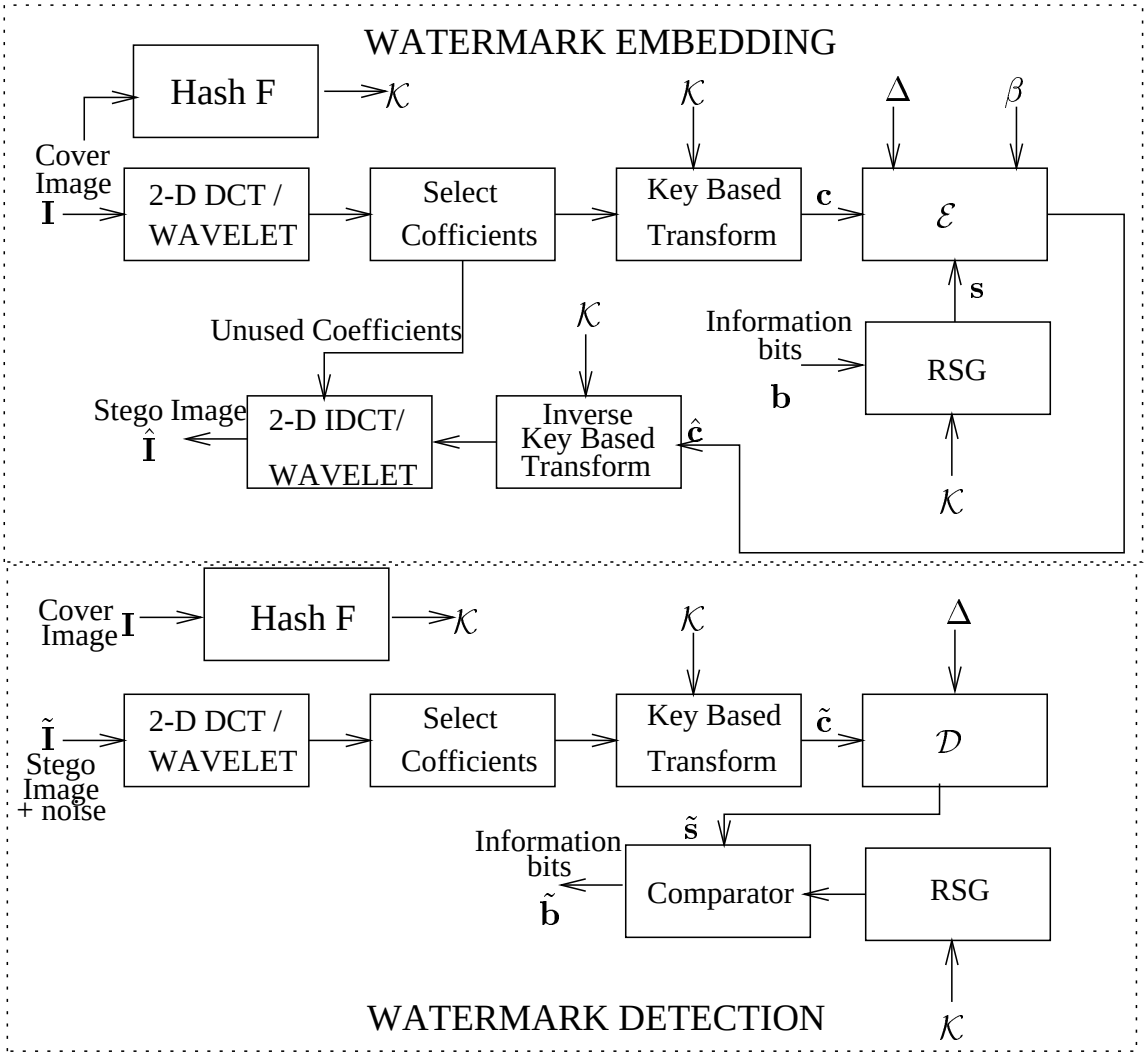

Figure 7.4 Block diagram of the watermark embedding and detection

For detection, of the signature the same operations are performed on the received noisy image $\tilde{\mathbf{I}}$ to get the corresponding coefficients $\tilde{\mathbf{c}}$. The detector function is used to extract the noisy signature sequence $\tilde{\mathbf{s}}$ which is compared with the vector $\mathbf{s}$ generated by the RSG at the receiver (using the original image I) to extract the hidden bits.

Note that any permitted watermarking algorithm should have very little freedom in choosing arbitrarily defined parameters. For example in this case, the protocol may impose a condition that all watermarking algorithms should use the same Δ (which should be chosen after a lot of thought). A less restrictive (and probably more reasonable) rule could be that the value Δ be at least 5 significant digits - while the first digit may be chosen based on the design criteria, the next 4 digits should be derived from the key $\mathcal{K}$ using the RSG.

CHAPTER 8
CONCLUSIONS AND FUTURE RESEARCH DIRECTIONS

This thesis is a comprehensive study of the issues involved in multimedia steganography, and more specifically for image and video steganography.

The thesis views the problem of data hiding as a communication system where the resource is the distortion that can be introduced without changing the original content perceivably. The distortion introduced should be used efficiently to communicate information bits by using an appropriate *signaling* technique.

The thesis first examines linear data hiding methods, which are also referred to as Type I methods. A comprehensive analysis of Type I methods is addressed in Chapter 3. We then take a novel approach to the problem of signaling for multimedia steganography and introduce the concept of *floating* signal constellations. It is seen that the new signaling method is a generalization of the so called Type II methods, based on quantization, widely used by many researchers. We further extend the Type II methods by introducing thresholding in Type II signaling methods. The extended Type II (or Type III) methods is then shown to be a generalization of both Type I and Type II methods. It is also seen that neither a (oblivious) Type I or a Type II method can be optimal. For low SNR communication channels, the optimal Type III is close to Type I. On the other hand, for high SNR channels, the optimal Type III is closer to Type II.

