UNIVERSITY of CALIFORNIA
Santa Barbara

Multimedia Data Hiding: From Fundamental Issues to Practical Techniques

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

Doctor of Philosophy
in
Electrical and Computer Engineering

by

Kaushal M. Solanki

Committee in charge:

Professor B. S. Manjunath, Co-Chair
Professor Upamanyu Madhow, Co-Chair
Professor Shivkumar Chandrasekaran, Co-Chair
Professor Jerry Gibson

December 2005

The dissertation of Kaushal M. Solanki is approved.

Professor Jerry Gibson

Professor Shivkumar Chandrasekaran, Committee Co-Chair

Professor Upamanyu Madhow, Committee Co-Chair

Professor B. S. Manjunath, Committee Co-Chair

September 2005

Multimedia Data Hiding:

From Fundamental Issues to Practical Techniques

by

Kaushal M. Solanki

To my parents,

for their support and encouragement.

If I had been with them, as I wished, and they deserved,

this thesis would not have been completed.

And to my loving sister, Dhara,

whose energy inspired me countless number of times.

Acknowledgements

I would like to express sincere gratitude to my advisors, Professors Manjunath, Madhow, and Chandrasekaran, for their guidance, encouragement, and support. I am extremely fortunate to have the opportunity to work closely with three outstanding, highly knowledgable, yet very friendly persons. Prof. Manjunath has always been very patient and understanding, and has taken extremely good care of us, the students. I would like to thank him for his insightful comments and suggestions throughout my graduate studies, which have greatly improved the clarity of this thesis and publications. Prof. Madhow has always been available for detailed technical discussions, which has shaped my thinking through the years. He has been instrumental in my development as a researcher; my writing skills have improved manyfold under his tutelage. I would like to thank Prof. Chandrasekaran, among other things, for acting, at times, as a 'devil's advocate', often exposing the loopholes and refining the ideas. I am also thankful to Prof. Gibson for his comments and advices through the past years.

But for the discussions and sessions with my advisors and colleagues during the *data-hiding meetings*, many ideas presented in this thesis would never have taken shape. I would like to thank Kenneth Sullivan, my data hiding colleague, for collaborations, brainstorming sessions, and more importantly, his friendship. I am very fortunate to share my workplace with a knowledgable and very helpful person. It has always been fun traveling with him.

It was pleasure working with Onkar Dabeer, who always had interesting insights to offer. I would like to thank Noah Jacobsen for collaboration as well as many helpful discussions during the early years. Thanks also to Jiyun Byun for

many useful discussions during the time she was in the project. I am thankful to David Wheland, Zhiqiang Bi, and Jiyun for their help, at various stages, in the development of software prototypes for data hiding.

I would like to thank all my current and former colleagues at the vision research laboratory (VRL): Sitaram Bhagavathy, Bariş Sümengen, Dmitry Federov, Marco Zuliani, Jelena Tešić, Motaz El-Saban, Xinding Sun, Nhat Wu, Laura Boucheron, Shawn Newsam, Peng Wu, Ching-Wei Chen, Ohashi Gosuke, and all the past and current visiting researchers. It has been a pleasure to be among very bright and friendly people. Our group lunch has always been something to look forward to. Thanks also to many other fellow graduate students for their help and support: Vinoo Margasahayam, Jayanth Nayak, Gabriel Gomes, and Ashish Aggarwal.

Special thanks to my 'old' friends, Shivprakash Iyer, Anand Nanavati, Vishwa Ved, Sinjeet Parekh, Milind Mistry, Gilroy Menezes, Ojas Gandhi, and Shetal Shah for their lifelong friendship, and the fun we had. You have always been with me in my good and bad times alike. I am very grateful to my Tabla guruji, Pandit Homnath Upadhyayji for being kind, friendly, and extremely patient in teaching me to play *Tabla*, the classical Indian drums. Though it may sound unusual at this point, I would also like to thank all my high-school teachers and classmates at Kendriya Vidyalaya Surat, for creating a friendly atmosphere, and providing a very strong foundation.

I have no words to express gratitude towards my family, without whose support and encouragement, this journey would not even have begun. I especially thank my sister, Dhara, and my cousins, Nitin and Nainesh, for every bit of their help, support, and encouragement. They made sure that I do not have to worry about

anything else while I was working on the thesis, or for an approaching deadline. My parents deserve all the credit for what I have achieved, and what I may, in the future. To them, nothing was more important than my education. They asked for more when I was happy with what I had, and inspired and supported me when I was not. I am grateful to my late grandmother and grandfather for the endless love and blessings.

Last but not least, I would like to thank Office of Naval Research (ONR grant #N00014-01-1-0380 and #N00014-05-1-0816) for supporting the work presented in this dissertation.

Disclaimer

Curriculum Vitæ

Kaushal M. Solanki

September 2005	Doctor of Philosophy Department of Electrical and Computer Engineering University of California, Santa Barbara
December 2001	Master of Science Department of Electrical and Computer Engineering University of California, Santa Barbara
June 2000	Bachelor of Science Department of Electronics Engineering National Institute of Technology, Surat, India

Fields of Study

Information hiding, digital watermarking, steganography, image processing, and digital communication.

Honors and Awards

IBM Student Paper Award of the 2004 IEEE International Conference on Image Processing.

Publications

K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Robust Image-Adaptive Data Hiding using Erasure and Error Correction," In *IEEE Transactions on Image Processing*, vol. 13, no. 12, pp 1627-1639, December 2004.

K. Solanki, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "'Print and Scan' Resilient Data Hiding in Images," Submitted for publication, *IEEE Transactions on Information Forensics and Security*, September 2005.

K. Solanki, K. Sullivan, U. Madhow, B. S. Manjunath and S. Chandrasekaran, "Statistical Restoration for Robust and Secure Steganography," *Proceedings of the IEEE International Conference on Image Processing*, Genoa, Italy, September 2005.

K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Modeling the Print-Scan Process for Resilient

Data Hiding," In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII*, vol. 5681, pp. 418-429, San Jose, CA, USA, January 2005.

K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Estimating and Undoing Rotation for Print-Scan Resilient Data Hiding," In *Proceedings of the IEEE International Conference on Image Processing*, vol. 1, pp. 39-42, Singapore, October 2004.

K. Solanki, O. Dabeer, U. Madhow, B. S. Manjunath, and S. Chandrasekaran, "Robust Image-Adaptive Data Hiding: Modeling, Source Coding, and Channel Coding," Invited paper, In 41^{st} *Annual Allerton Conference on Communications, Control, and Computing*, Monticello, IL, USA, October 2003.

K. Solanki, O. Dabeer, B. S. Manjunath, U. Madhow, and S. Chandrasekaran, "Joint Source-Channel Coding Scheme for Image-in-Image Data Hiding," In *Proceedings of the IEEE International Conference on Image Processing*, vol. 2, pp. II - 743-746, Barcelona, Spain, September 2003.

N. Jacobsen, K. Solanki, U. Madhow, B. S. Manjunath, S. Chandrasekaran, "Image-Adaptive High Volume Data Hiding Based on Scalar Quantization," In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, vol. 1, pp. 411-415, Anaheim, CA, USA, October 2002.

K. Solanki, N. Jacobsen, S. Chandrasekaran, U. Madhow, B. S. Manjunath, "High Volume Data Hiding: Introducing Perceptual Criteria into Quantization-based Embedding," In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, vol. 4, pp. 3485-3488, Orlando, FL, USA, May 2002.

Abstract

Multimedia Data Hiding:

From Fundamental Issues to Practical Techniques

by

Kaushal M. Solanki

The rapid growth in the demand and consumption of the digital multimedia content in the past decade has led to some valid concerns over issues such as content security, authenticity, and digital rights management. Multimedia data hiding, defined as imperceptible embedding of information into a multimedia host, provides potential solutions, but with many technological challenges. In this thesis, we address several fundamental issues in this field, which provide the framework for the design of practical techniques that can seamlessly be deployed in real-world applications.

The first problem we address is that of embedding high volume of information in an image without incurring any perceptual distortion, and achieve robustness against compression, additive noise, and image tampering attacks. Key to this is the use of image-adaptive perceptual criteria, and a coding framework that employs turbo-like codes, leveraging the huge advances in coding theory made over the last decade. Next, a hybrid digital-analog scheme is proposed for hiding an image into another image in such a way that the quality of the recovered image improves as the attack gets milder. This graceful improvement is permitted by a novel joint source-channel coding scheme. We then present techniques that allow

robust embedding of hundreds of bits into images, in a manner that survives printing followed by scanning. Autocalibration methods, such as automatic algorithm for undoing rotation induced by the scanning process, play a key role. Finally, we present a framework for the design of perfectly secure covert communication (steganographic) techniques that can potentially evade any statistical steganalysis of the stego signal.

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Ever improving network bandwidths, computer speeds, digital storage capacities, and wireless capabilities are changing our lives right from the way we entertain ourselves, communicate with each other, or assimilate and disseminate knowledge, to the way we operate our bank accounts. A key driver for these changes has been the rapid growth in the demand and consumption of digital multimedia content. This has, however, lead to some valid concerns over multimedia content security, authenticity, and intellectual property rights. There is an urgent need to address these issues, failing which, the true potential of recent as well as future technological advances (in this area) may not be realized.

Multimedia data hiding, defined as imperceptible embedding of information into a multimedia host, provides potential solutions, though with many unseen challenges. Because of its potential applications in multimedia content security, data hiding continues to receive considerable attention from the research community. Multimedia data hiding offers unique challenges that require integration of

various disciplines, such as image processing, computer vision, information theory, signal compression, error correction coding, and communication theory. In this dissertation, we address several fundamental issues in this field, which provide the framework for the design of practical techniques that can seamlessly be deployed in real-world systems. Through a mix of experimental and analytical approach, we are able to provide practical solutions to several problems important to the research community. The work presented in this dissertation is mainly focussed on embedding information into images, however, several of the proposed approaches and analyses are general, and can be easily applied for other media data, such as audio and video.

Data hiding can be defined more formally as the process by which a message signal or *signature* is imperceptibly embedded into a *host* or *cover* to get a *composite* signal. The general framework of a data hiding system is shown in Figure 1.1. There are three main conflicting requirements of a multimedia data hiding system: perceptual transparency, robustness, and capacity. Information embedding into a multimedia host should not incur any perceptual distortion to the host, i.e., the composite signal should be perceptually *transparent*. The data should be recoverable even after the composite multimedia signal has undergone a variety of processing, intentional or unintentional, to remove the embedded data. In other words, the hidden data must be *robust* against a variety of *attacks*. We would also like to embed as many bits into the host as possible, or, the *capacity* of the embedding system should be high. Different applications have different specific requirements of robustness and the volume of embedding. Most applications, however, require near-perfect perceptual transparency. There are several

other design issues, based, again, on the target applications, which are considered or defined in this thesis. This include maintaining *statistical* transparency to conceal the presence of embedded data, or providing *graceful improvement* in the quality of recovered signature data as the attack strength reduces. We shall elaborate these issues later in this chapter.

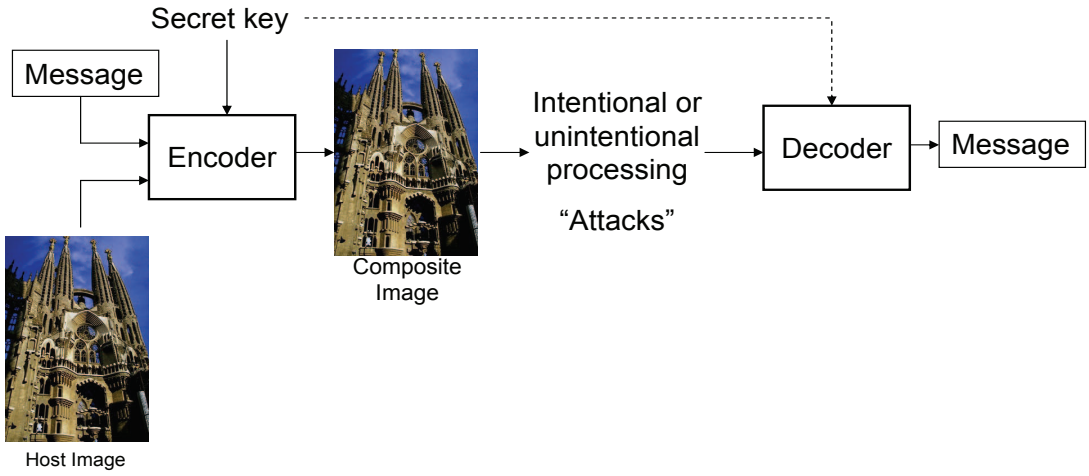


Figure 1.1: General framework of a data-hiding system.

The first problem we consider is that of embedding high volume of information into images, which could survive attacks such as compression and additive noise. A significant issue here is to embed large number of bits without causing perceptual degradation to the host image. This requires embedding data in a way that adapts with the local characteristics of an image. Key to this is a coding framework that employs turbo-like codes, leveraging the huge advances in communication/coding theory made over the last decade.

Next, a hybrid digital-analog scheme is proposed for hiding an image into another image in such a way that the quality of the recovered image improves as

the attack gets milder. This *graceful improvement* is permitted by a novel joint source-channel coding scheme. To the best of our knowledge, this is the first joint source-channel coding approach for data hiding proposed in the literature.

We present methods to hide data into images that achieves resilience to printing and scanning process. The design of these techniques is based on extensive analytical as well as experimental modeling of the print-scan process. The embedding rates we report provide significant improvement over the state of the art.

A framework, termed *statistical restoration*, for the design of techniques for secret communication is proposed next, which can potentially evade any statistical detection of the presence of hidden data. Using the techniques based on the framework, several thousand bits[1] can be hidden into images without modifying the relevant statistics of the cover image, so that the presence of embedded information cannot be detected by statistical analysis.

1.1 Motivation

This dissertation is motivated by several emerging applications of multimedia data hiding. The advent of digital age with the Internet revolution has empowered consumers with capabilities and luxuries that were unthinkable just a decade ago. However, the availability of inexpensive hardware (such as printers, scanners, and compact disc and digital versatile disc burners), and powerful software (such as image, video, and audio editing and processing software) have made it

[1]For example, 30 000 bits can be hidden into 512×512 images while maintaining complete statistical transparency.

very easy for users to make illegal copies of copyrighted material, and share it with other Internet users through one of several available peer-to-peer file-sharing utilities (such as KaZaA, BitTorrent, and eDonkey2000). Now, users can easily *photoshop* digital images, or edit audio or video clips. The advent of digital age has, ironically, destroyed the authenticity of digital multimedia information.

To counter this, *digital watermarking* is a technology being developed, in which, copyright information is embedded into the host in a way that is robust to a variety of processing intended to remove the watermark. In multimedia authentication applications, the embedded digital watermark must detect malicious tampering, but should not get destroyed by 'benign' attacks, such as compression and enhancement. We present our approach for image tamper detection and localization in Chapter 3.

In copyright protection applications, the embedded digital watermark must survive extreme malicious processing of the image. Several freeware packages are available that attack the images without inducing perceptual distortion (e.g., *Stirmark* [85], and *Checkmark* [84]). The ease with which images can be converted from the print to the digital form and vice versa makes it necessary that the embedded digital watermark is resilient to the print-and-scan operation. In Chapter 5, we study data hiding methods that are resilient to the print-scan operation as well as the attacks included in the Stirmark package.

Security concerns have grown tremendously in past few years all over the world. The main concern for government agencies is to catch the malicious elements, but at the same time, provide hassle-free movement for law-abiding citizens. This calls for developing strong deterrents against forgery of important

documents such as passports, driving licences, and ID cards. Here too, print-scan resilient data hiding provides a potential solution: security information (such as fingerprints, signature, or passport number) can be imperceptibly embedded into a picture in the document. Only specific devices, which have access to a secret key, can decode and authenticate the hidden information. Forgery of such documents become extremely difficult because the embedded data is inseparable from the picture.

With the technological advances made in telecommunications and networking, the world is connected today. So are the terrorists. It is easier than ever before for them to plan large-scale destructions because they can communicate anonymously across the globe without inciting anyone's suspicion. Government agencies, such as the central intelligence agency (CIA), are concerned that the terrorists might be communicating secretly by embedding information in images or video and passing them around through the World Wide Web (for example, see an article that appeared in the popular press [55]). An application of data hiding is *steganography*, the art and science of communicating in such a way that the very existence of communication is not known to the third party. It is very important to investigate *steganalysis*, the study of techniques to detect the presence of hidden data. Also significant is to understand the limits of steganography, and analyze how much information can be embedded into images, audio, or video hosts without being detected. In this context, we present, in Chapter 6, a framework to design steganographic techniques that hide significant volume of information, yet, evade most steganalysis techniques available in the literature.

The consumer electronics and computer industry is advancing rapidly with

the products gaining performance à la the famous Moore's Law [96]. New functionalities are being added everyday and the older devices are getting outdated quickly. For multimedia-related devices such as satellite television receivers, it is not realistic to ask consumers to buy new receivers frequently. In such cases, it is desirable to be compatible with the older devices, and provide new facilities to those receivers that have the advanced features. This *seamless upgrade* of multimedia can be provided by embedding additional control information imperceptibly into the video or audio, which can be interpreted by those receivers that have the know-how. The older receivers would continue to decode the stream in the usual fashion and would not be affected. A system like this would require embedding significant amount of data and must also be be robust to compression and additive noise. In Chapter 3, we study techniques that fulfil these requirements.