The fundamental difference between the classic Type II methods and the generalization proposed in this thesis is a result of the realization that it is the *periodic* nature of quantization that is useful for the signaling method, to achieve self-noise suppression. The generalization permits use of other periodic functions. We proposed two such periodic functions - a continuous triangular function (CM-SNS) and a sinusoid (CsM). Though the superiority of the two over quantization was clearly shown, the problem of finding the *best* periodic function is still open. The best

periodic function would obviously depend on the nature of the additive noise in the channel. For both Type II and Type III methods, binary sequences seem to be the best choices for signaling. However, as pointed out in Chapter 5, the optimality of the conventional signaling scheme that follows SNS may demand use of non-binary signatures. In this case a joint optimization of the SNS and and conventional signaling method is called for. This is perhaps another area for future research. The thesis also points out that Type III methods are still sub-optimal. It is well established that oblivious methods can approach the capacity of escrow methods. However, to achieve that we might need to use very large codebooks, which may not be practical. Other (probably sub-optimal) alternatives which may perform better than Type III methods is a direction for future research.

The thesis then addressed the problem of maximizing the resource - the distortion that can be introduced in the content. However, the problem has been addressed only for images. We suggested a practical option of introducing the distortion in the magnitude DFT domain. However, it appears that much more robust data hiding can be achieved if practical solutions to the inverse problem of *moving an image to a specified state* or introducing a distortion that is *close* to a *desired* distortion, by introducing imperceptible geometric distortions or histogram modification, or both, can be found.

The thesis finally addressed the problem of watermarking images for unambiguous resolution of ownership. The thesis proposes a protocol to be followed for watermarking, which can drastically increase the complexity of engineering an effective counterfeit attack.

APPENDIX A

IMPLEMENTATION OF CYCLIC 2-BAND FILTERBANKS

Let $\mathbf{h} \in \Re^N$ and $\mathbf{h} \leftrightarrow \mathbf{H}$, where $\leftrightarrow$ denotes a discrete Fourier transform (DFT) pair. Let

$$h_e(n) = h(2n), h_o(n) = h(2n+1),\ n = 0, \ldots, \frac{N}{2} - 1. \tag{A.1}$$

As $\mathbf{h}$ is orthogonal to alternate *cyclic* shifts,

$$\sum_{n=0}^{\frac{N}{2}-1} \{h_e(n)h_e(n-p) + h_o(n)h_o(n-p)\} = \delta(p). \tag{A.2}$$

Let $\mathbf{H}_e \leftrightarrow \mathbf{h}_e$ and $\mathbf{H}_o \leftrightarrow \mathbf{h}_o$. Taking the DFT of both sides of Eq. (A.2),

$$\mathbf{H}_e.\mathbf{H}_e^* + \mathbf{H}_o.\mathbf{H}_o^* = [1\ 1\ \cdots\ 1] \in \Re^{\frac{N}{2}} \tag{A.3}$$

where $(,.,)$ stands for the Hadamard product (multiplication of corresponding elements) of two vectors. It can be easily shown that the l^{th} elements of $\mathbf{H}_e$ and $\mathbf{H}_o$ are given by

$$\begin{aligned} H_e(l) &= \sum_{n=0}^{\frac{N}{2}-1} h(2n) \exp\left(\frac{-j2\pi nl}{\frac{N}{2}}\right) = \frac{H(l) + H(l+\frac{N}{2})}{2} \\ H_o(l) &= \frac{1}{2} \exp\left(\frac{j2\pi l}{N}\right) \Big[H(l) - H(l+\frac{N}{2})\Big]. \end{aligned} \tag{A.4}$$

Substituting Eqn. (A.4) into Eqn. (A.3) and simplifying,

$$|H(l)|^2 + |H(l+\frac{N}{2})|^2 = 2 \text{ for } l = 0, \cdots, \frac{N}{2} - 1. \tag{A.5}$$

Equation (A.5) is a necessary and sufficient condition for the vector $\mathbf{h}$ to be orthogonal to all its alternate circular shifts. Note that in addition to the freedom in selecting the DFT magnitudes of $\mathbf{H}$, there is complete freedom in the choice of their phases (except, of course if $\mathbf{h}$ has to be real, only $\frac{N}{2}-1$ phase values are independent). Now $\frac{N}{2}$ orthonormal basis vectors can be obtained from $\mathbf{h}$. We now want to obtain

$\frac{N}{2}$ complementary basis vectors, to complete the basis for $\Re^N$. Let $\mathbf{g}$ be a vector which is also orthogonal to its alternate shifts. Then

$$|G(l)|^2 + |G(l + \frac{N}{2})|^2 = 2 \text{ for } l = 0, \cdots, \frac{N}{2} - 1. \tag{A.6}$$