A lot of images are being created in a variety of disciplines, such as biology, geography, medicine, and geology. For these images to be useful, some extra information detailing the context is required, for every image. For example, a biologist studying retinal images would want to know when the image was created, what microscope was used, what colorant was used, and so on. Presently, this *meta data* is either stored in a huge database (e.g., the biological image database at UCSB [2]), or is stored in the headers of specialized formats specific to the particular field. For some applications, creating a database might be an overkill. On the other hand, using specialized formats to put the meta data in the headers takes away the flexibility and limits the portability. The meta data would be lost if the images are converted from its original proprietary format to any other (compressed or non-compressed) format. Also, specialized viewing programs might be

needed to interpret the formats. Having the flexibility of changing the storage format or allowing compression is especially significant now, as the researchers in these fields are collaborating with those in image processing and computer science in order to understand, interpret or process these images efficiently. Image data hiding can provide a way to get around this problem: the meta-data can be embedded into the images without distorting the images. This way, no specialized formats are required, and the meta-data stays with the image even when it changes the storage format or if it is compressed. Note that, in these applications, it is very important to preserve the perceptual quality of the images while embedding the significant number of bits. The techniques presented in Chapter 3 can be employed these applications too, such as for the annotation of medical, biological, geo-spatial, or cartographic images. We now summarize the main contributions of this thesis.

1.2 Summary of Contributions

We address several important problems in data hiding, add new requirements, and propose practical schemes that meet many stringent design requirements. Below is a list, with brief description, of the fundamental contributions of this thesis, which, we believe, have applicabilities beyond the schemes presented in this thesis.

1. **A coding framework for adaptive hiding:** A flexible coding framework is presented, which allows the encoder to select hiding locations dynamically without needing to send the side information about hiding locations to the

decoder. The framework is applied in two embedding schemes presented in this thesis: high volume hiding using perceptual criteria, and print-scan resilient embedding.

2. **A joint source-channel coding method to hide analog information:** A method to embed *analog* information into general host samples is proposed. We show that the mean squared distortion in the recovered data reduces as the attack gets milder. This method is used in our image-in-image hiding scheme that uses hybrid digital-analog embedding scheme to achieve graceful improvement in the received image quality.

3. **Data hiding resilient to printing followed by scanning:** The print and scan process has been systematically characterized, and the main sources of distortion during the print-scan process have been identified. These findings, based on detailed analysis of some components, are used to design practical print-scan resilient hiding schemes. We can further improve upon these techniques by studying the other components that have not yet been studied in detail.

4. **A framework for design of statistics-preserving data hiding schemes:** A statistical restoration framework is proposed that allows design of schemes that can embed data into a host without changing its relevant statistics. We show that this framework can be used to design steganographic techniques that can evade the best image steganalysis tool out there. This framework, however, is general and can be applied to any host data, not just images, and can also be employed to restore any particular statistics.

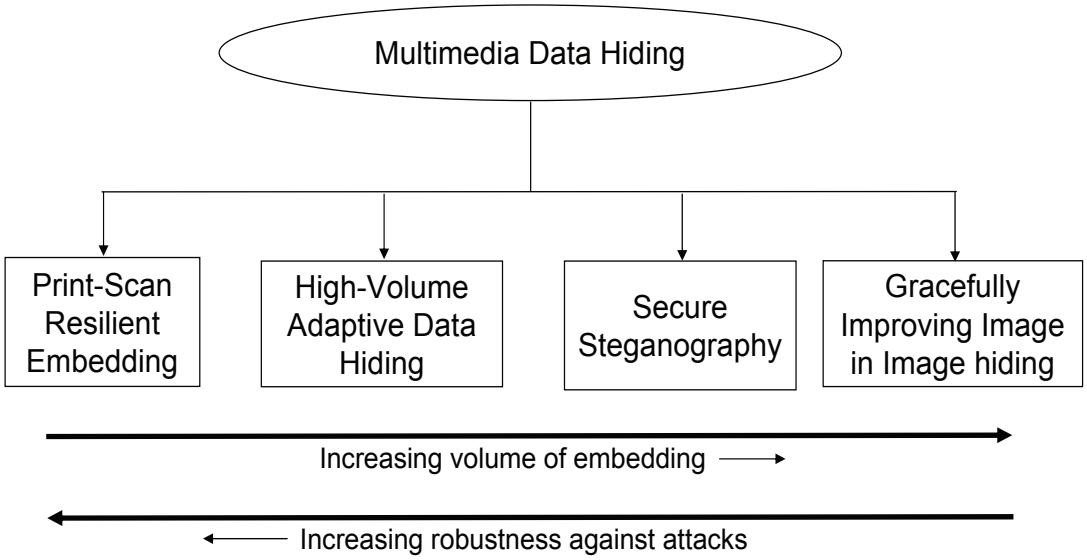


Figure 1.2: The main contributions of the thesis, sorted (roughly) according to the capacity and robustness.

We propose several practical techniques that are based on the above fundamental contributions. Figure 1.2 shows the various parts of the thesis, and where they fit in terms of the volume and robustness requirements. In the following, we study each of them separately.

1.2.1 Image-Adaptive High-Volume Data Hiding

Embedding high volume of information into images without causing perceptual distortion has been quite challenging. The earliest approaches were to simply modify the least significant bits (LSB) of the image samples to hide the data (see [53], Chapter 2). However, embedding in LSBs is very fragile, since the hidden data will be lost by simple modifications of the image, such as compression. Spread spectrum (SS) techniques were proposed to counter this problem [26].

Here, a spread version of the data is added to the image either in spatial or transform domain. Perceptual transparency is achieved in these techniques by an approach called *perceptual shaping*, in which the added spread sequence is scaled by a parameter determined by the perceptual sensitivity of the region. While spread-spectrum methods do provide robustness against attacks such as compression and noise, it is difficult to embed higher volume of information in images using these techniques.

A class of data hiding methods, called quantization index modulation (QIM), based on quantization of the host samples were proposed and shown to be superior to spread-spectrum techniques [19, 18, 21, 20]. Using a simplified version of QIM, called the scalar QIM, data can be hidden such that it can survive attacks like compression and additive noise. However, when hiding large volume of data, we must adjust to local characteristics within an image in order to control perceptual distortion. For QIM hiding, adaptation cannot be done as simply as SS that employs perceptual shaping.

We show, in Chapter 3, that high volume of information can be hidden in images by using dynamically selected discrete cosine transform (DCT) coefficients for embedding (also see [109, 51, 108]). The use of local criteria to choose where to hide data can potentially cause desynchronization of the encoder and decoder. This synchronization problem is solved by the use of powerful, but simple to implement, erasures and errors correcting codes, which also provide robustness against a variety of attacks.

The problem of adaptive hiding has been addressed by prior researchers with varying degree of success. Wu et al [136, 137] propose an adaptive embedding

method, termed *uneven* hiding. This system either uses a fixed embedding rate through an approach called shuffling, or explicitly sends the side information about hiding locations in a variable rate embedding approach. Apart from a complicated implementation, the volume of data hidden using this approach is quite less. More recently, Fridrich et al [41] propose an interesting approach, called *wet paper codes*, which allows the encoder to choose the embedding locations without needing to send any side information to the decoder. This approach, however, is primarily geared towards applications in steganography, and is fragile against any attacks or modifications to the image. On the other hand, the coding framework proposed in this thesis (also published in [109] and [51]) not only does not require any side information to be sent, but it also allows information to be recovered against a number of attacks such as compression, additive noise, resizing, or tampering.

The framework can also be employed to design a system for multimedia authentication. With appropriate design, one can detect malicious tampering of the image at the decoder and also localize the tampered area. An example is presented in Figure 1.3, in which we embed 6912 bits into a 512×512 Lenna image. Even after the tampering of the image as shown in the figure, all the hidden bits are received successfully. This system, described in Chapter 3, can distinguish between the malicious tampering of the image and benign processing such as compression.

The effectiveness of the system is demonstrated by an online system available at [1]. The interface allows the user to upload an image, provide a text message which is to be hidden in the image, and also give a secret passcode. An option to

Figure 1.3: High-volume data hiding with robustness against malicious tampering. All the embedded 6912 bits are recovered successfully at the decoder in spite of the attack.

choose the desired volume of embedding is provided that determines the amount of robustness. If low volume of data is embedded, the composite image will have higher robustness, and vice versa. The data can be recovered after the hidden image has undergone several attacks such as compression, additive noise, or tampering, as stated before.

1.2.2 Gracefully Improving Image-In-Image Hiding

We here consider the problem of image-in-image hiding, in which an image, called the *signature* image, is to be embedded into another image, called the *host*

image, to get a *composite* image. The high volume embedding method described in the previous section can be used to hide an image into another image. However, the system must be is designed for the worst anticipated attack. In practice, the attack level is seldom known apriori, and if the actual attack is less severe than the design attack, we are still stuck with the design signature image quality. Ideally, we would like an image-in-image hiding scheme that results in graceful improvement in the image quality with less severe attacks. Such schemes require *joint* source-channel coding, which has been studied for the Gaussian channel (see, for example, [17, 103]).

To the best of our knowledge, such schemes have not been studied for the data hiding channel[2]. An important contribution of this thesis is the development of joint source-channel coding techniques for data hiding. In Chapter 4, we present a hybrid digital-analog (joint source-channel) coding scheme for image-in-image hiding (also published in [107]). It leverages the digital scheme (described in previous section) based on image-adaptive criteria and turbo-like codes (Chapter 3, and [109, 51]), and involves the transmission of the analog residue using a new method.

Focussing on JPEG compression attacks, we derive the minimum mean squared error (MMSE) decoding strategy for the proposed hybrid embedding scheme. We demonstrate a practical image-in-image hiding system that can hide signature images as big as 256×256 into a 512×512 host image, in such a way that there is perceptual as well as mean-squared error improvement in the recovered image quality as the attack gets milder.

[2]A somewhat-related approach is discussed in [136, 137], in which a multi-level embedding is considered. We discuss this work in Chapter 4.

1.2.3 Print-Scan Resilient Hiding

In Chapter 5, we consider the problem of hiding information into an image in such a way that the embedded data can be recovered even after it is printed and scanned. There has been a growing interest among researchers in the area of print-scan resilient embedding, but little progress has been made because of the complex nature of the problem. One of the first approaches was by Lin and Chang [61], who model the print-scan process by considering the pixel value and geometric distortions separately. There are some watermarking methods [93, 105, 10] that were not specifically designed for the print-scan attack, but they do report robustness against the print-scan operation under specified experimental setup.

Most of the above methods embed only a single bit (or a few bits) of information, as they assume the availability of the watermark sequence at the decoder. In Chapter 5 of this thesis (also see [112, 110, 111, 106]), we propose methods to hide information into images that achieves robustness against printing and scanning. Using these techniques, several hundred information bits can be embedded into images with perfect recovery after the print-scan operation, which is a significant improvement over the state of the art. An example is presented in Figure 1.4, in which we embed several bytes of information using a technique proposed in the chapter, and successfully recover the embedded data after the print-scan operation.

An important contribution of this work is a systematic analytical modeling of the print-scan process by breaking it down into simpler sub-processes, which is appropriately complemented by extensive practical experiments. The analytical

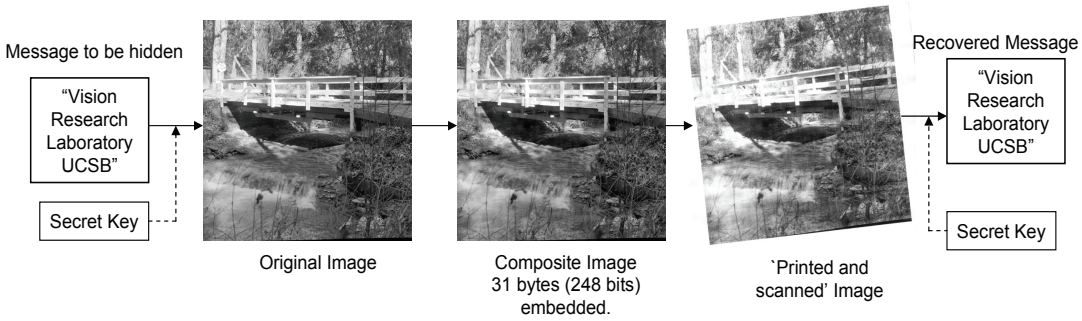


Figure 1.4: An example of print-scan resilient data hiding presented in the chapter. The number of bits that can be embedded in a typical 512×512 image varies from 200 to 500 bits depending on the detail and texture content in the image.

and experimental findings form the basis of the proposed embedding schemes, in which data is hidden in dynamically chosen transform coefficients, with synchronization and error correction using powerful turbo-like channel codes. This also provides robustness to the hidden data against several other attacks included in *Stirmark*, such as Gaussian or median filtering, scaling or aspect ratio change, heavy JPEG compression, and rows and/or columns removal.

Also proposed is a novel approach for estimating the rotation that an image might undergo during the scanning process, by exploiting knowledge of the digital halftoning scheme employed by the printer. The employed derotation method is completely different from the previously used approaches, in which rotation invariance is typically achieved by using FM transform [61, 93]. The advantage of the proposed technique for print-scan resilient hiding is that there is no penalty in hiding rate for achieving robustness against rotation.

1.2.4 A Framework for Secure Steganography

In Chapter 6, we propose a framework that allows design of embedding schemes that can evade statistical steganalysis while hiding at high rates, and also achieve robustness against attacks. We are motivated by the notion of ϵ-secure steganography proposed by Cachin [12], in which the relative entropy (also called Kullback-Leibler or K-L divergence) between the cover and stego distributions is less than or equal to ϵ. Our approach for achieving a small ϵ is to employ *statistical restoration*, wherein a portion of the data-hider's "distortion budget" is spent in repairing the damage done to the image statistics by the embedding process.

Modern steganography is a game with escalating sophistication between the hider and the steganalyst. One of the first popular steganalysis tools was *Stegdetect* [90], which uses a chi-square statistic on the histogram of transform coefficients to detect least significant bit (LSB) hiding. Stegdetect can be improved upon by more sophisticated detection-theoretic approaches [29]. Such methods, which are based on the histogram of the host coefficients, have spurred the development of hiding techniques that make as little change to the histogram as possible. Provos' Outguess algorithm [89] was an early attempt at histogram compensation for LSB hiding, while Eggers et al [32] suggest a more rigorous approach to the same end, using histogram-preserving data-mapping (HPDM). In turn, steganalysis tools that counter such histogram-preserving hiding methods have been developed, such as detection, for image-based hiding, of block-DCT embedding by evaluation of the increase in blockiness due to hiding [39, 128]. While both HPDM and OutGuess attempt to match the quantized histogram of the discrete cosine transform (DCT) coefficients, more recent proposals [48, 129]

try to match the continuous marginal statistics.

Unlike most of the steganography approaches in the literature, our framework allows design of schemes that can have perfect security by achieving zero Kullback-Leibler (K-L) divergence between the cover and the stego signals. One can match continuous statistics using the proposed approach, not just discrete (or quantized) statistics. Only a couple of prior schemes, to the best of our knowledge, can potentially achieve zero KL divergence for continuous host statistics: Gullion et al [48], and Wang and Moulin [129, 75]. Both the approaches, however, have some serious issues that limit their practical applicability. Guillon et al [48] suggest transforming the source to get a uniform PMF source. The message is hidden in this with the quantization hiding scheme, which is known not to change the PMF of uniform sources. Therefore, the PMF after transforming back is also the same as the original. This method, however, is not likely to be robust, and also, there is no way to control the distortion induced by the embedding process. Wang and Moulin [129] propose a reduced rate variant of standard QIM, called the stochastic QIM, which can be made to have zero K-L divergence. However, because of the stochastic nature of the hiding process, the method is likely to yield high error rates when embedding large volumes of data. Note that in [75], the proposed stochastic QIM technique embeds only one bit of information.

The proposed framework allows design of robust techniques that are not fragile against attacks, unlike most of the methods proposed in the literature so far. While certainly not the most important issue for steganographic systems, robustness against "natural" attacks such as compression or additive noise is highly desirable. Most of the prior schemes, such as OutGuess [89], HPDM [32], Sallee's

model based methods [94, 95], and Fridrich et al's perturbed quantization [40], are fragile against any modifications to the image.

The techniques do not rely on accurate modeling of the host statistics. This is unlike Sallee's model-based steganography [94, 95], in which the hider ensures that the stego signal conforms to a given model. In the absence of a perfect model for the host, nothing stops the steganalyzer from selecting a *better* model by spending more computational power, and hence detect the embedded data. This is indeed practically shown in [11], where Sallee's Cauchy-model based JPEG steganography is broken by using only the first order statistics. Our approach is very difficult to detect in this manner, since the stego marginals are simply restored to conform to the host's empirical density, rather than invoking a statistical model for the host's marginals.

The framework can be employed for restoring statistics of any order, not just the first-order statistics. Most of the histogram preserving techniques can be detected by steganalysis approaches that use cover memory, such as Fridrich et al [39], and Wang and Moulin [128], who detect block-DCT embedding by modeling the increase in *blockiness* of the image due to the hiding in DCT coefficients. Our framework can be employed to design methods that can restore such statistics as well.

1.3 Organization of the Dissertation

The rest of this thesis is organized as follows. We start with an overview of data hiding field along with a survey on prior approaches in Chapter 2. Here we

discuss information-theoretic analyses, robust watermarking techniques, image-adaptive techniques, as well as approaches for steganography and steganalysis. Having provided the context, in Chapter 3, we move on to our image-adaptive embedding schemes that allows us to embed high volume of information without causing perceptual degradation, and also be robust to attacks such as compression, additive noise, and image tampering. We add a new design requirement for data hiding systems in Chapter 4: along with robustness and perceptual transparency, we would like to recover the signature data with high fidelity if the attack strength is small. To achieve this goal, we propose a hybrid digital-analog joint source-channel coding scheme. An image-in-image hiding system is demonstrated, which achieves perceptual as well as mean-squared error improvement in the recovered image quality as the attack gets milder. In Chapter 5, we address the problem of embedding information robust to the printing followed by scanning operation. Extensive experimental modeling is taken up to learn the channel characteristics, which leads to a couple of image-adaptive embedding schemes. We then move on, in Chapter 6, to the problem of hiding large volume of data without changing the statistical properties of the host data so as to communicate without inciting anyone's suspicion. A framework, called statistical restoration, is proposed to this end, which allows the design of such embedding schemes, providing several advantages over the current state-of-the-art techniques. Finally, in Chapter 7, we present the concluding remarks and discuss some interesting avenues for future work.

Chapter 2

Data Hiding: Overview and Prior Art

Secure transmission of information has always been very important to mankind. There is some historical evidence that covert communication is as old as the civilization itself. Secret writing has been traced back to ancient China, India, and Greece. Interesting discussions on the history of data hiding can be found in [117, 86]. Ancient Chinese rulers were known to communicate secretly by writing the messages in thin sheets of silk or paper, and making them into small balls, which are then swallowed by the messengers. Several ancient Indian texts (for example, Kautilya's *Artha-śāstra*, which dates back to 321-300 B.C., and Vātsāyana's *Kāmasūtra*) discuss the art of covert communication in detail, with explicit formulas for secret writing. Trithemius, in 1500 A.D., defined the term *steganography* (as *secret writing*) in his book *Steganographia* [119]. There are several other examples of secret communication in history, such as the use of

invisible inks, or writing a message on a shaved head and then growing the hairs.

Digital data hiding, however, is a relatively young field with a majority of publications coming up in less than a decade. The potential for solving some important problems like content authentication, and digital rights management, and several emerging applications such as seamless upgrade of multimedia, and annotation of images, have made sure that the interest in this field keeps growing. This is evident from the fact that a new journal (*IEEE Transactions on Information Forensics and Security*) has been started for research publications in data hiding, digital watermarking, information security, biometrics, and forensics. A series of *Supplements on Secure Media* for the *IEEE Transactions on Signal Processing* have appeared recently (October 2004 and February 2005).

Along with the excitement among researchers about the potential applications of data hiding and digital watermarking, there have also been some counter-views on whether or not can data hiding solve the problems in digital rights management. Interesting discussions by the intelligentsia of the field on these issues can be found in the literature (see Herley's article on "Why watermarking is nonsense" [50], and Moulin's comments on this article [74]).

In a relatively short span, notable progress has been made both in the theoretical and the practical aspects of the information hiding problem. Several books are available now that provide a comprehensive treatment of the well-established concepts: Johnson, Duric and Jajodia [53], Cox, Miller and Bloom [28], Eggers and Girod [34], and Barni and Bartolini [9]. A recent tutorial paper by Moulin and Koetter [76] provides an excellent overview of the field, with a focus on the core mathematical concepts. Some earlier survey papers in this area include

[117, 86, 135].

With many books and good survey papers available, our treatment, in this chapter, of the overview of data hiding will be brief. Readers are referred to the above references for a more comprehensive study. In the short survey presented here, we focus mainly on the techniques that are closely related to the ones presented in this thesis.

The rest of the chapter is organized as follows. We introduce the data hiding problem as communications with side information at the encoder in Section 2.1. This is followed by a discussion on the design issues (Section 2.2). The basic embedding methods: least significant bit (LSB), spread spectrum (SS), and quantization index modulation (QIM) are briefly described in Section 2.3, followed by an overview of the information-theoretic and game-theoretic results (Section 2.4). Next, in Section 2.5, we discuss several robust techniques popular in the literature. Image-adaptive techniques are studied next (Section 2.6). Finally, we survey the steganography and steganalysis literature in Section 2.7, followed by a brief summary of the chapter in Section 2.8.

2.1 The Data Hiding Problem

It was recognized quite early that data hiding can be modeled as a communications problem [26, 27], with the adversary's "attack" being the *de facto* channel. In *non-blind* data hiding systems, it is assumed that the original host or cover is available at the decoder. In this case, the problem reduces to the classical communications (or data transmission) problem, in which a message has to be

transmitted to a receiver in the presence of *noise* due to the attacks. For *blind* information hiding systems, the decoder does not have access to the original host signal, so the host itself acts as a noise in the system. However, viewing the host signal as noise disregards the fact that it is actually known to the encoder, who can use this extra information to its advantage. By considering the knowledge of the host signal at the encoder, the information hiding problem can be modeled as communications with *side information* about the channel-state at the encoder.

Figure 2.1 shows a typical data hiding scenario: we want to embed a message m into a host signal $\mathbf{x} \in \Re^N$ to get the composite signal $\mathbf{s} \in \Re^N$. The received signal $\mathbf{y}$ is corrupted by noise $\mathbf{n}$ due to attacks, from which the decoder estimates the message $\hat{m}$ that was hidden.

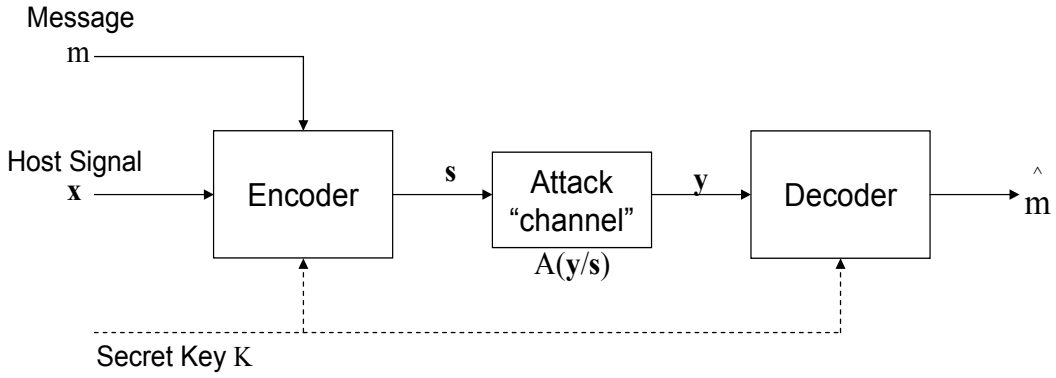


Figure 2.1: A typical data hiding scenario.

A natural requirement of a data hiding system is that there should not be any perceptual distortion during the embedding process. This is modeled by a constraint on the amount of change that is made to the host signal. Furthermore, the attacker is also limited by the amount of distortion he or she can induce

to the signal, because the usability of the attacked composite signal has to be maintained. The distortion constraints can be complex functions motivated by the human visual system. For simplicity of analysis, an average mean squared distortion is quite commonly used.

$$D(\mathbf{s}, \mathbf{x}) = \frac{1}{N} \sum_{n=1}^{N} (s_n - x_n)^2$$

As said before, the data hider is allowed to induce the distortion of at most D_1, i.e., $D(\mathbf{s}, \mathbf{x}) \leq D_1$, and the attacker can induce a maximum distortion of D_2, i.e., $D(\mathbf{x}, \mathbf{y}) \leq D_2$. The constraint at the encoder is similar to power constraint in the classical communication setting. Likewise, the attack distortion constraint is equivalent to the noise power. Several authors have analyzed the problem from an information-theoretical point of view, which we described in Section 2.4. Let us now move on to the issues involved in the design of various data hiding systems.

2.2 Design Issues

There are several issues, or requirements, that are involved in the design of data hiding systems. Figure 2.2 illustrates various requirements of a data hiding system. This figure is an update of Figure 1.1 presented in the previous chapter. In the following, we briefly describe the important technical considerations that come up regardless of the particular application.

- **Perceptual transparency**. Almost all application require that the distortion induced to the host signal is not perceptible, or in other words the composite signal is perceptually *transparent*.

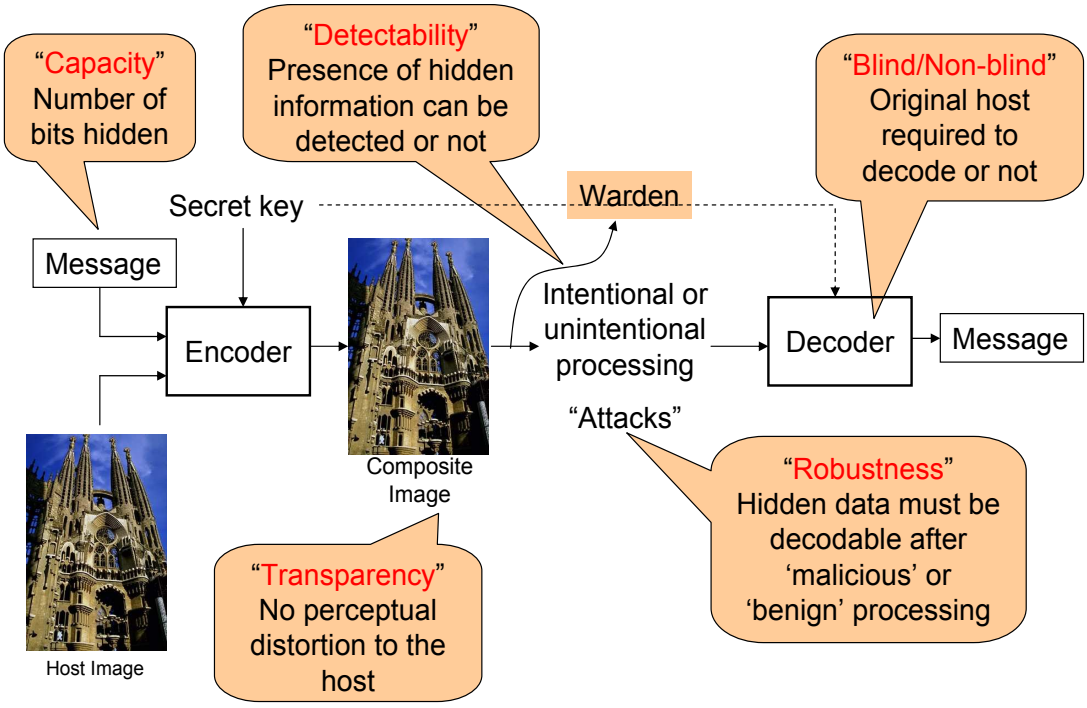


Figure 2.2: General requirements of data hiding systems, which come up irrespective of the particular application.

- **Robustness**. The composite signal must survive several intentional or unintentional *attacks* that might remove the embedded data. The attacks that a system must survive can be 'benign' operations such as compression, additive noise, and filtering, or they can be malicious attacks such as geometric transformations, and printing followed by scanning. The exact robustness requirements of a system would obviously depend on the particular application.

- **Capacity**. This refers to the number of bits that can be embedded by a given system while satisfying the other design constraints (such as robustness). We would like to embed as many bits as possible while satisfying these other constraints.

- **Statistical transparency**. This refers to the change that occurs in the statistics of the host signal during the embedding process. Embedding in such a way that there is only a minimal change to the statistics of the host signal (*statistical transparency*), is required when the data-hiding system is employed for secret communication, in which the existence of communication is not to be revealed. Applications other than steganography may not require that the presence of embedded data is kept a secret.

- **Graceful improvement**. This refers to the improvement in the fidelity of the received signature signal at the decoder as the attack strength reduces. Since the attack strength is seldom known apriori, graceful improvement is highly desirable for *media* signature signals.

- **Computational complexity**. In many applications, it is important to have fast encoding or decoding or both. Low computational complexity may be required for applications where data must be embedded or recovered in real-time (such as having a watermark encoder and/or reader in a video or still-picture camera). Fast decoding is also desirable in systems that spider the Web for copyrighted images with inserted watermark.

Of the above list of design issues, only perceptual transparency is required in almost all applications. Most applications have different specific requirements for the other factors. For example, watermarking systems designed for copyright protection require that the system is robust to a variety of intentional and unintentional processing. However, the payload, or the number of bits hidden, need not be too high. The complexity of decoder must be low if images are to be crawled from the Internet. Other design issues, such as statistical transparency and graceful improvement are not generally important. Note that the design of robust techniques have received the most attention in the literature.

Steganographic systems, on the other hand, require statistical transparency, but they need not be robust against intentional attacks. The volume of embedding should be as high as possible. Computational complexity, unless prohibitive, is not a significant issue, and graceful improvement is desirable if the signature is a media data.

For systems that target applications such as seamless upgrade, or annotation of multimedia databases, the requirements for robustness are moderate (need to survive compression and/or additive noise). The the volume of embedding should not be too low, but again, it need not be very high. The complexity of systems

providing seamless upgrade should be low since it must work real time.

As seen in above paragraphs, the requirements vary from application to application, and hence, a good data hiding scheme would be one which provides trade-offs between the design issues. Let us now describe the basic embedding methods that are popular in the literature.

2.3 Embedding Methods

A natural way to embed information into a media host without inducing any perceptual distortion is to modify the least significant bit of the media samples. The method, accordingly called least significant bit or LSB hiding, is one of the first methods proposed for data embedding. This scheme has been applied for both the classical applications of data hiding: digital watermarking [124], and covert communication [53]. Though the method is quite simple to encode and decode, it has severe limitations for both the applications. Any processing of the image, e.g. compression, will change the LSBs and hence, render the hidden data undecodable. The data hidden in LSBs of images or other media can be easily detected using simple statistical analysis. In spite of these limitations, LSB hiding is quite popular even today, with a number of freeware and shareware packages based on LSB embedding available online (check [3]).

Spread-spectrum (SS) hiding was introduced by Cox et al [26] to alleviate the problems of LSB hiding against attacks. The method, derived from its communications counterpart, involves adding a spread sequence to the image. The spread sequence is constructed from the message to be hidden. The method and

its variations (e.g., [93, 65]), proposed for watermarking applications, are robust against many attacks such as compression, noise addition, and signal processing operations. Spread-spectrum hiding has been used for steganography [66] as well. However, in general, it is well-known that the embedding capacity of SS techniques is quite low[1], especially for blind implementations. This is because the additive methods do not utilize the fact that the host is known to the encoder. Thus the host itself ends up as being the noise, or interference, in the system.

Looking at information hiding in multimedia hosts as communication with encoder side information, new embedding methods have been proposed that reject the host signal interference. These methods are based on Costa's work on writing on dirty paper [24]. In Costa's setting, there is a Gaussian side information known only to the encoder, but not to the decoder (i.e., a paper with Gaussian *dirt*). There is a Gaussian noise which gets added to the *paper* before it reaches the decoder. Costa showed that there is no loss of embedding rate due to the presence of encoder side information. Based on this work, a new class of embedding schemes, called quantization index modulation (QIM), was proposed by Chen and Wornell [19, 18]. The data is hidden by the choice of quantizer (based on the message to be hidden) at the encoder. The decoder just determines which of the possible quantizers were used.

While the methods proposed by Chen and Wornell are based on vector quantizers, simplified version of the schemes employing scalar quantizers have been proposed and applied to multimedia hosts [109, 33]. In [109], it is shown that there is roughly only a 2 dB penalty in terms of resilience to attacks for using

[1]Recently, a high-capacity SS embedding scheme has been demonstrated in [71]. It should, however, be noted that the embedding capacities we consider in Chapter 3 are much higher.

scalar quantization as compared to vector quantization. When there are only two possible symbols in the message set, the method reduces to the well known odd-even embedding, where the odd reconstruction points represent, say a '1' being embedded, and even reconstruction points represent a '0'. More generally, in the information-theoretic terminology, the QIM methods are also known as *binning schemes* [76]. The binning schemes have been proved to achieve very good performance as compared other embedding methods such as SS or LSB [76].

It should be noted that any of the above techniques, LSB, SS, or QIM, can be employed on the pixels of the image (i.e., the spatial domain), or the transform coefficients. Because of their compatibility with the joint pictures expert group (JPEG) [127] and JPEG 2000 compression standards (see [4] for an implementation), discrete cosine transform (DCT) and discrete wavelet transform (DWT) remain the most popular transforms used for data embedding. Discrete Fourier transform (DFT) is also employed because of its properties (e.g., DFT magnitudes are invariant to translation). For LSB embedding in transform domain, such as DCT or DWT, the coefficients must have been already quantized (i.e., compressed).

In the following section, we turn our attention to the information-theoretic analysis of the information hiding problem.

2.4 Information Theoretic Analyses

Several authors have developed and analyzed the data hiding (or watermarking) problem from an information theoretic perspective. Right from the landmark

paper from Shannon [98], which started the field of information theory, there have been a number of works analyzing the problem of communication with side information about the channel-state at the encoder [100, 44, 24]. Shannon himself had introduced this problem in [100]. Gel'fand and Pinsker [44] consider this problem in more detail and prove some results for this scenario (communication with channel state information at the encoder but not at the decoder). Heegard and El Gamal [49] analyzed a closely related problem of storing in memory with defects. Based on these works, Costa [24] showed, for Gaussian side information and Gaussian noise channel, that the capacity is same for the encoder side-information case as that for the no-side-information case.

Since then, several authors developed and analyzed the data hiding (or watermarking) problem from an information-theoretic and game-theoretic perspective (Steinberg and Mehrav [114], Chen and Wornell [19], Chou et al [20], Cohen and Lapidoth [23], and Moulin and O'Sullivan [79], and Moulin and Mihcak [78, 77]).

It has been shown that the information-theoretic prescriptions (for mean-squared distortions D_1 and D_2) translate, roughly speaking, to hiding data by means of choice of the vector quantizer for the host data (i.e., the QIM scheme discussed in the previous section), with the additive white Gaussian noise (AWGN) attack being the worst-case under certain assumptions.

Game-theoretic analyses of data hiding, with the hider and attacker as adversaries, have been provided by Moulin and O'Sullivan [79], and by Cohen and Lapidoth [23]. Estimates of the hiding capacity of an image, based on a parallel Gaussian model in the transform domain, have been provided by Moulin and Mihcak [78, 77]. The method of *types*, an important concept from information-theory,

was leveraged for data embedding in [46]. We now move on to practical schemes in the following sections.