Since we desire $\mathbf{g}$ and its alternate cyclic shifts to complement the basis vectors derived from $\mathbf{h}$, $\mathbf{g}$ should further satisfy

$$\sum_{n=0}^{\frac{N}{2}-1} \{h_e(n)g_e(n-p) + h_o(n)g_o(n-p)\} = 0, \tag{A.7}$$

where, $g_e(n)$ and $g_o(n)$ are respectively the even and odd indexed elements of $\mathbf{g}$. Taking the DFT of Eqn. (A.7),

$$H_e(k)G_e^*(k) + H_o(k)G_o^*(k) = 0 \ \forall \ k. \tag{A.8}$$

Using Eqn. (A.4), and similar relations for $G_e(l)$ and $G_o(l)$, Eqn. (A.8) can be rewritten as

$$H(k)G^*(k) = -H(k + \frac{N}{2})G^*(k + \frac{N}{2}). \tag{A.9}$$

Equation (A.9) is satisfied if we choose

$$G(k) = H^*(k + \frac{N}{2}) \exp\left(\frac{j2\pi k}{N}\right) \exp(j\theta) \tag{A.10}$$

where θ is an arbitrary phase angle. Choosing $\theta = 0$, we get

$$g(n) = (-1)^{n-1}h(N - 1 - n). \tag{A.11}$$

A.1 Forward Transform

Define

$$y_h(m) = \sum_{n=0}^{N-1} x(n)h(n-m), \ m = 0, \cdots, N-1 \tag{A.12}$$

and

$$y_g(m) = \sum_{n=0}^{N-1} x(n)g(n-m), \ m = 0, \cdots, N-1. \tag{A.13}$$

Let $\mathbf{Y}_h \leftrightarrow \mathbf{y}_h$ and $\mathbf{Y}_g \leftrightarrow \mathbf{y}_g$. Taking the DFT of Eqs. (A.12) and (A.13),

$$Y_h(k) = X(k)H^*(k), \text{ and } Y_g(k) = X(k)G^*(k). \tag{A.14}$$

In view of Eqn. (A.15), we can obtain the transform coefficients $x_h(m)$ and $x_g(m)$ by sub-sampling the IDFTs of $\mathbf{Y}_h$ and $\mathbf{Y}_g$. Alternatively, from Eqs (A.12) and (A.13) we have

$$x_h(m) = y_h(2m);\ x_g(m) = y_g(2m). \tag{A.15}$$

Therefore,

$$\begin{aligned} x_h(m) &= y_h(2m) = \frac{1}{N}\sum_{k=0}^{N-1} Y_h(k)\exp\left(\frac{j4\pi mk}{N}\right) \\ &= \frac{1}{N}\sum_{k=0}^{\frac{N}{2}-1} Z_h(k)\exp\left(\frac{j4\pi mk}{N}\right), \end{aligned} \tag{A.16}$$

where

$$Z_h(k) = Y_h(k) + Y_h(k+\frac{N}{2}),\ k = 0,\ldots,\frac{N}{2}-1. \tag{A.17}$$

Similarly,

$$x_g(m) = \frac{1}{N}\sum_{k=0}^{\frac{N}{2}-1} Z_g(k)\exp\left(\frac{j4\pi mk}{N}\right). \tag{A.18}$$

where

$$Z_g(k) = Y_g(k) + Y_g(k+\frac{N}{2}),\ k = 0,\ldots,\frac{N}{2}-1. \tag{A.19}$$

Thus $x_h(m)$ and $x_g(m)$ can be determined by computing the $\frac{N}{2}$ -point IDFTs of $\mathbf{Z}_h$ and $\mathbf{Z}_g$, instead of computing the N-point IDFTs of $\mathbf{Y}_h$ and $\mathbf{Y}_g$ and sub-sampling them.

The implementation of the forward transform of $\mathbf{x}$ thus consists of the following steps

1. Obtain the DFT $\mathbf{X}$ of $\mathbf{x}$.

2. Compute the Hadamard products $\mathbf{Y}_h = \mathbf{X}.\mathbf{H}^*$ and $\mathbf{Y}_g = \mathbf{X}.\mathbf{G}^*$.

3. Split the N-vector $\mathbf{Y}_h$ into two $\frac{N}{2}$-vectors and add them to obtain the $\frac{N}{2}$-vector $\mathbf{Z}_h$. Form the $\frac{N}{2}$-vector $\mathbf{Z}_g$ from the N-vector $\mathbf{Y}_g$ in a similar fashion.

4. Obtain $\mathbf{x}_h$ and $\mathbf{x}_g$ as the IDFTs of $\mathbf{Z}_h$ and $\mathbf{Z}_g$ respectively.

A.2 Inverse Transform

Let $\mathbf{X}_h$ and $\mathbf{X}_g$ denote the periodic extensions of the $\frac{N}{2}$-point DFTs of $\mathbf{x}_h$ and $\mathbf{x}_g$ respectively, i.e.,

$$X_h(k) = \sum_{m=0}^{\frac{N}{2}-1} x_h(m) \exp\left(\frac{-j4\pi km}{N}\right), \ k = 0, \ldots, N-1, \tag{A.20}$$

$$X_g(k) = \sum_{m=0}^{\frac{N}{2}-1} x_g(m) \exp\left(\frac{-j4\pi km}{N}\right), \ k = 0, \ldots, N-1, \tag{A.21}$$

It can be shown that (see Appendix B)

$$x(n) = \frac{1}{N} \sum_{k=0}^{N-1} [X_h(k)H(k) + X_g(k)G(k)] \exp\left(\frac{j2\pi nk}{N}\right) \tag{A.22}$$

The implementation of the inverse transform therefore, consists of the following steps:

1. Obtain the $\frac{N}{2}$ length DFTs of $\mathbf{x}_h$ and $\mathbf{x}_g$.

2. Make periodic extensions of these DFTs to length N to obtain $\mathbf{X}_h$ and $\mathbf{X}_g$.

3. Compute the Hadamard products $\mathbf{X}_h.\mathbf{H}$ and $\mathbf{X}_g.\mathbf{G}$.

4. Compute the IDFT of $\mathbf{X}_h.\mathbf{H} + \mathbf{X}_g.\mathbf{G}$ to obtain $\mathbf{x}$.

APPENDIX B
MATHEMATICAL PROOFS

B.1 Proof of Eq (5.7)

$$h(n) = \sum_{k=0}^{N-1} e^{j(\frac{2\pi kn}{N}+\phi_k)} \quad f(n) = \sum_{k=0}^{N-1} a_k e^{j(\frac{2\pi kn}{N}+\theta_k)}$$

for $n = 0 \cdots N-1$. From Eq (5.6),

$$\begin{aligned}
\varepsilon &= \sum_{n=0}^{N-1}\sum_{l=0}^{N-1}\sum_{k=0}^{N-1}\left[e^{j(\frac{2\pi kn}{N}+\phi_k)} - a_k e^{j(\frac{2\pi kn}{N}+\theta_k)}\right] \times \left[e^{-j(\frac{2\pi ln}{N}+\phi_l)} - a_l e^{-j(\frac{2\pi ln}{N}+\theta_l)}\right] \\
&= \sum_{k=0}^{N-1}\sum_{l=0}^{N-1}\left(\sum_{n=0}^{N-1}\left[e^{j(\frac{2\pi(k-l)n}{N})}e^{j(\phi_k-\phi_l)} - a_l e^{j(\frac{2\pi(k-l)n}{N})}e^{j(\phi_k-\theta_l)}\right.\right. \\
&- \left.\left. a_k e^{j(\frac{2\pi(k-l)n}{N})}e^{j(\theta_k-\phi_l)} + a_k a_l e^{j(\frac{2\pi(k-l)n}{N})}e^{j(\theta_k-\theta_l)}\right]\right). \qquad \text{(B.1)}
\end{aligned}$$

Using the identity

$$\sum_{n=0}^{N-1} e^{j(\frac{2\pi(k-l)n}{N})} = \begin{cases} N & \text{for } k = l \\ 0 & \text{otherwise} \end{cases}, \qquad \text{(B.2)}$$

Eq (B.1) reduces to

$$\varepsilon = N\left[N - 2\sum_{k=0}^{N-1} a_k cos(\phi_k - \theta_k) + \sum_{k=0}^{N-1} a_k^2\right]. \qquad \text{(B.3)}$$

B.2 Proof of Eq (5.10)

Given that $\mathbf{h} \in \Re^N$, $\mathbf{H} = \mathcal{F}_N(\mathbf{h})$, and $h_e(n) = h(2n)$ for $n = 0, \ldots, \frac{N}{2}-1$, and $\mathbf{H}_e = \mathcal{F}_{N/2}(\mathbf{h}_e)$, we need to show

$$H_e(l) = \frac{H(l) + H(l+\frac{N}{2})}{2},\ l = 0 \cdots \frac{N}{2} - 1. \qquad \text{(B.4)}$$

$$\begin{aligned}
H_e(l) &= \sum_{n=0}^{\frac{N}{2}-1} h(2n) \exp\left(\frac{-j2\pi nl}{\frac{N}{2}}\right) \qquad \text{(B.5)} \\
&= \sum_{n=0}^{\frac{N}{2}-1} \frac{1}{N}\sum_{k=0}^{N-1} H(k) \exp\left(\frac{j4\pi nk}{N}\right)\exp\left(\frac{-j4\pi nl}{N}\right) \\
&= \frac{1}{N}\sum_{k=0}^{N-1} H(k) \sum_{n=0}^{\frac{N}{2}-1} \exp\left(\frac{j4\pi n(k-l)}{N}\right) \\
&= \frac{H(l) + H(l+\frac{N}{2})}{2},\ l = 0 \cdots \frac{N}{2} - 1.
\end{aligned}$$

B.3 Proof of Eq. (A.22)

$$\begin{aligned} x(n) &= \frac{1}{N}\sum_{k=0}^{N-1}[X_h(k)H(k) + X_g(k)G(k)]\exp\left(\frac{j2\pi nk}{N}\right) \\ &= T_1(n) + T_2(n). \end{aligned} \tag{B.6}$$

Consider the first term, $T_1(n)$ of (B.6),

$$T_1(n) = \frac{1}{N}\sum_{k=0}^{N-1} X_h(k)H(k)\exp\left(\frac{j2\pi nk}{N}\right) \tag{B.7}$$

$$= \frac{1}{N}\sum_{k=0}^{N-1}\sum_{m=0}^{\frac{N}{2}-1} x_h(m)\exp\left(\frac{-j4\pi mk}{N}\right)H(k)\exp\left(\frac{j2\pi nk}{N}\right) \tag{B.8}$$

As $x_h(m) = y_h(2m)$, and $\boldsymbol{y}_h \leftrightarrow \boldsymbol{Y}_h$, we have

$$x_h(m) = \frac{1}{N}\sum_{l=0}^{N-1} Y_h(l)\exp\left(\frac{j4\pi ml}{N}\right) \tag{B.9}$$

Substituting for $x_h(m)$ from Eq. (B.9) into Eq. (B.8), we obtain

$$\begin{aligned} T_1(n) &= \frac{1}{N^2}\sum_{k=0}^{N-1}\sum_{m=0}^{\frac{N}{2}-1}\sum_{l=0}^{N-1} Y_h(l)H(k)\exp\left(\frac{j2\pi[2ml+nk-2mk]}{N}\right) \\ &= \frac{1}{N}\sum_{l=0}^{N-1}\sum_{m=0}^{\frac{N}{2}-1}\left\{\frac{1}{N}\sum_{k=0}^{N-1} H(k)\exp\left(\frac{j2\pi k(n-2m)}{N}\right)\right\} Y_h(l)\exp\left(\frac{j4\pi ml}{N}\right) \\ &= \frac{1}{N}\sum_{l=0}^{N-1} Y_h(l)\sum_{m=0}^{\frac{N}{2}-1} h(n-2m)\exp\left(\frac{j4\pi ml}{N}\right). \end{aligned}$$