2.5 Robust Data Hiding: Techniques and Attacks

Here we provide a brief overview of the techniques for robust data hiding. We do not attempt to provide a comprehensive overview here, but rather focus only on approaches that are closely related to the schemes presented in this thesis. As expected, almost all robust embedding methods are designed for digital watermarking applications. Note that many of these techniques assume the availability of the watermark sequence at the decoder and simply correlate it with the received sequence to *detect*, essentially hiding only one bit of information.

Some of the earlier work on robust watermarking focussed on additive techniques [26, 59, 93, 135]. These techniques are based on spread-spectrum embedding, and could survive a number of attacks, such as compression, additive noise, and signal processing operations. However, a significant downside of some of these early methods is that they require the presence of the original host signal at the decoder (i.e., they are non-blind). Moreover, as stated before, most of these techniques also assume the availability of the watermark sequence at the decoder.

Ruanaidh and Pun [93] propose a rotation, scale, and translation (RST) invariant watermarking scheme based on embedding in the log polar map of the discrete Fourier transform (DFT) coefficients (also called Fourier-Mellin or FM transform). FM transform has been popular in the pattern-recognition literature

as RST invariant features. The problem with using FM transform for data hiding is that it is difficult to modify the log-polar coefficients without inducing perceptual distortion to the image. Lin et al [62] alleviate the problem to some extent using a slightly modified approach. It should be noted that these approaches, still, have effectively only a single bit of payload.

Although it is now well-accepted that binning methods (QIM) are better suited for high-capacity hiding [76], SS techniques continue to receive a lot of attention because of their perceived advantage for achieving robustness. In [109], robust QIM-based schemes are demonstrated that provide robustness against several attacks while embedding large number of bits.

Powerful attack freeware packages are available now, such as Stirmark [85], and Checkmark [84], that induce severe geometric transformations without causing significant visual distortion to the image, effectively de-synchronizing the encoder and the decoder rendering the watermark undetectable. For example, random bending of the grid of an image turns out to be a simple yet very effective attack. For applications in costumer tracing, in which a watermark is used as a *digital fingerprint* (embedding different specific watermark sequences in different versions of the same work), attacks by the way of collusion of many costumers need to be survived [118].

To counter the desynchronization attacks, several approaches have been proposed that either attempt to resynchronize at the decoder using pilot sequences [33, 83], or embed data in geometrically invariant feature spaces [10, 7]. In the pilot sequence based schemes, the idea is to periodically embed a sequence known to both the encoder and the decoder, which can be used to synchronize. In [10],

tessellation points that are invariant to geometric transformations are used to embed information. These tessellation points can be recovered after any geometric processing such as rotation, cropping, or random bending. Another interesting approach is by Mihcak and Venkatesan [69], who embed data in some semi-global statistics of the image.

Printing followed by scanning represents a valid attack on the image with hidden copyright information. The print-scan operation induces severe distortions to the image, which include non-linear processing and geometric distortions. Only a few embedding approaches proposed in the literature can survive printing followed by scanning operation. Notable work is by Lin and Chang [61, 62], who model the print-scan process by considering pixel value and geometric distortions separately. In the pixel value model, they consider non-linear processing, blurring due to halftoning, and noise at the edges. In the geometric distortion model, authors consider rotation, scaling, and cropping attacks. An embedding scheme is proposed that is based on the log-polar map of the DFT coefficients. The effective payload here, again, is one bit as the watermark sequence is correlated at the decoder to detect the watermark. In Chapter 5, we present techniques that provide significant improvement over these schemes in terms of volume of embedding[2].

As seen above, most of the schemes achieve robustness against many severe attacks, however, they are either non-blind, or have a payload of just one bit, or both. In practical applications, it is desirable to have much higher payloads. We consider robust techniques with higher capacity in this thesis. Let us now describe how various approaches achieve perceptual transparency by image-adaptive

[2]The work has also been published in [110, 111, 112]

hiding.

2.6 Image-Adaptive Techniques

For data hiding in images (or any media), it is necessary to be adaptive to the local characteristics of the host signal, because in general all the parts of the host image do not have the same hiding *capacity*. In other words, we cannot make same amount change all the porions of an image to hide data. Thus, any scheme embedding data into a media host must provide for a way to adapt to local perceptual characteristics. Many methods for perceptual adaptation have been proposed in the literature. Early works include Chae [14], Wolfgang et al [135], and Podilchuk and Zeng [87].

For JPEG data hiding, now it is well accepted that hiding in DCT coefficients whose values are zero (after JPEG quantization), should not be used for embedding information. Examples of DCT LSB techniques that do not embed in DCT coefficients that are either 0 or 1 are JSTEG [122], and OutGuess [89]. In the following we briefly review the approaches for perceptual adaptation for SS and QIM approaches.

2.6.1 Perceptual Shaping for Spread-Spectrum Hiding

In SS hiding, a spread version of the message signal is added to the host in order to embed the data. The strength of the watermark that is added is controlled a scaling factor by which the spread sequence is multiplied before adding. These techniques adapt the strength of the watermark based on a strategy commonly

known as *perceptual shaping* (see, for example, [135], [87], and [28]). Perceptual shaping refers to the idea of adjusting the strength of the watermark based on the perceptual sensitivity of a region in the image. All these methods use some perceptual model (e.g., Watson's DCT [130] and wavelet models [131]) that assigns weights to various regions of the image. This weight determines the strength of the watermark that is added to that part of the image. A disadvantage of perceptual shaping is that, by reducing the strength of the hidden data in the perceptually sensitive area, the robustness of this data against attacks is compromised. Still, the use of perceptual shaping for image-adaptation remains the most popular approach to maintain perceptual transparency.

2.6.2 Adaptive QIM Schemes

For quantization based hiding, the idea of perceptual shaping cannot be readily applied because here the watermark is not being added whose strength be adjusted. As seen before, the QIM schemes provide a good performance by rejecting the host signal interference. However, without a good way to control the perceptual degradation by local adaptation, quantization based schemes cannot be employed for embedding high volumes of information in media hosts.

One of the earlier work on adaptive quantization based embedding was by Ramkumar [91], in which the zero-valued DCT coefficients were not modified. However, this method was designed for JPEG compression attack only, and it could not survive any other attacks. Also notable is the work by Mukharjee et al [81] who use lattice quantization to embed data and provide image adaptation by choosing different lattice structures for different *types* of blocks. A perceptual

model determines the level of embedding in a block.

Wu and Lui [136, 137] propose an adaptive method for QIM, called *uneven embedding*, in which the encoder chooses the hiding locations based on a perceptual model. In their implementation, either the information about the embedding locations is sent as side information (variable embedding rate), or the rate is fixed and embedding locations vary by shuffling (constant embedding rate). They can embed 1024 bits in a 512x512 image that survives JPEG compression attacks and moderate additive noise attacks. In Chapter 3, we propose a new image-adaptive framework that enable embedding more than 7500 bits in a 512x512 image that can also survive JPEG compression, additive noise, image resizing and tampering attacks[3].

Fridrich et al [40, 41], propose an interesting approach, called *wet paper codes* for adaptive data hiding. The idea is to write on a paper with some *wet* pots where one cannot write. In data hiding terms, there are some locations that are not good for embedding, so that the encoder is now allowed to embed information there. A disadvantage is that the method is fragile against attacks (or performs very poorly against attacks). This technique is targeted towards applications in steganography, where no attack is anticipated. In the following section, we look into approaches for steganography and steganalysis in more detail.

[3]The work has also been published in [109, 106, 51, 108]

2.7 Prior Work in Steganography and Steganalysis

Steganography, the art and science of communicating in a manner that the very presence of communication is not known to a third party, has a rich history (e.g., [119], and references in [117] and [86]). In 1983, Simmons [102] introduced the modern version of the problem: Alice and Bob are in jail, and want to hatch up an escape plan, but all their communication pass through Willie, the warden. Hence, the communication should be hidden, so that it does not incite the suspicion of Willie. The challenge in the design of steganographic systems is to communicate at high rates without being detectable via statistical, or perceptual analysis. It is also desirable that the embedded data is robust against benign attacks, such as recompression, and additive noise.

From a historical perspective, it is interesting to mention Shannon's work on cryptography [99], in which three possible methods for secure data transmission are pointed out. First, what he called, concealment systems, in which the existence of the message is concealed from the enemy (what we call, steganography), second, privacy systems, in which a technology is employed, such as advanced radio systems, that noone else has access to, and third, "true" secrecy systems, the ones in which the meaning of the message is concealed by a code or cipher. The paper (i.e., [99]) deals with the third type, the cryptography. It is argued that the second method (privacy systems) is a technological problem, and the first, i.e., steganography, is a psychological problem, which is indeed true. However, for multimedia hosts, steganography has also become a *statistical* problem, because

the process of data embedding changes the statistics of the media host, which can be detected by the steganalysts. We now describe the particular algorithms used for steganography and how steganalysts have attempted to detect them.

One of the earliest steganography approaches were based on modulating the least significant bit (LSB), both in spatial domain and transform domain (e.g., JSTEG [122]). Hiding in LSB changes the histogram in a predictable way: the number of pixels in adjacent bins with different LSBs would get equalized after embedding a binary message with equiprobable 0 and 1. The was recognized early, and used for steganalysis by using the chi-squared, or χ^2, statistic [133]. Provos's *stegdetect* [90] algorithm uses the same statistic to detect the JPEG LSB hiding. Stegdetect can be improved upon by more sophisticated detection-theoretic approaches [29, 116].

Westfeld proposed a LSB-based JPEG steganography scheme, named F5 [132], which can evade the χ^2 steganalysis. In this technique, instead of replacing the LSB, the DCT coefficients are either increased or decreased by one. This way, the equalization of adjacent frequency bins of the histogram happening due to the replacement of LSBs with random-bit messages can be avoided. This method has since been generalized to $\pm k$ embedding. A closely related approach, called stochastic modulation, was recently proposed by Fridrich et al [36]. Note that both $\pm k$ and stochastic modulation are primarily designed for spatial domain hiding; not for the transform domain.

Provos' OutGuess [89] algorithm is another technique that can evade the χ^2 steganalysis. In this algorithm, only about half of the coefficients are used for embedding, and the rest are used to compensate for the hiding, so that the his-

togram looks the same as before. Thus, if the hiding process changes a coefficient from value A to B, another coefficient is with value B is found, and changed to A. Eggers et al [32] suggest a more rigorous approach to the same end, using a method of data-mappings that preserve the frequencies of occurrence, called histogram-preserving data-mapping (HPDM).

To counter the JPEG LSB schemes such as OutGuess and F5, Fridrich et al [39, 38] propose a steganalysis technique that detects the JPEG-based methods by evaluating the increase in blockiness in the image due to block-DCT embedding. More recently, Wang and Moulin [128] propose a similar steganalysis method that evaluates the increase in smoothing within a block in addition to the increase in blockiness.

A powerful technique to detect LSB hiding, called RS steganalysis, was recently proposed by Fridrich et al [37], which can detect LSB hiding schemes, such as the $\pm k$ steganography. In this technique, sample pairs of adjacent pixels is classified into three types: regular group (R), singular group (S), and unusable group. The number of pixel pairs in each group in the cover image is approximately the same, however, LSB embedding such as F5, changes this. Thus, a statistic derived from RS analysis can be used to detect this type of steganography.

Sallee's model-based steganography [94] provides an interesting and different perspective in the design of steganographic systems, with the hider ensuring that the stego signal conforms to a given model. A method for JPEG steganography is proposed, in which the DCT coefficients are models as Cauchy random variables. It should, however, be noted that in the absence of a perfect model for the host, nothing stops the steganalyzer from selecting a *better* model by spending more

computational power. This is indeed practically shown in [11], where Sallee's Cauchy-model based JPEG steganography is broken by using only the first order statistics. In order to evade the blockiness-based steganalysis, Salle also proposes a method that compensates for blockiness [95].

While the above methods focus on quantized statistics for embedding, there are only a few approaches that look for security in the continuous domain (Guillon et al [48], and Wang and Moulin [129]). Guillon et al [48] suggest transforming the source to get a uniform PMF source. The message is hidden in this with the quantization hiding scheme, which is known not to change the PMF of uniform sources. Therefore, the PMF after transforming back is also the same as the original. This method, however, is not likely to be robust, and also, there is no way to control the distortion induced by the embedding process. Another interesting approach is that of Wang and Moulin's [129], who propose a reduced rate variant of standard QIM, called the stochastic QIM, which can be made to have zero K-L divergence. However, because of the stochastic nature of the hiding process, the method is likely to yield high error rates when embedding large volumes of data.

Most of the approaches for steganalysis focus on detecting a particular steganographic technique. Lyu and Farid [63] propose a *universal* steganalysis method based on supervised learning machine (SVM). The features they use are higher-order statistics of wavelet subband coefficients. A few more approaches use supervised learning using various features for detecting the presence of data (e.g., [115, 35]). These schemes perform very well when the SVM is trained and tested for one particular steganography algorithm. The good performance of supervised

learning can be explained by the fact that, unlike other steganalysis schemes, the decision of the detector is not based only on one image. Here, the typical changes made by the hiding algorithm can be *learned* by the detector. Obviously, the learning-based techniques would not perform that good if it is to be employed as truly *universal* (i.e., without knowing the hiding algorithm).

Analysis of steganography problem from a theoretical perspective was done by Cachin [12]. Here, the author propose that, for achieving secure communication, the Kullback-Leibler (KL) divergence between the cover and the stego distributions should be less than ϵ. Taking this perspective, the capacity of steganographic schemes has been analyzed for certain specific constraints on host distributions, and embedding schemes (O'Sulivan et al [82], and Moulin and Wang [80]). The capacities for more general cases, however, still need to be analyzed.

2.8 Summary

In this chapter, we have provided a brief overview of the data hiding problem, and discussed several techniques that are closely related to the methods proposed in this thesis. There are many issues that have not been addressed adequately in the state of the art. For example, an important point is to investigate the performance of practical image hiding schemes in the context of the capacities predicted by the information-theoretic analysis. We propose high-volume image hiding schemes in the next chapter, which employs image-adaptive criteria for embedding along with an error and erasure correction coding framework.

Chapter 3

Image-Adaptive Data Hiding

The past decade has witnessed a surge of research activity in multimedia information hiding, targeting applications such as steganography (or covert communication), digital rights management, and document authentication. Another important class of applications is the seamless upgrade of communication or storage systems: additional data and meta-content can be hidden in existing data streams, such that upgraded receivers can decode both the original and the hidden data, while existing receivers can still decode the original data. This application requires embedding relatively large volumes of data, compared to, say copyright protection applications. Robustness against attacks such as compression, and additive noise is also required. Annotation of images in the fields of medicine, biology, geography, and geology, is another application where we must hide large number of bits with robustness against a variety of compression attacks. In both these applications, it is very important not to induce any perceptual distortion to the host due to data embedding.

3.1 Introduction

In this chapter, we propose a framework for hiding large volumes of data in images while incurring minimal perceptual degradation. The embedded data can be recovered successfully, without any errors, after operations such as compression, additive noise, and image tampering. The proposed methods can be employed for applications that require high-volume embedding with robustness against certain non-malicious attacks (example applications include the ones discussed in the previous paragraph: seamless upgrade of multimedia, and annotation of images). Readers are referred to Section 1.1 for a more detailed discussion on the motivation for this work.

The hiding methods we propose in this chapter are guided by the growing literature on the information theory of data hiding (summarized in the next paragraph), but are adapted to the specific application of hiding in images. Because of our target applications, we aim for robustness not against malicious attacks such as Stirmark's geometric attacks, but against "natural" attacks such as compression (e.g., a digital image with hidden content may be compressed as it changes hands, or as it goes over a low bandwidth link in a wireless network). It turns out, however, that our schemes are actually robust against a broader class of attacks than we initially designed for, such as tampering, and a limited amount of resizing.

Information-theoretic treatments of the data hiding problem typically focus on hiding in independent and identically distributed (i.i.d.) Gaussian host samples. The hider is allowed to induce a mean squared error of at most D_1, while an

attacker operating on the host with the hidden data is allowed to induce a mean squared error of at most D_2. Information-theoretic prescriptions in this context translate, roughly speaking, to hiding data by means of the choice of the vector quantizer for the host data, with the AWGN attack being the worst-case under certain assumptions. This method of hiding was first considered by Costa [24], based on results of Gel'fand and Pinsker [44] on coding with side information (with the host data playing the role of side information). Game-theoretic analyses of data hiding, with the hider and attacker as adversaries, have been provided by Moulin and O'Sullivan [79], and by Cohen and Lapidoth [23]. Estimates of the hiding capacity of an image, based on a parallel Gaussian model in the transform domain, have been provided by Moulin and Mihcak [77]. Chen and Wornell [19] present a variety of practical approaches to data hiding, with a focus on scalar quantization based hiding, and show that these schemes are superior to spread spectrum hiding schemes, which simply add a spread version of the hidden data to the host [26]. A scalar quantization based data hiding scheme, together with turbo coding to protect the hidden data, is considered in [56], while a trellis coded vector quantization scheme is considered by Chou et al [21].

Relative to the preceding methods, a key novelty of our approach is that our coding framework permits the use of local criteria to decide where to embed data. The main ingredients of our embedding methodology are as follows.

(a) As is well accepted, data embedding is done in the transform domain, with a set of transform coefficients in the low and mid frequency bands selected as possible candidates for embedding. (These are preserved better under compression attacks than high frequency coefficients)

(b) A novel feature of our method is that, from the candidate set of transform coefficients, the encoder employs local criteria to select which subset of coefficients it will actually embed data in. In example images, the use of local criteria for deciding where to embed is found to be crucial to maintaining image quality under high volume embedding.

(c) For each of the selected coefficients, the data to be embedded indexes the choice of a scalar quantizer for that coefficient. We motivate this by an information-theoretic analysis showing that, for an idealized model [24], scalar quantization based hiding is only about 2 dB away (in terms of resilience to attack) from optimal vector quantization based hiding.

(d) The decoder does not have explicit knowledge of the locations where data is hidden, but employs the same criteria as the encoder to guess these locations. The distortion due to attacks may now lead to insertion errors (the decoder guessing that a coefficient has embedded data, when it actually does not) and deletion errors (the decoder guessing that a coefficient does not have embedded data, when it actually does). In principle, this can lead to desynchronization of the encoder and decoder.

(e) An elegant solution based on erasures and errors correcting codes is provided to the synchronization problem caused by the use of local criteria. Specifically, we use a code on the hidden data that spans the entire set of candidate embedding coefficients, and that can correct both errors and erasures. The subset of these coefficients in which the encoder does not embed can be treated as *erasures at the encoder.* Insertions now become errors, and deletions become erasures (in addition to the erasures already guessed correctly by the decoder, using the same

local criteria as the encoder). While the primary purpose of the code is to solve the synchronization problem, it also provides robustness to errors due to attacks.

Two methods for applying local criteria are considered. The first is the block-level Entropy Thresholding (ET) method, which decides whether or not to embed data in each block (typically 8×8) of transform coefficients, depending on the entropy, or energy, within that block. The second is the Selectively Embedding in Coefficients (SEC) method, which decides whether or not to embed data based on the magnitude of the coefficient. Reed-Solomon (RS) codes [134] are a natural choice for the block-based ET scheme, while a "turbo-like" Repeat Accumulate (RA) code [31] is employed for the SEC scheme. We are able to hide high volumes of data under both JPEG and AWGN attacks. Moreover, the hidden data also survives wavelet compression, image resizing and image tampering attacks.

The use of perceptual models and image-adaptation is not new in the watermarking literature. Many of the techniques proposed in the literature are based on a strategy commonly known as *perceptual shaping* (see, for example, [135], [87], and Chapter 7 in [28]). Mostly used in conjunction with spread-spectrum watermarking, perceptual shaping refers to the idea of adjusting the strength of the watermark based on the perceptual sensitivity of a region in the image. All these methods use some model that assigns weights to various regions of the image. This weight determines the strength of the watermark that is added to that part of the image. However, by reducing the strength of the hidden data in the perceptually sensitive area, the robustness of this data against attacks is compromised. It should be noted that the hiding techniques presented in this paper are significantly different from the aforementioned methods. Our approach is based

on the idea of not "disturbing" the sensitive coefficients, so as to achieve good image quality without compromising robustness. The number of bits hidden is determined dynamically by the scheme based on the host image content.

Wu and Lui [136, 137] also propose the concept of *uneven* embedding, where certain transform coefficients are not used for embedding based on a perceptual criteria. Their method, however, requires side information about the hiding locations to be sent to the decoder, which reduces the size of the payload. In contrast, our coding framework obviates the need for sending synchronization data explicitly, while providing great flexibility in terms of the use of application-specific local adaptation criteria (e.g., not hiding data in a sensitive portion of a medical image). In addition, it provides robustness against a variety of attacks such as tampering and resizing.

Note that, while the proposed coding schemes solve the specific insertion-deletion problem that arises in this setting, they do not apply to the more general insertion-deletion channel considered in [30], where the length of the overall symbol sequence can vary. In our situation, the set of candidate coefficients for embedding is the same, and is known to both encoder and decoder: the uncertainty only lies in which of these candidates were actually used for embedding.

Apart from the use of the local criteria and the coding framework, the information-theoretic analysis of scalar quantization based hiding for the idealized model in the paper by Costa [24] is also new. A similar result has been derived in independent work by Eggers et al [33]. In order to compare the theoretical capacity with practically achievable rates, we have also implemented a hiding scheme specifically optimized for AWGN attacks, which gets to within 2 dB of the scalar hiding

capacity.

The rest of the chapter is organized as follows. In section 3.2, we find the mutual information for the scalar quantization based hiding methods and also derive a decision statistic to be passed to the decoder. In Section 3.3, we introduce our image-adaptive hiding schemes. The coding framework to counter insertions/deletions and errors is described in Section 3.4 followed by a discussion on decoding (Section 3.5). A hiding method optimized to AWGN attacks is described in Section 3.6. Results are presented in section 3.7 and discussed in section 3.8.

3.2 Quantization based data hiding

In this section, we introduce our quantization-based embedding methods and derive the decision-statistic for the AWGN attack.

3.2.1 Embedding data in choice of quantizer

Data is embedded in the host medium through the choice of scalar quantizer, as in [19]. For example, consider a uniform quantizer of step size Δ, used on the host's coefficients in some transform domain. Let odd reconstruction points represent a hidden data bit '1'. Likewise, even multiples of Δ are used to embed '0'. Thus, depending on the bit value to be embedded, one of two uniform quantizers of step size 2Δ is chosen. Moreover, the quantizers can be pseudo-randomly dithered, where the chosen quantizers are shifted by a pseudo-random sequence available only to encoder and decoder. As such, the embedding scheme is not

readily decipherable to a third party observer, without explicit knowledge of the dither sequence.

Hard decision decoding in this context is performed by quantizing the received coefficient to the nearest reconstruction point of all quantizers. An even reconstruction point indicates that a '0' has been hidden. Likewise, if a reconstruction point lies on an odd quantizer, a '1' has been hidden. However, if more information regarding the statistics of the attack is available, soft decisions can be used to further improve performance. In Section 3.2.2, we compute the capacity of scalar quantization based hiding for the specific case of AWGN attacks. Implicit in our formulation is the use of soft decisions that account for both the quantization noise and the AWGN.

3.2.2 Capacity of scalar quantization based data hiding

We now show that our scalar quantization based hiding incurs roughly only a 2 dB penalty for the worst-case AWGN attack. Letting D_1 and D_2 denote the mean squared embedding induced distortion and mean squared attack distortion, the hiding capacity with AWGN attack is given by $C_v = \frac{1}{2}\log(1 + \frac{D_1}{D_2})$, in the small D_1, D_2 regime that typical data hiding systems operate [24, 79]. We compare this "vector capacity" (termed thus because the optimal strategy involves vector quantization of the host) to the mutual information of a scalar quantizer embedding scheme with soft decision decoding.

Consider a data hiding system where the information symbol to be embedded is taken from an alphabet $\mathcal{X}$. The host's original uniform quantizer is divided into M uniform sub-quantizers (each with quantization interval $M\Delta$), where $M = |\mathcal{X}|$,

a power of two. Thus, $\log_2 M$ bits are hidden per host symbol.

We consider the distortion-compensated quantization embedding scheme of [19] with soft decision decoding. Here, the uniform quantizer is scaled by $\alpha \in (0, 1]$, increasing the distance between adjacent quantizers to Δ/α. As such, the embedding robustness is increased by a factor $1/\alpha^2$ (in the squared minimum distance sense), and embedding induced distortion is increased by the same factor. Encoding the information symbol as a linear combination of the host symbol and its quantized value, as in the following, compensates for the additional distortion. Denoting the host coefficient by C, and the hidden message symbol by X, the symbol transmitted by hider is given by

$$Q_X(C) = \alpha q_X(C) + (1 - \alpha)C \tag{3.1}$$

where $q_x(\cdot)$ the scaled uniform quantizer used to embed the information symbol x (with quantization interval $M\Delta/\alpha$). Under an AWGN attack, the received symbol is

$$\begin{aligned} Y &= Q_X(C) + W \\ &= \alpha q_X(C) + (1 - \alpha)C + W \\ &= q_X(C) + (1 - \alpha)(C - q_X(C)) + W \end{aligned}$$

where W is AWGN with mean zero and variance D_2.

The parameter α achieves a tradeoff between uniform quantization noise and AWGN. The optimal value for α for maximizing the signal-to-noise ratio (SNR) at the decoder, which we have found numerically also to maximize the mutual information $I(X; Y)$, is [19]

$$\alpha_{opt} = \frac{D_1}{D_1 + D_2} \tag{3.2}$$

The probability density function of the combined additive interferers, $N = (1-\alpha)Z + W$, where $Z \equiv C - q_X(C)$ is the uniform quantization noise, is given by convolving the uniform and Gaussian densities:

$$f_N(x) = \frac{\alpha(2\pi D_2)^{-\frac{1}{2}}}{(1-\alpha)M\Delta} \int_{-\frac{(1-\alpha)M\Delta}{2\alpha}}^{\frac{(1-\alpha)M\Delta}{2\alpha}} \exp(-\frac{(x-\tau)^2}{2D_2})\mathrm{d}\tau \tag{3.3}$$

We compute the mutual information $I(X;Y) = H(X) - H(X|Y)$ for X uniform over its M-ary alphabet as an estimate of the capacity with scalar quantization based embedding. Thus, $H(X) = \log_2 M$. To find, $H(X|Y)$, we now compute $p_{X|Y}$, the conditional probability mass function of X given Y, and f_Y, the probability density function of Y.

Consider the quantization interval in which the received symbol Y appears, and define its midpoint as the origin. Letting y denote the abscissa, the nearest quantizers appear at $y = \pm\frac{\Delta}{2\alpha}$. Conditioned on the input $X = x$ and host coefficient $C = c$, the distribution of Y is given by $f_{Y|X,C}(y|x,c) = f_N(y - m_x\frac{\Delta}{2\alpha} - k_c\frac{M\Delta}{\alpha})$, with f_N as in (3.3). Here, $m_x \in \mathcal{M} = \{\pm 1, \pm 3, ..., \pm 2M-1\}$ is uniquely determined by the information symbol x, $k_c \in \mathbb{Z}$ by the host coefficient c, and the hidden quantized host coefficient $q_x(c)$ by the pair (m_x, k_c). Thus we have

$$\begin{aligned} f_{Y|X}(y|x) &= \int_{\mathcal{C}} f_{Y|X,C}(y|x,c) f_C(c)\mathrm{d}c \\ &\propto \sum_{k\in\mathbb{Z}} f_N(y - m_x\frac{\Delta}{2\alpha} - k\frac{M\Delta}{\alpha}) \end{aligned} \tag{3.4}$$

$$\begin{aligned} f_Y(y) &= \sum_{x\in\mathcal{X}} f_{Y|X}(y|x) p_X(x) \\ &\propto \sum_{m\in\mathcal{M}} \sum_{k\in\mathbb{Z}} f_N(y - m\frac{\Delta}{2\alpha} - k\frac{M\Delta}{\alpha}) \end{aligned} \tag{3.5}$$

where we have assumed that the host C and message X are statistically independent, and that the host's density f_C is roughly constant on an interval around Y, an assumption that is reasonable in the low distortion regime, where the quantization interval is small with respect to variations in the host's density. This implies that the density of Y is $\frac{\Delta}{\alpha}$–periodic, so that it suffices to restrict attention to the interval $[-\frac{\Delta}{2\alpha}, \frac{\Delta}{2\alpha}]$, with f_Y normalized accordingly. Applying Bayes' rule, the distribution of X given Y is

$$p_{X|Y}(x|y) = \frac{f_{Y|X}(y|x)p_X(x)}{f_Y(y)} \tag{3.6}$$

so that we can now compute

$$H(X|Y) = \int_{\mathcal{Y}} \sum_{x \in \mathcal{X}} p_{X|Y}(x|y) \log p_{X|Y}(x|y) f_Y(y) dy$$

and hence $I(X;Y)$.

Due to the exponential decay of the Gaussian density, the summation in (3.4) is well approximated with only the $k = 0$ term, i.e. the nearest quantization point to y corresponding to x being transmitted. Figure 3.1 plots the mutual information obtained with 2, 4 and 8-ary signaling, as well as the vector capacity. We observe roughly a 2 dB loss due to the suboptimal scalar quantization encoding strategy.

3.2.3 Soft decision statistic for Distortion Compensated hiding

We conclude our analysis by noting that the soft decision statistic, used by an iterative decoder, is the log likelihood ratio (LLR), given in the following for the

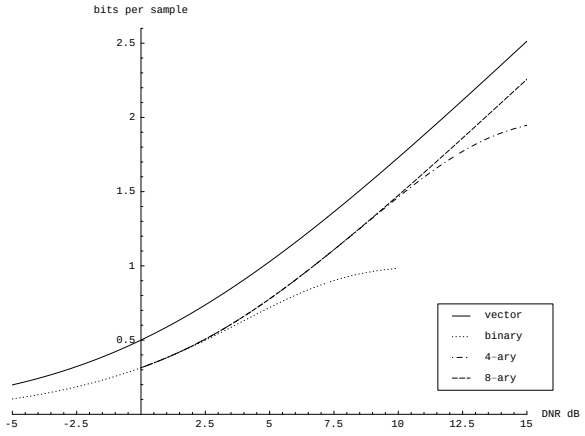


Figure 3.1: Gap between scalar and vector quantizer data hiding systems.

case of binary signaling.

$$\Lambda(y) = \log \frac{p_{X|Y}(0|y)}{p_{X|Y}(1|y)} = \log \frac{f_{Y|X}(y|0)}{f_{Y|X}(y|1)} \tag{3.7}$$

When $\alpha = 1$ and (3.4) is approximated with $k = 0$ term, the LLR reduces to

$$\Lambda(y) = \log \frac{f_W(y - \frac{\Delta}{2})}{f_W(y + \frac{\Delta}{2})} = \frac{y\Delta}{D_2} \tag{3.8}$$

We now compute log likelihood ratio (LLR) for any value of $\alpha \in (0, 1]$. We proceed by finding the conditional probability density functions $f_{Y|X}(y|0)$ and $f_{Y|X}(y|1)$, which could be written using (3.4) as convolution of uniform and Gaussian densities. Again, approximating (3.4) using the $k = 0$ term, we obtain,

$$f_{Y|X}(y|0) = \frac{\alpha(2\pi D_2)^{-\frac{1}{2}}}{2(1-\alpha)\Delta} \int_{-\frac{(1-\alpha)\Delta}{\alpha}}^{\frac{(1-\alpha)\Delta}{\alpha}} \exp(-\frac{(y - \tau - \frac{\Delta}{2\alpha})^2}{2D_2}) d\tau$$

$$f_{Y|X}(y|1) = \frac{\alpha(2\pi D_2)^{-\frac{1}{2}}}{2(1-\alpha)\Delta} \int_{-\frac{(1-\alpha)\Delta}{\alpha}}^{\frac{(1-\alpha)\Delta}{\alpha}} \exp(-\frac{(y - \tau + \frac{\Delta}{2\alpha})^2}{2D_2}) d\tau$$

The integrals in the above equations can be written as difference of two Q functions, the complimentary cumulative distribution function of a standard Gaussian random variable. We get,

$$f_{Y|X}(y|0) = \frac{\alpha}{2(1-\alpha)}\{Q(\frac{y+\Delta-\frac{3\Delta}{2\alpha}}{\sqrt{D_2}}) - Q(\frac{y-\Delta+\frac{\Delta}{2\alpha}}{\sqrt{D_2}})\}$$

$$f_{Y|X}(y|1) = \frac{\alpha}{2(1-\alpha)}\{Q(\frac{y+\Delta-\frac{\Delta}{2\alpha}}{\sqrt{D_2}}) - Q(\frac{y-\Delta+\frac{3\Delta}{2\alpha}}{\sqrt{D_2}})\}$$

Substituting above equations in LLR expression (3.7), we get,

$$\Lambda = \log \frac{Q(\frac{y+\Delta-\frac{3\Delta}{2\alpha}}{\sqrt{D_2}}) - Q(\frac{y-\Delta+\frac{\Delta}{2\alpha}}{\sqrt{D_2}})}{Q(\frac{y+\Delta-\frac{\Delta}{2\alpha}}{\sqrt{D_2}}) - Q(\frac{y-\Delta+\frac{3\Delta}{2\alpha}}{\sqrt{D_2}})} \tag{3.9}$$

Thus we get a relatively simple expression for the soft decision statistic for a general value of $\alpha \in (0, 1]$. The decision- statistic derived here is employed in the iterative decoding of the AWGN optimized hiding (Section 3.5). Note that, while we have used the $k = 0$ term in (3.4) in deriving these analytical expressions, an arbitrary degree of accuracy can be obtained by considering more terms.

3.3 Image adaptive data hiding

In order to robustly hide large volumes of data in images without causing significant perceptual degradation, hiding techniques must adapt to local characteristics within an image. Many prior quantization based blind data hiding schemes use global criteria regarding where to hide the data, such as statistical criteria independent of the image (e.g. embedding in low or mid-frequency

bands), or criteria matched to a particular image (e.g. embedding in high-variance bands). These are consistent with information theoretic guidelines [77], which call for hiding in "channels" in which the host coefficients have high variance. This approach works when hiding a few bits of data, as in most watermarking applications. However, for large volumes of hidden data, hiding based on such global statistical criteria can lead to significant perceptual degradation. Figure 3.2 shows 512×512 Harbor image with 16,344 bits hidden using local criteria and with 16,384 bits hidden (a rate of 0.0625 bits/pixel) using statistical criteria (hiding in low frequency band). Both the images were designed to survive JPEG compression at a quality factor of 25. Note that the *statistical criteria* based scheme is one that hides in all the coefficients in a predefined band. In this particular example, a low frequency band comprising of 4 AC coefficients was used. It is observed that the perceptual quality as well as the PSNR is better for the image with hidden data using local criteria. Note that though the PSNR is only marginally better (0.8 dB higher), the actual perceptual quality is much better. This illustrates that local criteria must be used for robust and transparent high volume embedding.