For even n, i.e. $n = 2q$, we have $h(n-2m) = h_e(q-m)$ (see Eq. (A.1)). Therefore,

$$\begin{aligned} T_1(2q) &= \frac{1}{N}\sum_{l=0}^{N-1}\left\{\sum_{m=0}^{\frac{N}{2}-1} h_e(q-m)\exp\left(\frac{j4\pi ml}{N}\right)\right\} Y_h(l) \\ &= \frac{1}{N}\sum_{l=0}^{N-1}\left\{\sum_{p=q}^{q+1-\frac{N}{2}} h_e(p)\exp\left(\frac{-j4\pi lp}{N}\right)\right\} Y_h(l)\exp\left(\frac{j4\pi lq}{N}\right). \\ &= \frac{1}{N}\sum_{l=0}^{N-1} H_e(l)Y_h(l)\exp\left(\frac{j4\pi lq}{N}\right). \end{aligned} \tag{B.10}$$

Substituting for $H_e(l)$ from Eq. (A.4) into Eq. (B.10),

$$T_1(n) = \frac{1}{N}\sum_{l=0}^{N-1}\frac{1}{2}\left[H(l) + H(l+\frac{N}{2})\right] Y_h(l)\exp\left(\frac{j2\pi ln}{N}\right) \text{ for even } n. \tag{B.11}$$

Similarly it can be easily shown that

$$T_1(n) = \frac{1}{N}\sum_{l=0}^{N-1}\frac{1}{2}\left[H(l) - H(l+\frac{N}{2})\right]Y_h(l)\exp\left(\frac{j2\pi ln}{N}\right) \text{ for odd } n. \qquad \text{(B.12)}$$

Similar expressions can be derived for $T_2(n)$ to obtain

$$T_2(n) = \begin{cases} \frac{1}{N}\sum_{l=0}^{N-1}\frac{1}{2}\left[G(l) + G(l+\frac{N}{2})\right]Y_g(l)\exp\left(\frac{j2\pi ln}{N}\right) \\ \frac{1}{N}\sum_{l=0}^{N-1}\frac{1}{2}\left[G(l) - G(l+\frac{N}{2})\right]Y_g(l)\exp\left(\frac{j2\pi ln}{N}\right) \end{cases}$$

for even and odd n respectively.

In view of Eqs. (A.10) and (A.5)),

$$H^*(l)H(l+\frac{N}{2}) + G^*(l)G(l+\frac{N}{2}) = 0. \qquad \text{(B.13)}$$

$$|H(l)|^2 + |G(l)|^2 = 2. \qquad \text{(B.14)}$$

Combining Eq. (A.14), viz.,

$$Y_h(k) = X(k)H^*(k), \text{ and } Y_g(k) = X(k)G^*(k),$$

with the equations for $T_1(n)$ and $T_2(n)$, and using Eqs. (B.13) and (B.14),

$$T_1(n) + T_2(n) = \frac{1}{N}\sum_{l=0}^{N-1}X(l)\exp\left(\frac{j2\pi ln}{N}\right) = x(n). \qquad \text{(B.15)}$$

REFERENCES

1. R.J. Anderson, F.A. Petitcolas, "On the Limits of Steganography", *IEEE Journal on Selected Areas of Communications*, vol **16**, No 4, pp 474-481, May 1998.

2. B. Schneier, *Applied Cryptography*, Second Edition, Wiley & Sons, 1996.

3. G.J. Simmons, "The History of Subliminal Channels", *IEEE Journal on Selected Areas of Communications*, vol **16**, No 4, pp 452-461, May 1998.

4. G.J. Simmons, "Results Concerning Bandwidth of Subliminal Channels", *IEEE Journal on Selected Areas of Communications*, vol **16**, No 4, pp 462-473, May 1998.

5. K. Mantusi, K. Tanaka, "Video Steganography: How to secretly embed a signature in a picture?", *Proceedings of IMA Intellectual Property Project*, Interactive Multimedia Association, Annapolis, MD, pp 263-272, 1994.

6. J. Zhao, E. Koch, C. Luo, "In Business Today and Tomorrow", *Communications of the ACM*, vol **41**, No 7, pp 66-72, July 1998.

7. A. H. Tewfik, White Paper on Data Embedding, Media Annotation and Copyright Protection Product Line, available for download from `http://www.cognicity.com`.

8. J. Brassil, S. Low, N. Maxemchuk, L. Ó Gorman, "Electronic Marking and Identification Techniques to Discourage Document Copying", *IEEE Journal on Selected Areas in Communication*, vol **13**, No 8, pp 1495-1504, October 1995.

9. M. Ramkumar, A.N. Akansu, "Information Theoretic Bounds for Data Hiding in Compressed Images", *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, California, USA, pp 267-272, December 1998.

10. M.Ramkumar, A.N. Akansu, " Theoretical Capacity Measures for Data Hiding in Compressed Images", *SPIE Multimedia Systems and Applications*, Boston, MA, **3528**, pp 482 - 492, November 1998.