Although we do not use specific perceptual models, we refer to our criteria as 'perceptual' because our goal in using local adaptation is to limit perceivable distortion. As evident in the example presented (Figure 3.2), the employed criterion does succeed in limiting perceptual distortion when hiding a large volume of data. We describe two image-adaptive hiding techniques, which we had first proposed for uncoded hidden data in [108] and then with a coding framework in [51]. Figure 3.3 shows a high-level block diagram of the hiding methods presented in the following. Both the embedding methods, the entropy thresholding (ET)

(a) 16,344 bits hidden using local criteria, PSNR = 32.6 dB

(b) 16,384 bits hidden using statistical criteria, PSNR = 31.8 dB

Figure 3.2: Local vs Statistical criteria: 512×512 Harbor image with approximately same number of bits hidden using local and statistical criteria. It can be seen that the perceptual quality of the composite image is better in the former.

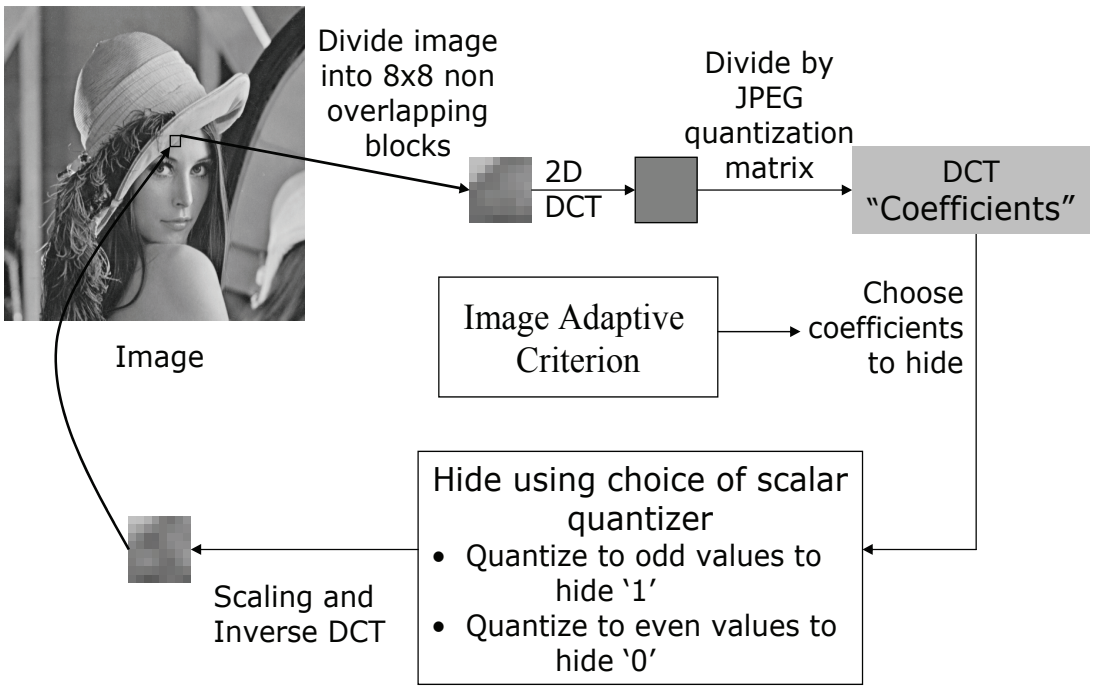


Figure 3.3: Image-adaptive embedding methodology. Data is hidden by quantizing dynamically selected DCT coefficients. In the ET scheme, the selection is done for every 8×8 block, while for the SEC scheme, a per-coefficient selection is done.

scheme, and the selectively embedding in coefficients (SEC) scheme, are based on joint photographic experts group (JPEG) compression standard. As seen in the Figure 3.3, the techniques involve taking 2D discrete cosine transform (DCT) of non-overlapping 8×8 blocks, followed by embedding in selected DCT coefficients. We now explain these two methods in more detail.

3.3.1 Entropy Thresholding scheme

The entropy thresholding (ET) scheme uses the energy (or 2-norm entropy) of an 8×8 block to decide whether to embed in the block or not. Only those blocks whose entropy exceeds a predetermined threshold are used to hide data.

The embedding procedure is outlined as follows. The image is divided into 8×8 non-overlapping blocks, and an 8×8 DCT of the blocks is taken. Let us denote the intensity values of the 8×8 blocks by a_{ij} and the corresponding DCT coefficients by c_{ij}, where $i, j \in \{0, 1, ..., 7\}$. Thus,

$$\mathbf{c} = \mathrm{DCT}_2(\mathbf{a}) \tag{3.10}$$

where DCT_2 denotes a 2D DCT.

Next, the energy of the blocks is computed as follows

$$E = \sum_{i,j} \|c_{ij}\|^2, \quad \forall \quad i, j \in \{0, 1, ..., 7\}, (i, j) \neq 0.$$

It should be noted that the DC coefficient is neither used for entropy calculation nor for information embedding. This is because JPEG uses predictive coding for the DC coefficients and hence, any embedding induced distortion would not be limited a single 8×8 block.

The blocks whose energy E is greater than a predefined threshold are selected for information embedding. These blocks are now divided by the JPEG quantization matrix whose entries are computed for a given design quality factor (QF) as per the codec implementation of *independent JPEG group* (IJG) [127]. The design quality factor determines the maximum JPEG compression that the hidden image will survive. Let us denote the quantization matrix entries for a particular quality factor QF as M_{ij}^{QF}, where $i, j \in \{0, 1, ..., 7\}$ and QF $\in \{1, 2, ..., 100\}$, where QF = 100 corresponds to the best quality image. The coefficients c_{ij} used for information embedding are computed as

$$\tilde{c_{ij}} = \frac{c_{ij}}{M_{ij}^{QF}}, \quad \forall \quad i, j \in \{0, 1, ..., 7\}. \tag{3.11}$$

Next, the coefficients $\tilde{c_{ij}}$ are scanned in zig-zag fashion, as in JPEG, to get one dimensional vector $\tilde{c_k}$ where $0 \leq k \leq 63$. The first n of these coefficients are used for hiding after excluding the DC coefficient ($k = 0$ term). Thus, low frequency coefficients are used for embedding. Bits are hidden using choice of scalar quantizer (Section 3.2). For a binary signature bitstream $\mathbf{b}$, the hidden coefficients $\tilde{d_k}$ are given using the notation in (3.1) as,

$$\tilde{d_k} = \begin{cases} Q_{b_l}(\tilde{c_k}) & \text{if } 1 \leq k \leq n, \\ \tilde{c_k} & \text{otherwise.} \end{cases} \tag{3.12}$$

where $b_l \in \{0, 1\}$ is the incoming bit that determines which one of the two quantizers $Q_1(\cdot)$ and $Q_0(\cdot)$ is used.

The hidden coefficients $\tilde{d_k}$ are reverse scanned to form an 8×8 matrix $\{\tilde{d_{ij}}\}_{i,j=1}^{8}$, and multiplied by the JPEG quantization matrix to obtain $\{d_{ij}\}_{i,j=1}^{8}$. Finally, the inverse DCT of $\{d_{ij}\}_{i,j=1}^{8}$ yields the hidden image intensity values a'_{ij} for that

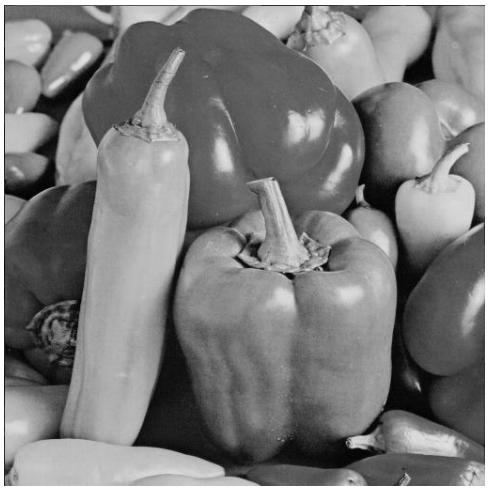
block.

Low frequency coefficients are used to embed in qualifying blocks (i.e., blocks that satisfy the entropy test). Hiding in these coefficients induces minimal distortion due to JPEG's finer quantization in this range. Thus, this scheme employs a statistical criterion by hiding in the frequency subbands of large variance, while satisfying a local perceptual criterion via the block entropy threshold.

In general, compression (quantization of the DCT coefficients) decreases the entropy of the block. Hence, in the uncoded version of the scheme, it is necessary to check that the entropy of each block used to embed information, compressed to the design quality factor, still exceeds the threshold entropy. If a particular block passes the test before hiding but fails the test after the hiding process, we keep it as such, and embed the same data in the next block. However, such a test becomes unnecessary when the ET scheme is used along with a coding framework (Section 3.4).

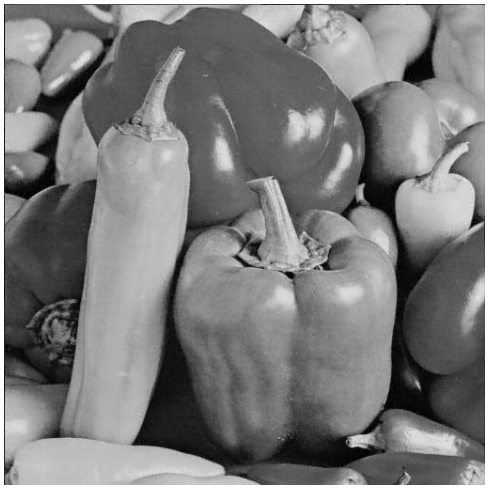
The decoder checks the entropy of each 8×8 block to decide whether data has been hidden. Two parameters are shared by the encoder and decoder in this scheme, namely, the block entropy threshold and the set of coefficients used for embedding in a block. As stated, the coefficients are scanned in zig-zag fashion, and only first n are used, excluding the DC coefficient. The parameters values are independent of the host image, and are determined based on the design quality factor used for embedding. Table 3.1 shows the values of these parameters used in our experiments.

Figure 3.4 shows the 512×512 peppers image with data hidden using the ET scheme at varying design quality factors. It can be seen that the composite images

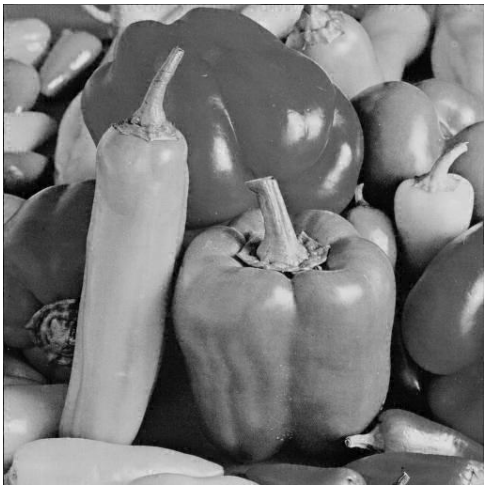


(a) The original 512×512 peppers image.

(b) Embedded Image (design QF 75) with 35,540 bits hidden.



(c) Embedded Image (design QF 50) with 14,658 bits hidden.



(d) Embedded Image (design QF 25) with 6,504 bits hidden.

Figure 3.4: ET scheme example: Thousands of bits hidden into 512×512 peppers image at varying design quality factors. As the design quality factor decreases, the robustness increases, but the volume of embedding reduces.

Table 3.1: Typical values of parameters used in ET scheme for various design quality factors

Design Quality Factor	Number of coefficients/block	Block Entropy Threshold
75	20	4000
50	14	14000
25	8	25000

are perceptually very similar to the original images, in spite of embedding several thousand bits, and robustness against high levels of JPEG compression attacks.

3.3.2 Selectively Embedding in Coefficients scheme

In the Selectively Embedding in Coefficients (SEC) scheme, instead of deciding where to embed at the block level, we do a coefficient-by-coefficient selection, with the goal of embedding in those coefficients that cause minimal perceptual distortion.

Here too, an 8×8 DCT of non-overlapping blocks is taken and the coefficients are divided by the JPEG quantization matrix at design quality factor. Thus, c_{ij} are computed using (3.10) and then divided by JPEG quantization matrix using (3.11) to get $\tilde{c_{ij}}$ in the same way as in ET scheme, but the entropy calculation and thresholding steps are skipped. Again, the coefficients are zig-zag scanned (to get $\tilde{c_k}$) and only a predefined low frequency band is considered for hiding (i.e., $1 \leq k \leq n$).

Next, we quantize these coefficient values c_k to nearest integers and take their magnitude to get r_k,

$$r_k = |Q_I(\tilde{c_k})|, \quad 1 \leq k \leq n. \tag{3.13}$$

We embed in a given coefficient only if r_k exceeds a positive integer threshold t. Embedding is again done using choice of scalar quantizers. We send either $Q_1(\tilde{c_k})$ or $Q_0(\tilde{c_k})$ depending on the incoming bit. Thus $\tilde{d_k}$ can be given as

$$\tilde{d_k} = \begin{cases} Q_{b_l}(\tilde{c_k}) & \text{if } 1 \leq k \leq n, \text{ and } r_k > t, \\ r_k & \text{if } r_k = t, \\ \tilde{c_k} & \text{otherwise.} \end{cases} \tag{3.14}$$

After reverse scanning, multiplication by JPEG quantization matrix, and inverse DCT, we get the hidden image intensity values a'_{ij} for that block.

A check is required in the scheme when the magnitude of the coefficient lies between t and $t+1$. If the quantized value $Q_{b_l}(\tilde{c_k})$ equals t in (3.14), then the decoder cannot tell whether this coefficient was not chosen for hiding because of the threshold criteria, or whether b_l was hidden in this coefficient. In coded version of the scheme, this is regarded as an erasure and decoding is performed accordingly. In the uncoded version of the scheme, the same bit b_l is embedded in the next coefficient eligible for embedding. This is done in order to maintain synchronization between encoder and decoder. Note that the decoder simply disregards all coefficients that quantize to a value with magnitude $\leq t$. This check also makes sure that there are no insertions or deletions for JPEG attacks with smaller quantization intervals (higher QFs).

The simplest SEC scheme is the zero-threshold SEC scheme ($t = 0$), where the coefficients that are not quantized to zero are used to embed information. High embedding rates are achieved using this zero-threshold SEC scheme with very low perceptual degradation, which resembles that due to JPEG compression. To understand this intuitively, it should be noted that there are many image

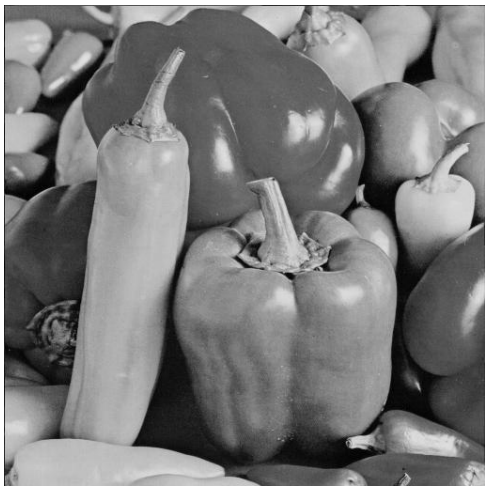
coefficients that are very close to zero once divided by the JPEG quantization matrix, and would be quantized to zero upon JPEG compression. Embedding '1' in such coefficients introduces a large amount of distortion relative to the original coefficient size, a factor that seems to be perceptually important. This is avoided by choosing not to use zeros for embedding.

Figure 3.5 shows the peppers image with several thousand bits embedded using the zero-threshold SEC scheme at varying design. Notice that, the volume of embedding is, in general, higher than the that for the ET scheme at comparable design quality factors[1]. The difference gets higher at more severe attacks (i.e., higher attack quality factors).

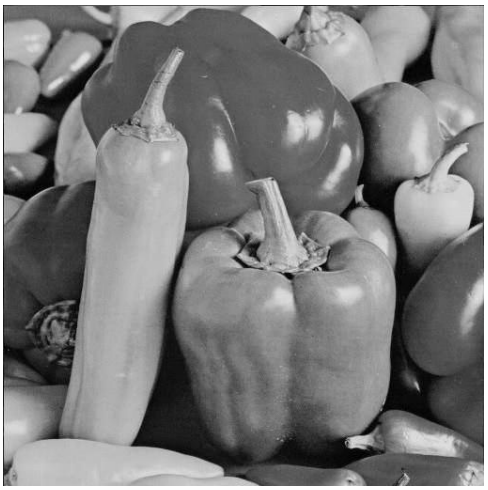
As the threshold increases, fewer coefficients qualify for embedding, and hence less data can be hidden, which provides a tradeoff between hiding rate and perceptual quality. For thresholds $t \geq 2$, it becomes difficult for a human observer to distinguish between the original and composite image, while embedding reliably at fairly high rates. Figure 3.6 shows example of embedding into a 512×512 peppers image such that it can survive 0.4 bpp JPEG compression (QF=25). Note that the composite images are indistinguishable from the original one.

In the SEC scheme, we have more control on *where to hide data* compared to the ET scheme, hence it achieves better performance in terms of smaller perceptual degradation for a given amount of data. Another key advantage of the scheme is that it automatically determines the right amount of data to be hidden in an image based on its characteristics.

[1]We use the same host image in presenting the examples for ease of comparison.

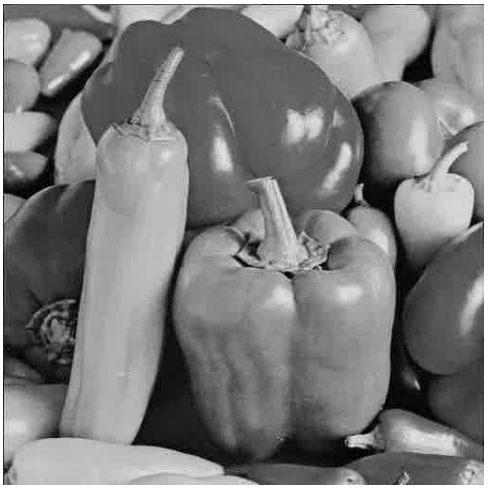


(a) The original 512×512 peppers image.