11. M.Ramkumar, A.N Akansu, "Capacity Estimates for Data Hiding in Compressed Images", Submitted to the *IEEE Trans. on Image Processing.*

12. M.Ramkumar, A.N. Akansu, A. Alatan, "On the Choice of Transforms for Data Hiding in Compressed Video", *IEEE International Conference on Acoustics, Speech and Signal Processing*, Phoenix, Arizona, vol **VI**, pp 3049 - 3052, March 1999.

13. A.Said and W.A.Pearlman, "A New Fast and Efficient Implementation of an Image Codec Based on Set Partitioning in Hierarchical Trees", *IEEE Transactions on Circuits and Systems for Video Technology*, vol **6**, pp. 243-250, June 1996.

14. Joan L. Mitchell, Didier Le Gall and Chad Fogg, *MPEG Video Compression Standard*, Chapman & Hall, 1996

15. M.Ramkumar, A.N. Akansu, "A Robust Scheme for Oblivious Detection of Watermarks/ Data Hiding in Still Images", *SPIE Multimedia Systems and Applications, Boston*, MA, vol **3528**, pp 474 - 481, November 1998.

16. M.Ramkumar, A.N. Akansu, "Self-Noise Suppression Schemes for Blind Image Steganography", *SPIE Multimedia Systems and Applications (Image Security)*, vol **3845**, Boston, MA, September. 1999.

17. M.Ramkumar, A.N. Akansu, "Floating Signal Constellations for Multimedia Steganography", submitted to the *IEEE International Conference on Communications*, 2000.

18. M.Ramkumar, A.N Akansu, "Optimal Signaling for Multimedia Steganography", submitted to the *IEEE Trans. on Signal Processing.*

19. M.Ramkumar, A.N. Akansu, "FFT-Based Signaling for Multimedia Steganography", submitted to the *IEEE International Conference on Acoustics, Speech and Signal Processing*, 2000.

20. M. Ramkumar, A.N. Akansu, "Optimal Design of Data Hiding Methods Robust to Lossy Compression", submitted to the *IEEE Trans. on Multimedia.*

21. M.Ramkumar, A.N Akansu, "On the Design of Robust Data Hiding Systems", to be presented at the 33rd *ASILOMAR Conference on Signals, Systems and Computers*, Pacific Grove, CA, October 1999.

22. M. Ramkumar, A.N. Akansu, "Robust Protocols for Proving Ownership of Still Images", submitted to the *IEEE Trans. on Multimedia.*

23. M.Ramkumar, A.N Akansu, "Image Watermarks and Counterfeit Attacks : Some Problems and Solutions", *Content Security and Data Hiding in Digital Media*, Newark, NJ, pp 102-112, May 1999.

24. M.Ramkumar, A.N. Akansu, "A Robust Protocol for Proving Ownership of Still Images", Submitted to the *International Conference on Information Technology: Coding and Computing*, 2000.

25. C. Cachin, "An Information Theoretic Model for Steganography", *2nd Workshop on Information Hiding*, Lecture Notes in Computer Science, Springer, 1998.

26. S. Craver, N. Memon, B-L. Yeo, M.M. Yeung, "Can Invisible Watermarks Resolve Rightful Ownerships", *IS & T/ SPIE Electronic Imaging: Human Vision and Electronic Imaging*, vol **3022**, pp 310-321, February 1997.

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

28. W. Zeng, B. Liu, "On Resolving Rightful Ownerships of Digital Images by Invisible Watermarks", *IEEE International Conference on Image Processing*, vol **1**, pp 552-555, 1997.

29. N.Memon, P.W. Wong, "A Buyer-Seller Watermarking Protocol", *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, California, USA, pp 273-278, December 1998.

30. M. Schneider, S.-F. Chang, "A Robust Content Based Digital Signature for Image Authentication", *IEEE International Conference on Image Processing*, **3**, pp 227-230, 1996.

31. R.B Wolfgang, E.J. Delp, "A Watermark for Digital Images" *IEEE International Conference on Image Processing*, vol **3**, pp 219-222, 1996.

32. D. Kundur, D. Hatzinokos, "Towards a Telltale Watermarking Technique for Tamper-Proofing", *IEEE International Conference on Image Processing*, Chicago, Illinois, vol **2**, pp 409-413, October 1998.

33. J. Fridich, "Image Watermarking and Tamper Detection", *IEEE International Conference on Image Processing*, Chicago, Illinois, vol **2**, pp 404-408, October 1998.

34. J. Kilian, F.T. Leighton, L.R. Matheson, T.G. Shamoon, R.E.Tarjan, F. Zane, "Resistance of Digital Watermarks to Collusive Attacks", *IEEE International Symposium on Information Theory*, pp.271-280, 1998.

35. I.J. Cox, J.P. Linnartz, "Some General Methods for Tampering with Watermarks", *IEEE Journal on Selected Areas in Communications*, **16**, No 4, pp 587-593, May 1998.

36. F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, "Attacks on Copyright Marking Systems", *Second Workshop on Information Hiding*, vol **1525**, Lecture Notes in Computer Science, Portland, Oregon, pp 218-238, April 1998.

37. M. Kutter and F. A. P. Petitcolas. "A Fair Benchmark for Image Watermarking Systems", *Electronic Imaging: Security and Watermarking of Multimedia Contents*, vol. **3657**, pp. 226-239, San Jose, CA, USA, January 1999.

38. R.G. van Schyndel, A.Z. Tirkel, C.F. Osborne, "A Digital Watermark", *IEEE International Conference on Image Processing*, vol **2**, pp 86-90, 1994.

39. G. Caronni, "Assuring Ownership Rights for Digital Images", *Proceedings of Reliable IT Systems*, VIS-95, Vieweg Publishing Company, 1995.

40. W. Bender, D. Gruhl, N. Morimoto, "Techniques for Data Hiding", *Proceedings of SPIE*, vol **2420**, pp 40-50, February 1995.

41. I. Pitas, T.H. Kaskalis, "Applying Signatures on Digital Images", *IEEE Workshop on Nonlinear Image and Signal Processing*, Neos Marmaras, Greece, pp 460-463, June 1995.

42. N. Nikolaidis, I. Pitas, "Copyright Protection of Images Using Robust Digital Signatures", *IEEE International Conference on Acoustics Speech and Signal Processing*, vol **4** pp 2168-2171, May 1996.

43. R.B. Wolfgang, E.J. Delp, "A Watermark for Digital Images", *International Conference on Image Processing*, Lausanne, Switzerland, vol **3**, pp 219-222, September 1996.

44. R.B. Wolfgang, E.J. Delp, "A Watermarking Technique for Digital Imagery: Further Studies", *International Conference on Imaging Science - Systems and Technologies*, Las Vegas, Nevada, pp 279-287, June 1997.

45. G.W. Braudaway, "Protecting Publicly-Available Images with an Invisible Image Watermark", *IEEE International Conference on Image Processing*, vol **2**, pp 1021-1025, 1997.

46. E. Koch, J. Zhao, "Towards Robust and Hidden Image Copyright Labeling", *IEEE Workshop on Nonlinear Signal and Image Processing*, Neos Marmaras, Greece, pp 123-132, June 1995.

47. I.J. Cox, J. Kilian, F.T. Leighton, and T.G. Shamoon, "Secure Spread Spectrum Watermarking for Multimedia", *IEEE Transactions on Image Processing*, vol **6**, No 12, pp 1673-1687, 1997.

48. J.J.K.Ó Ruanaidh, W.J.Dowling, F.M. Boland, "Watermarking Digital Images for Copyright Protection", *IEE Proceedings on Signal and Image Processing*, vol **143**, No 4, pp 250-256, August 1996.

49. A.G. Bors, I. Pitas, "Embedding Parametric Digital Signatures in Images", *Proceedings of EUSIPCO*, Trieste, Italy, vol **3**, pp 1701-1704, September 1996.

50. A.G. Bors, I. Pitas, "Image Watermarking Using DCT Domain Constraints", *IEEE International Conference on Image Processing*, Lausanne, Switzerland, vol **3**, pp 201-204, September 1996.

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

52. M.D. Swanson, B. Zhu, A.H. Tewfik, "Transparent Robust Image Watermarking", *IEEE International Conference on Image Processing*, vol **3** pp 211-214, 1996.

53. G.E. Legge, J.M. Foley, "Contrast Masking in Human Vision", *Journal of the Optical Society of America*, vol **70**, No 12, pp 1458-1471, 1980.

54. M.Wu, B. Liu, "Watermarking for Image Authentication", *IEEE International Conference on Image Processing*, Chicago, Illinois, USA, vol. **2**, pp 437 - 441, October 1998.

55. J. Fridrich, "Combining Low-Frequency and Spread Spectrum Watermarking", *SPIE International Symposium on Optical Science, Engineering, and Instrumentation*, San Diego, pp 203-212, July 1998.

56. Deepa Kundur, Dimitrios Hatzinakos, "A Robust Digital Image Watermarking Method Using Wavelet-Based Fusion", *IEEE International Conference on Image Processing, Santa Barbara*, CA, vol **1**, pp 544-547, October 1997.

57. T.A. Wilson, S.K. Rogers, L.R. Myers , "Perceptual- based Hyperspectral Image Fusion Using Multiresolution Analysis", *Optical Engineering*, vol **34**, No 11, pp 3154-3164, November 1995.

58. X.-G. Xia, C.G. Boncelet, G.R Arce, " Wavelet Transform Based Watermark for Digital Images", *Optics Express*, vol **3**, No 12, pp 497-511, December 1998.

59. H.-J.M. Wang, P.-C. Su, C.-C.J. Kuo, "Wavelet Based Digital Image Watermarking", *Optics Express*, vol **3**, No 12, pp 491-496, December 1998.

60. H. Inoue, A. Miyazaki, A. Yamamoto, T. Katsura, "A Digital Watermark Based on the Wavelet Transform and its Robustness on Image Compression", *IEEE International Conference on Image Processing*, Chicago, Illinois, vol **3**, pp 391-395, October 1998.

61. J.M. Shapiro, "Embedded Image Coding Using Zerotrees of Wavelet Coefficients", *IEEE Transactions on Signal Processing*, vol **41**, No 12, pp 3445-3462, 1993.

62. J.J.K.O. Ruanaidh, T. Pun, "Rotation, Scale and Translation Invariant Spread Spectrum Digital Image Watermarking", *Signal Processing*, vol **66** (3), pp. 303-317, 1998.

63. J.J.K.O. Ruanaidh, W.J.Dowling, F.M. Borland, "Phase Watermarking of Digital Images", *IEEE International Conference on Image Processing*, vol **2**, pp 239-242, Lausanne, Switzerland, September 1996.

64. J. Fridrich, "Robust Digital Watermarking based on Key Dependendent Basis Functions", *The Second Information Hiding Workshop*, Portland, Oregon, pp 543-552, April 1998.

65. M. Kutter, "Digital Signature of Color Images using Amplitude Modulation", *SPIE Storage and Retreival for Image and Video Databases*, San Jose, LA, pp 518-526, February 1997.