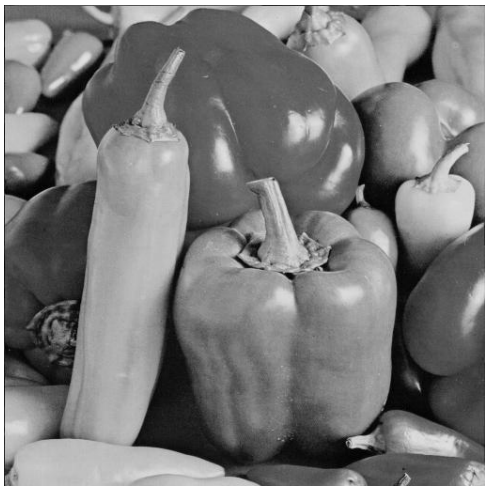
(b) Embedded Image (design QF 75) with 33,085 bits hidden.

(c) Embedded Image (design QF 50) with 19,477 bits hidden.

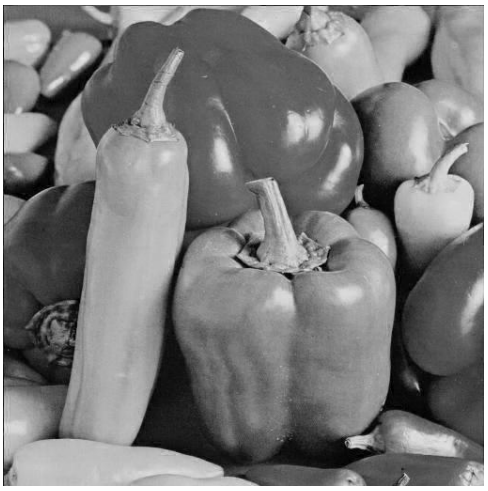


(d) Embedded Image (design QF 25) with 11,073 bits hidden.

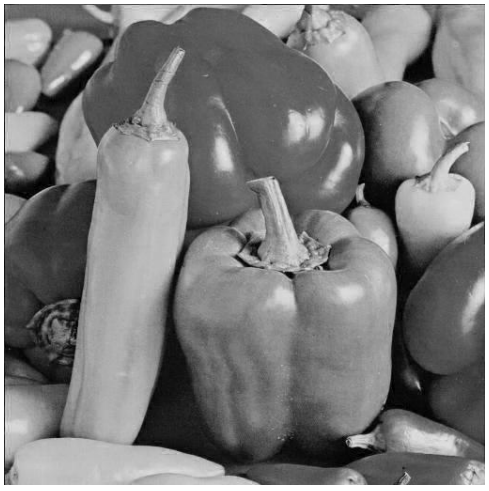
Figure 3.5: Zero-threshold SEC scheme example: Thousands of bits hidden into 512×512 peppers image at varying design quality factors.



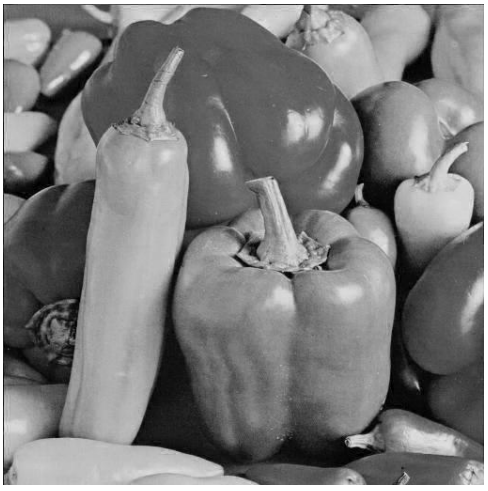
(a) The original 512×512 peppers image.



(b) Unity threshold SEC embedded image with 5,402 bits hidden.



(c) '2'-threshold SEC embedded image with 3,007 bits hidden.



(d) '3'-threshold SEC embedded image with 2,048 bits hidden.

Figure 3.6: Higher threshold SEC scheme example: Thousands of bits hidden into 512×512 peppers image at various threshold values. Design quality factor for all the hidden images is 25.

3.4 Coding for insertions and deletions

In the previous section, we noted that use of image-adaptive criteria is necessary when hiding large volumes of data into images. A threshold is used to determine whether to embed in a block (ET scheme) or in a coefficient (SEC scheme). More advanced image-adaptive schemes would exploit the human visual system (HVS) models to determine where to embed information. As shown in Figure 3.7, distortion due to attack may cause an insertion (decoder guessing that there is hidden data where there is no data) or a deletion (decoder guessing that there is no data where there was data hidden). There could also be decoding error, where the decoder makes a mistake in correctly decoding the bit embedded. While the decoding errors can be countered using simple error correction codes, insertions and deletions can potentially cause catastrophic loss of synchronization between encoder and decoder.

In the ET scheme, insertions and deletions are observed when the attack quality factor is mismatched with the design quality factor for JPEG attack. However, for the SEC scheme, there are no insertions or deletions for most of the images for JPEG attacks with quantization interval smaller than or equal to the design interval. This is because no hidden coefficient with magnitude $\leq t$ can be ambiguously decoded to $t+1$ due to JPEG quantization with an interval smaller than the design one. Both the ET and SEC schemes have insertions/deletions under other attacks.

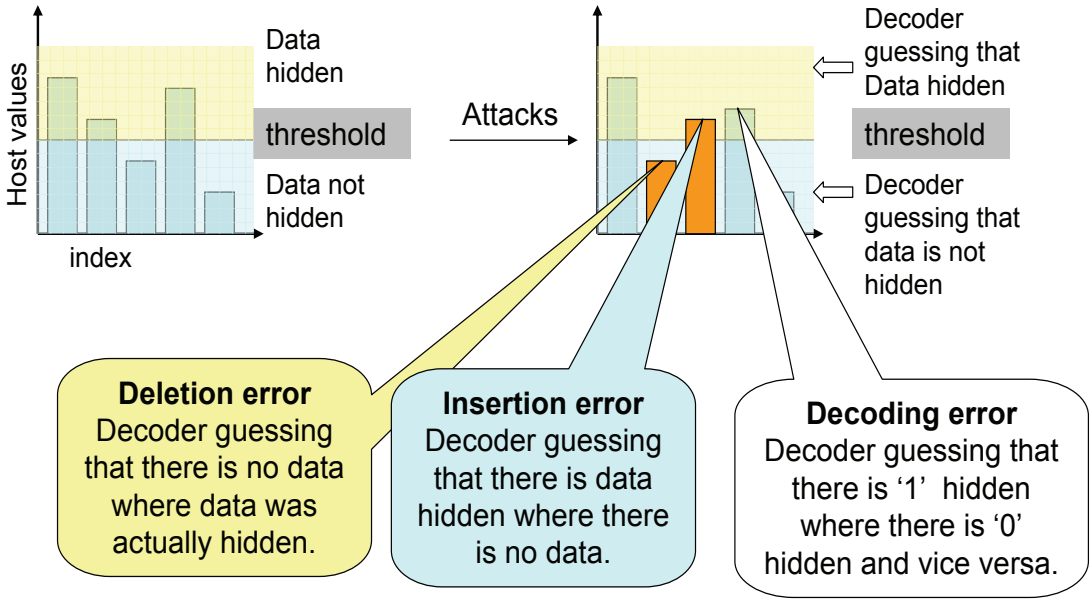


Figure 3.7: The insertion-deletion problem: Due to the presence of attacks, some coefficient values that are below the threshold increase above the threshold causing *insertions*, and values of some coefficient in which data was hidden as they were above the threshold, decreases below the threshold causing *deletions*.

3.4.1 Coding Framework

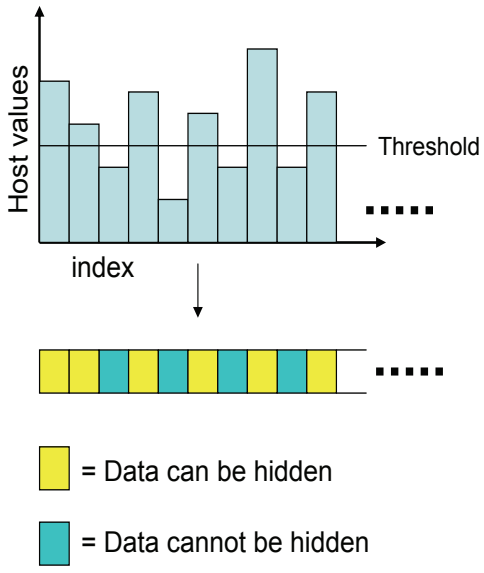
Figure 3.8 illustrates the coding framework that employs the idea of *erasures at the encoder*. The bit stream to be hidden is coded, using a low rate code, assuming that all host coefficients that meet the global criteria will actually be employed for hiding. A code symbol is erased at the encoder if the local perceptual criterion for the block or coefficient is not met. Since we code over entire space of coefficients that lie in a designated low-frequency band, long codewords can be constructed to achieve very good correction ability. A maximum distance separable (MDS) code, such as Reed Solomon (RS) code, does not incur any penalty for erasures at the encoder. Turbo-like codes, which operate very close to capacity, incur only a

minor overhead due to erasures at the encoder. Figure 3.9 shows how the sequence is decoded in the presence of attacks. As it is seen, insertions become errors, and deletions become additional erasures. It should be noted that a deletion, which causes an erasure, is about half as costly as an insertion, which causes an error. Hence, it is desirable that the data-hiding scheme be adjusted in such a manner that there are only a few insertions.

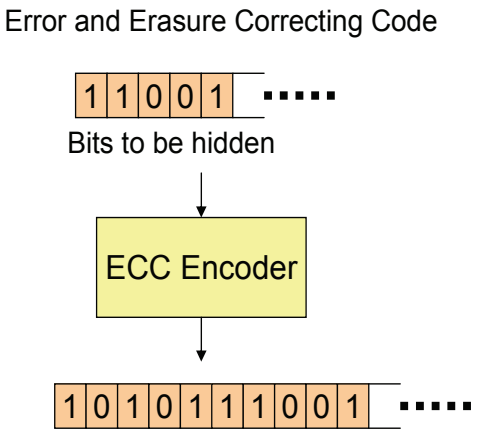
Thus, using a good erasures and errors correcting code, one can deal with insertions/deletions without a significant decline in original embedding rate. Reed Solomon codes [134] have been used for ET scheme and Repeat Accumulate codes [31] have been used for the SEC scheme as described in following sections.

3.4.2 Reed-Solomon (RS) coding for ET scheme

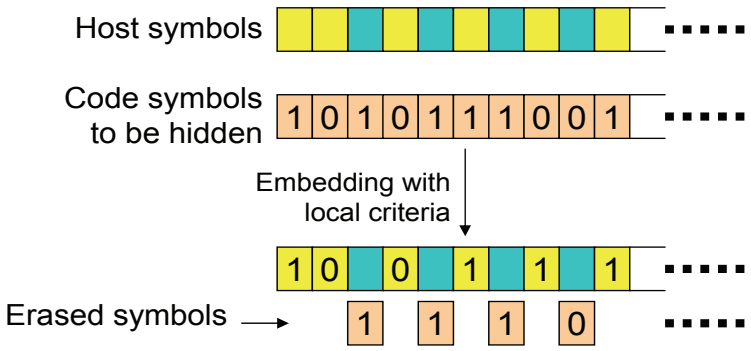
Reed Solomon codes [134] are MDS codes, such that any k coordinates of an (n,k) RS code can be used to recover the k message symbols, so that the code can correct (n-k) erasures, or half as many errors. The block length n of a Reed-Solomon code must be smaller than the symbol alphabet. More generally, an RS code can correct a pattern of e erasures and r errors as long as $e+2r \leq n-k$, which means that errors are twice as costly as erasures. RS codes use large nonbinary alphabets whose size is a power of 2, so that each symbol can be interpreted as a block of bits. This is well-matched to the block-based ET scheme, where an entire block gets inserted or deleted. Interleaving of the code symbols is required to deal with block erasures at the encoder, which tend to occur in bursts. For example, if an entire codeword were placed in a smooth area of the image, all or most of the symbols would be erased, and it would be impossible to decode this



(a) The host symbols or blocks.



(b) The error correcting code construction.



(c) Hiding with erasures at the encoder.

Figure 3.8: Coding framework illustration: How the idea of *erasures at the encoder* is employed to counter the synchronization problem. Note that the host value indicates either the block energy or the host coefficient value.

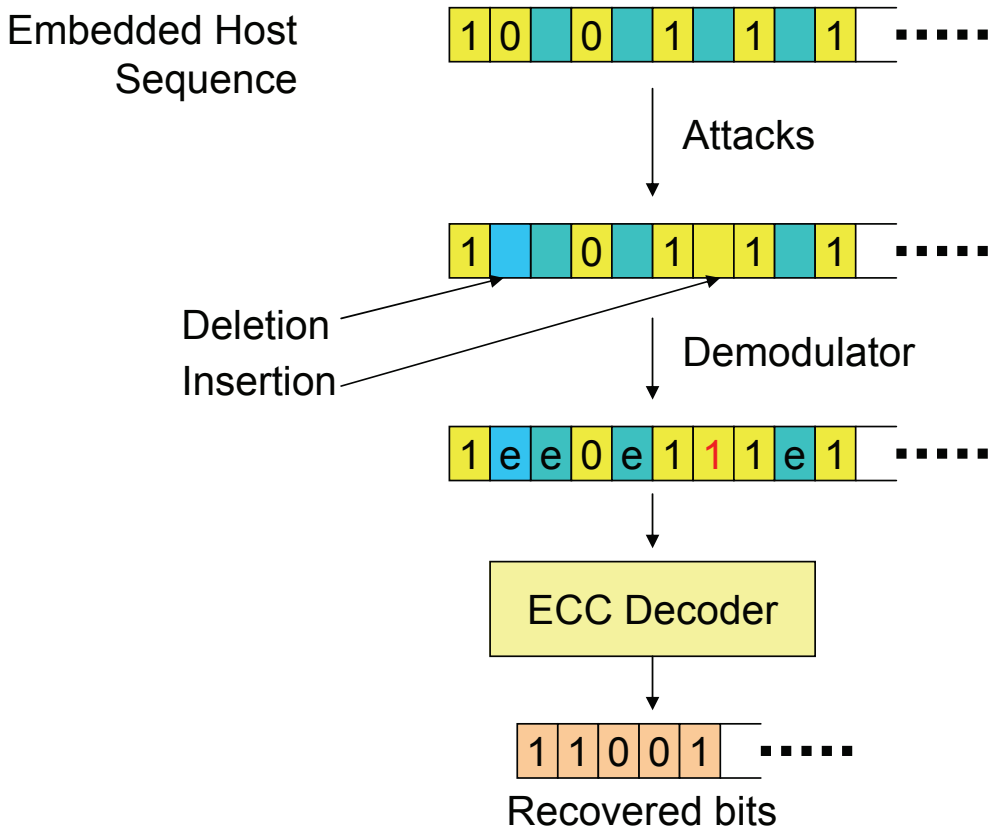


Figure 3.9: Coding framework at the decoder. Notice how the insertions become errors, and the deletions become additional erasures.

particular codeword at the receiver. The objective of the interleaving is to spread the erasures at the encoder as evenly as possible across codewords, so as to ensure that at least k out of n symbols are received at the decoder with high probability for each codeword. In particular, codewords are arranged in an image in such a way that at least certain code symbols of the codeword are in the center of the image, where the image is most likely to have details.

Let us consider an example of hiding in a 512×512 image. The image is partitioned into 4096 non-overlapping 8×8 blocks. A (128,32) RS code (i.e., rate 1/4) with symbols of size 7 bits is used. 14 coefficients are used per block. Thus there are 2 code symbols per block, and a total of 64 codewords spanning the whole image. The encoder scans the blocks one at a time, evaluates the entropy in the block, and embeds the two code symbols corresponding to the block if it passes the entropy threshold test. Otherwise, the code symbols are erased at the encoder. The rate achieved is computed as follows,

$$\begin{aligned} \text{Rate} &= 64 \, \frac{\text{codewords}}{\text{image}} \times 32 \, \frac{\text{symbols}}{\text{codewords}} \times 7 \, \frac{\text{bits}}{\text{symbol}} \\ &= 14,336 \ \text{bits/image} \\ &= 0.0547 \ \text{bits/pixel (bpp)} \end{aligned}$$

Reed-Solomon codes are not well matched to AWGN channels (where they might more typically serve as an outer code for cleaning up after an inner code matched to the channel), but are ideal for the purpose of illustrating how to deal with the erasures caused by application of local criteria at the encoder and decoder. We now turn to the SEC scheme, where we consider powerful binary codes that are well-matched to AWGN attacks, as well as close to optimal for

dealing with erasures.

3.4.3 Repeat-accumulate (RA) coding for SEC scheme

Any turbo-like code that operates close to Shannon limit for the erasures channel, while possessing a reasonable error-correcting capability, could be used with the SEC scheme. We used RA codes [31] in our experiments because of their simplicity and near-capacity performance for erasure channels [52]. A rate 1/q RA encoder involves q-fold repetition, pseudorandom interleaving and accumulation of the resultant bit-stream. Decoding is performed iteratively using the sum-product algorithm [58].