66. J. Puate, F. Jordan, "Using Fractal Compression to embed a Digital Signature into an Image", *SPIE Photonics East Symposium*, Boston, MA, vol **2365**, pp 23-32, November 1996.

67. G. Voyatzis, I. Pitas, "Chaotic Mixing of Digital Images and Applications to Watermarking", *Proceedings of ECMAST*, Belgium, vol **2**, pp 687-695, May 1996.

68. G. Voyatzis, I. Pitas, "Applications of Toral Automorphisms in image Watermarking", *IEEE International Conference on Image Processing*, vol **2**, pp 237-240, September 1996.

69. M.D. Swanson, B. Zhu, B. Chau, A.H. Tewfik, "Object-Based Transparent Video Watermarking", *IEEE Workshop on Multimedia Signal Processing*, Princeton, NJ, pp 369-376, June 1997.

70. F. Hartung, B. Girod, "Digital Watermarking of Raw and Compressed Video", *Proceedings of Systems for Video Communication*, pp 205-213, October 1996.

71. F. Hartung, B. Girod, "Copyright Protection in Video Delivery Networks by Watermarking of Pre-Compressed Video", *Multimedia Applications, Services and Techniques - ECMAST 97*, Springer Lecture Notes in Computer Science, vol **1242**, pp 423-436, 1997.

72. R.B. Wolfgang, C.I. Podilchuk and E.J. Delp, "The Effect of Matching Watermark and Compression Transforms in Compressed Color Images", *IEEE International Conference on Image Processing*, vol **1**, pp 440-443, Chicago, Illinois, October 1998.

73. J. R. Smith and B. O. Comiskey, "Modulation and Information Hiding in Images", *Workshop on Information Hiding*, University of Cambridge, UK, pp 463-470, May 1996.

74. S.D. Servetto, C.I. Podilchuk, K. Ramachandran, "Capacity Issues in Digital Watermarking", *IEEE International Conference on Image Processing*, vol **1**, pp 445-448, Chicago, Illinois, October 1998.

75. J.R. Hernandez, F.Perez-Gonzalez, J.M. Rodriguez and G. Nieto, "Performance Analysis of a 2-D Multipulse Amplitude Modulation Scheme for Data Hiding and Watermarking of Still Images.", *IEEE Journal on Selected Areas in Communications*, vol **16** (4), pp- 510-524, May 1998.

76. T. M. Cover, J. A. Thomas, *Elements of Information Theory*, Second Edition, John-Wiley and Sons Inc, 1991.

77. A. N. Akansu, R. A. Haddad, *Multiresolution Signal Decomposition: Transforms, Subbands and Wavelets*, Academic Press Inc., 1992.

78. P.P.Vaidyanathan, *Multirate Systems and Filter Banks*, Prentice Hall P T R, 1993.

79. M.Ramkumar, G.V. Anand and A. N. Akansu, "On the Implementation of 2-Band Cyclic Filterbanks", *IEEE International Symposium on Circuits and Systems*, Orlando, Florida, vol **III**, pp 520 - 523, May 1999.

80. B. Chen, G.W. Wornell, "Digital Watermarking and Information Embedding Using Dither Modulation", *IEEE Second Workshop on Multimedia Signal Processing*, Redondo Beach, California, USA, pp 273-278, December 1998.

81. M.Ramkumar A.N. Akansu, A.A Alatan, " A Robust Data Hiding Scheme for Digital Images Using DFT", *IEEE International Conference on Image Processing*, **II**, pp 211-215, October 1999.

82. M.H.M. Costa, "Writing on Dirty Paper", *IEEE Transactions on Information Theory*, **IT-29**, pp 439-441, May 1983.

83. S.B Wicker, *Error Control Systems for Digital Communication and Storage*, Prentice Hall, Englewood Cliffs, NJ, 1995.

84. C.H.Chou,Y.C. Li, "A Perceptually Tuned Subband Image Coder Based on the Measure of Just-Noticeable-Distortion Profile", *IEEE Trans. on Circuits, Systems and Video Technology*, vol **5**, No 6, pp 467-476, December 1995.

85. B. Zhu, M.D. Swanson, A.H. Tewfik, "Transparent Robust Authentication and Distortion Measurement for Images", *IEEE Digital Signal Processing Workshop*, Leon, Norway, pp 45-48, September 1996

86. D.Coltuc, P.Bolon, "Watermarking By Histogram Specification", *SPIE Security and Watermarking of Multimedia Contents*, San Jose, CA, vol **3657**, pp 252-263, January 1999.

87. P.M.J.Rongen, M.J.J.J.B. Maes, C.W.A.M. van Overveld, "Digital Image Watermarking by Salient Point Modification: Practical Results," *SPIE, Security and Watermarking of Multimedia Contents*, San Jose, CA, vol **3657**, pp 273-282, January 1999.

88. A.V. Oppenheim, J.S. Lim,"The Importance of Phase in Signals", *Proceedings of the IEEE*, vol **69**, No 5, pp 529 - 541, May 1981.

89. W.A. Pearlman, R.M. Gray, "Source Coding of the Discrete Fourier Transform", *IEEE Transactions on Information Theory*, vol **IT-24**, pp 683 - 692, November 1978.

90. A.G. Tescher, "The Role of Phase in Adaptive Image Coding", *Tech USCIPI Pub. 510*, Image Processing Institute, University of Southern California, Los Angeles, November 1978.

91. R. B. Wolfgang and E. J. Delp, "Overview of Image Security Techniques with Applications in Multimedia Systems," *SPIE Conference on Multimedia Networks: Security, Displays, Terminals, and Gateways*, vol. 3228, Dallas, Texas, pp. 297-308, November 1997.