The set of candidate coefficients, which governs the length of the RA code, lies within a designated low frequency band. Let us consider an example wherein we want to hide in a 512×512 Lena image. Here, 14 coefficients per block are used (note that this parameter is independent of the host image), giving us a total maximum codeword length of 14×4096 = 57,344 for a 512×512 image. It is observed that about 11,000 coefficients satisfy the zero-threshold test for the Lena image. We choose a hiding rate of 1/7, which gives us a payload of 8192 bits. This input bitstream is coded using rate 1/7 RA code to form a codeword which is 57,344 bits long. This codeword is now hidden using the local criteria such that if a coefficient does not pass the threshold test, the corresponding code symbol is erased (i.e. not hidden).

3.5 Decoding

Hard decision decoding is used for JPEG attacks for both the ET and the SEC schemes. For the case of the RA coded SEC scheme under AWGN attack, soft decision or probablistic decoding is employed. It is well known [88] that a soft decisions decoder, leveraging knowledge of attack statistics, outperforms the hard decisions decoder. Hard decision decoding is employed for all other attacks in this paper because a detailed statistical model for these attacks is not available.

3.5.1 Hard decision decoding for JPEG attacks

The decoder estimates the location of the embedded data, and uses hard decisions on the embedded bits in these locations. The bits in the remaining locations (out of the set of candidate frequencies) are set to erasures. Since the embedding procedure of both the ET and the SEC scheme is tuned to JPEG, the decoding of embedded data is perfect for all the attacks lesser than or equal to the design quality factor (QF). The coding framework imparts robustness against insertions/deletions as well as occasional errors.

3.5.2 Soft decision decoding for AWGN attacks

Soft decision decoding can be employed for RA coded SEC scheme under AWGN attack. The decoder uses the coefficient threshold to determine whether data has been hidden or not. If the coefficient exceeds the coefficient threshold, decoder passes a soft decision statistic computed using (3.7) to the RA decoder. Otherwise an erasure (LLR, $\Lambda = 0$) is passed. The RA decoder uses the sum-

product algorithm [58] to iteratively decode the bits. We now illustrate how the coding framework employed for correcting insertions and deletions, can deal with image tampering.

3.5.3 Image Tampering

The coding framework provides flexibility to the encoder in choosing the hiding locations. The code symbols that do not pass the hiding threshold test are *erased at the encoder*. The hiding rate is chosen such that it can deal with insertions/deletions as well as errors due to attacks so that the hidden data is decoded perfectly. Here we explain how this framework can be employed to recover the embedded data against local or global image tampering, and then localize the tampered area.

By image *tampering*, we mean that a part of image is maliciously replaced by some other image data. Such a tampering can be local or global. In order to survive tampering, the code rate used is further lowered so that we can deal with the errors caused due to the replacement of the image data. Note that code rate is a design parameter shared by encoder and decoder, and hence if tampering attack is anticipated, then a low enough code rate should be chosen beforehand.

Once the hidden bitstream is decoded, localization of the tampered area can be done easily. The decoded bitstream is encoded using the same RA code parameters, so that the originally hidden RA coded stream is reconstructed. Next, the locations in the host image where errors occurred can be found by comparison. If the host image has undergone tampering, then most of the errors would be concentrated at the locations where the tampering was done. Such an ability

to robustly decode the bitstream and then localize the tampered area can be useful in medical or forensic applications to detect whether a malicious attacker has tampered with the "evidence".

3.6 Hiding optimized for AWGN attacks

In this section we present a scalar quantization based hiding strategy that is specifically tuned to AWGN attacks. The goal is to compare the achievable rates with the scalar capacity bound derived in Section 3.2.2 and the vector capacity ([24],[77]). Note that the image adaptive hiding schemes considered so far are not optimized to AWGN attacks. They use a local criteria, so that some of the coding effort is 'used-up' in dealing with insertions and deletions. Also, the DCT coefficients are divided by JPEG quantization matrix, which does not provide equal robustness to all of them against AWGN attacks. In the following we describe the embedding system, which uses scalar quantization based distortion compensated hiding, RA codes, and soft decision decoding using the statistic derived in Section 3.2.3.

As in the theoretical formulations, the problem is to hide in a host in such a way that the data hider induces a mean squared error of at most D_1, while the attacker is allowed a maximum mean squared error of D_2. In order to compare with the information theoretic limits (see, for example, Costa [24] and Moulin and O'Sullivan [79]), we assume that both the encoder and the decoder know the D_1 and D_2 values. We employ the distortion compensated hiding scheme (Section 3.2.2), which has been shown in [19] to achieve capacity for some specific

cases. Here, the uniform quantizer is scaled by $1/\alpha$, where $\alpha \in (0, 1]$, and the information symbol is encoded as a linear combination of the host symbol and its quantized value as in (3.1). Local criteria are not used, and the quantizer step size is kept same for all DCT coefficients (as opposed to using the JPEG quantization matrix). $\alpha \in (0, 1]$ is computed using (3.2) and is known to both encoder and decoder. RA codes are used to code the input bitstream to generate a huge codeword. This codeword is embedded bit-by-bit in all the coefficients within a designated band using distortion compensation. At the receiver, the soft decisions are computed using (3.9) and passed to the RA decoder which uses the sum-product algorithm [58] to iteratively decode the bits.

We use this hiding strategy to illustrate that using relatively simple RA codes with distortion compensated hiding, we can reach about 2 dB close to the scalar capacity (Section 3.7). However, it should be noted that this scheme is not likely to survive other attacks, and cannot be applied practically unless the attack is known to be AWGN.

3.7 Results

We now show that using the proposed image-adaptive hiding methods, one can hide a large volume of data with minimal perceptual degradation. We use peak signal-to-noise ratio (PSNR) as an objective metric to quantify the quality of the hidden image. PSNR is defined as,

$$PSNR = 10 \log_{10} \left(\frac{255^2}{MSE} \right) \tag{3.15}$$

where MSE stands for average mean squared error between the original and the given image. Table 3.2 shows the number of bits hidden and the corresponding observed PSNR for various images with data hidden using uncoded zero-threshold SEC scheme. Data is hidden in raw (uncompressed) images, and robustness of these images is characterized by the design QF, which determines the maximum level of JPEG compression the images can survive. It is observed that the PSNR of the hidden image is significantly higher than that of the corresponding JPEG compressed image at the same design QF. Note that, the PSNR is measured with respect to the original uncompressed image in both the cases. For example, the PSNR of JPEG compressed Baboon image at QF = 25 is 25.89 dB, while a much higher PSNR of 32.27 dB is observed for the same image with 25,331 bits hidden at a design QF of 25. Similar behavior has been observed for all the test images. The hidden image quality can be further improved by using higher threshold SEC scheme, which provides us with a trade-off between the image quality and the volume of embedding at a given robustness (determined by design QF). Table 3.3 shows the performance of the higher threshold SEC scheme for various images at a design QF of 25. In almost all these cases, it is impossible for a human observer to tell the hidden image apart from the original one.

We now present the performance of our schemes under various attack scenarios. Coding is used in all the attack scenarios (except JPEG compression where uncoded transmission is good enough for error free recovery), so that all the hidden bits can be decoded in spite of the errors due to attack. Note that the 'number of bits' reported in the following sections are actually the 'number of *information* bits' (i.e., the number of bits hidden before coding). Results for both

RS-ET and RA-SEC systems have been provided for JPEG and AWGN attacks. For all other attacks, only the RA-SEC system is used. We discuss in Section 3.8 why RA-SEC system is preferred.

3.7.1 JPEG attacks

Since the embedding procedure of both ET and SEC schemes is tuned to JPEG, the decoding of embedded data is perfect for all the attacks lesser than or equal to the design quality factor (QF). Table 3.4 shows the number of bits embedded (with perfect recovery) in uncoded and coded ET and SEC schemes at various design QFs, under JPEG attacks for 512×512 Lena image.

3.7.2 AWGN attacks

Table 3.5 summarizes the results for the ET scheme with RS coding and SEC scheme with RA coding against AWGN attack. The number of bits embedded is listed for the 512×512 Lena image. The 'attack power' reported here is the actual power of the added noise converted to the dB scale (i.e., the ratio of variance of the added noise to that of a Gaussian with unit variance). Figure **??** shows the attacked 512×512 Lenna image, in which 6301 bits are embedded.

Although the RS code is not the best choice for AWGN, it is adequate for mild attacks. RA-coded SEC scheme uses soft decision statistic of the AWGN for decoding (as in (3.8) in Section 3.2.2), and performs better than RS coded ET system at higher attack powers. A worst case attack D_2 is assumed by the decoder to compute the soft-decision statistic, and the hidden image is also attacked at

Table 3.2: Zero-threshold SEC scheme: PSNR and number of bits hidden for various 512×512 images at different design quality factors. The number of bits hidden are reported for uncoded hiding.

Image	QF=25		QF=50		QF=75	
	# bits	PSNR (dB)	# bits	PSNR (dB)	# bits	PSNR (dB)
Lena	11,044	34.58	18,786	38.07	31,306	39.90
Peppers	10,447	35.89	18,972	38.03	32,567	39.63
Baboon	25,331	32.27	44,142	34.50	66,911	36.05
Bridge	24,633	32.34	42,615	34.64	63,955	36.32
Couple	15,545	34.05	27,823	36.25	44,227	38.03
Boat	15,234	34.21	26,518	36.47	41,826	38.33

Table 3.3: Higher-threshold SEC scheme: PSNR and number of bits hidden for various 512×512 images using different threshold values at design QF=25. Using higher thresholds provide very good quality hidden images with a lower volume embedding.

Image	Thresold = 1		Thresold = 2		Thresold = 3	
	# bits	PSNR (dB)	# bits	PSNR (dB)	# bits	PSNR (dB)
Lena	4,913	41.43	2,595	44.58	1,820	46.60
Peppers	5,063	41.12	2,810	44.09	1,976	46.18
Baboon	13,065	35.98	5,763	39.92	3,247	43.27
Bridge	11,403	37.19	5,202	41.03	3,185	43.96
Couple	7,329	39.20	3,751	42.76	2,513	45.18
Boat	6,859	39.39	3,362	42.97	2,264	45.46

Table 3.4: Performance of coded and uncoded ET and SEC schemes under JPEG attacks at various quality factors

QF	attack compr. (bpp)	ET scheme # of bits		SEC scheme # of bits	
		uncoded	coded	uncoded	coded
25	0.42	6,240	4,608	11,044	7,168
50	0.66	15,652	12,096	18,786	13,824
75	1.04	34,880	30,560	31,306	23,893

Table 3.5: Performance of ET scheme with RS coding and SEC scheme with RA coding under AWGN attack. For the ET scheme, one codeword (8 bits long) is hidden per block. 20 AC coefficients constitute the candidate embedding band for the SEC scheme.

Attack power (dB)	ET Scheme		SEC Scheme	
	# of bits	RS code (n,k)	# of bits	RA code (1/q)
10.0	7,040	(256,55)	7,447	1/11
12.5	6,528	(256,51)	6,826	1/12
15.0	3,584	(256,28)	6,301	1/13

the same D_2. Note that if the actual attack is lesser than D_2, the performance would at least be as good as the one reported here.

3.7.3 Wavelet compression attacks

Wavelet compression (JPEG 2000) was used to attack the images with hidden data using SEC scheme with RA coding. Table 3.6 gives the number of bits hidden in 512×512 Lena image under various levels of attack compression. Figure 3.11 shows the composite Lenna image after wavelet compression attack at 0.8 bits per pixel. Note that, in the results reported in Table 3.6 (including the image in Figure 3.11), data was hidden in the image using SEC scheme at design quality factor of 25, and 20 coefficients were used per block, scanned in the zig-zag fashion.

Figure 3.10: AWGN attacked composite Lenna image. 6301 hidden bits hidden against an additive noise (SNR = 15dB). All the embedded bits are recovered successfully.

The JPEG 2000 compression was done using the Jasper codec [4].

3.7.4 Image Tampering

The hiding schemes presented here are resilient to image tampered in various ways. Table 3.7 gives the number of bits hidden in 512×512 Lena image when a part of host image is replaced by other image data. Figure 3.12 shows an example attacked image where 20% of the image is cropped out and new image data is put in that place. In spite of this malicious tampering of the image, all the embedded 5,208 bits are recovered successfully after the attack. The hidden data can be

Table 3.6: Performance of RA coded SEC scheme for 512×512 Lena image under wavelet compression attack

Attack Compression (bpp)	Hiding Rate # of bits	RA code rate (1/q)
0.800	7,447	1/11
0.530	4,096	1/20
0.400	2,730	1/30

Table 3.7: Performance of RA coded SEC scheme for 512×512 Lena image under image tampering. Here, 27 coefficients are used per block

Percentage of image tampered	Number of bits	RA code rate (1/q)
10 %	9,216	1/12
20 %	5,820	1/19
30 %	4,608	1/24

decoded even if the tampering is not localized. Figure 3.13(a) shows Lena image tampered globally, and still all the 6,301 hidden bits can be recovered successfully. Figure 3.13 (b) shows the localization results for the tampered image of Figure 3.13 (a).

3.7.5 Image Resizing

Image resizing is a popular attack method wherein the image is shrunk to a smaller size and scaled back to its original size so that there is loss of information in the process without causing significant perceivable distortion. Various interpolation methods can be used to resize and the most popular ones are bilinear, bicubic and nearest neighbor interpolations. Again, the RA coded SEC scheme is used for hiding in 512×512 Lena image at design quality factor of 25 and 20 coefficients are used per block. The hidden image survives large amount of resizing

Figure 3.11: Wavelet compression attack: all the hidden 7447 bits are recovered successfully after the composite image is compressed using wavelet transform at 0.8 bits per pixel.

using bicubic interpolation method. Table 3.8 gives the number of bits hidden against the percentage of resizing done using bicubic interpolation. Less data can be hidden when hidden image is resized using other interpolation techniques. Table 3.9 gives the number of bits hidden against bilinear and nearest neighbor resizing attacks. It should be noted that the perceptual quality of the attacked image is also worse in the latter cases, which forbids the attacker from using a higher percentage of resizing with bilinear or nearest neighbor interpolation.

Figure 3.12: 20 % of 512×512 Lena image tampered. All the embedded 5820 bits were recovered successfully after the tampering attack.

3.7.6 Image-in-Image hiding

In steganographic applications it is desirable to hide an image called signature image into another image called host or cover image. The hiding techniques developed here allows us to hide large volume of data with perfect recovery and hence can be used to hide large signature images with robustness against JPEG attacks. For example, signature images as large as 256×256 pixels can be hidden in a 512×512 cover image (Figure 3.14). The uncoded scheme is employed here, because we need robustness only against JPEG compression and higher embedding rate is desirable. First, the maximum number of bits that can be hidden

(a) 512×512 Lena image tampered globally

(b) Localization of tampered area at the decoder for the globally tampered image above

Figure 3.13: Global and Localized image tampering and localization of the tampered area. All the embedded 6301 bits are recovered after the attack.

Table 3.8: Performance of RA coded SEC scheme for 512×512 Lena image under image resizing attack using bicubic interpolation

Percentage Resizing	Hiding Rate # of bits	RA code rate (1/q)
10 %	7,447	1/11
15 %	6,826	1/12
20 %	6,301	1/13

Table 3.9: Performance of RA coded SEC scheme for 512×512 Lena image under image resizing attack using bilinear and nearest neighbor interpolation

	Nearest neighbor interpolation		Bilinear interpolation	
Percentage Resizing	Number of bits	RA code (1/q)	Number of bits	RA code (1/q)
2 %	6,301	1/13	2,275	1/36
5 %	4,096	1/20	2,155	1/38
10 %	2,275	1/36	1,241	1/66

in the host image is determined by going through the image and computing the number of coefficients that satisfy the local criteria at desired design quality factor. Then, the signature image is hidden after being JPEG compressed to a level that its size is smaller than the maximum number of bits that can be hidden.

3.7.7 AWGN optimized hiding

For the AWGN optimized hiding scheme discussed in Section 3.6, we found the minimum distortion to noise ratio (DNR) for which decoding was perfect for a 512×512 image at various RA code rates. Table 3.10 compares the DNR observed for simple scalar quantization based hiding ($\alpha = 1$), and distortion compensated scalar quantization hiding with optimal α ($= \frac{D_1}{D_1+D_2}$) to the theoretical scalar (Section 3.2.2) and vector [77] capacities.

We observe that we are only about 2 dB away from the theoretical scalar

(a) Original 512×512 Harbor image

(b) Composite image

(c) Original 256×256 signature image

(d) Recovered signature image

Figure 3.14: Image-in-Image hiding example

Table 3.10: Comparison of observed and theoretical capacities

RA code rate	Scalar quant. schemes, DNR (dB)		Theoretic Capacity DNR (dB)	
	($\alpha = 1$)	(opt. α)	Scalar	Vector
1/3	4.3180	2.1261	0.2500	-2.3107
1/4	3.2790	0.8365	-1.0000	-3.8278

capacity using distortion compensated quantization based hiding with RA coding. Most of this gap is probably due to the limits on the performance of the regular RA codes, which exhibit gaps of comparable size (e.g., about 1.5 dB for rate 1/3) from the Shannon limit over the classical AWGN channel as well [31]. An interesting question for future study is whether this gap can be closed further using more powerful codes such as regular and irregular LDPCs [43, 64] and irregular RA codes [52], known to work close to the Shannon limit over the AWGN channel. Another significant observation is that there is a gain of more than 2 dB when distortion compensation scheme is used as compared to the performance without distortion compensation ($\alpha = 1$).

3.7.8 Online Demonstration

A Web demo of the system proposed in this chapter is available at [1]. A screen-shot of the demo webpage is shown in Figure 3.15. The user is allowed to select the volume of data that is to be embedded, which determines the amount of robustness. User can provide an image and a message, which is then hidden into the uploaded image. A secret passcode needs to be given by the user while encoding, which is needed to retrieve the message at the decoder. This demo uses the SEC scheme with RA coding, and can survive the attacks mentioned earlier.

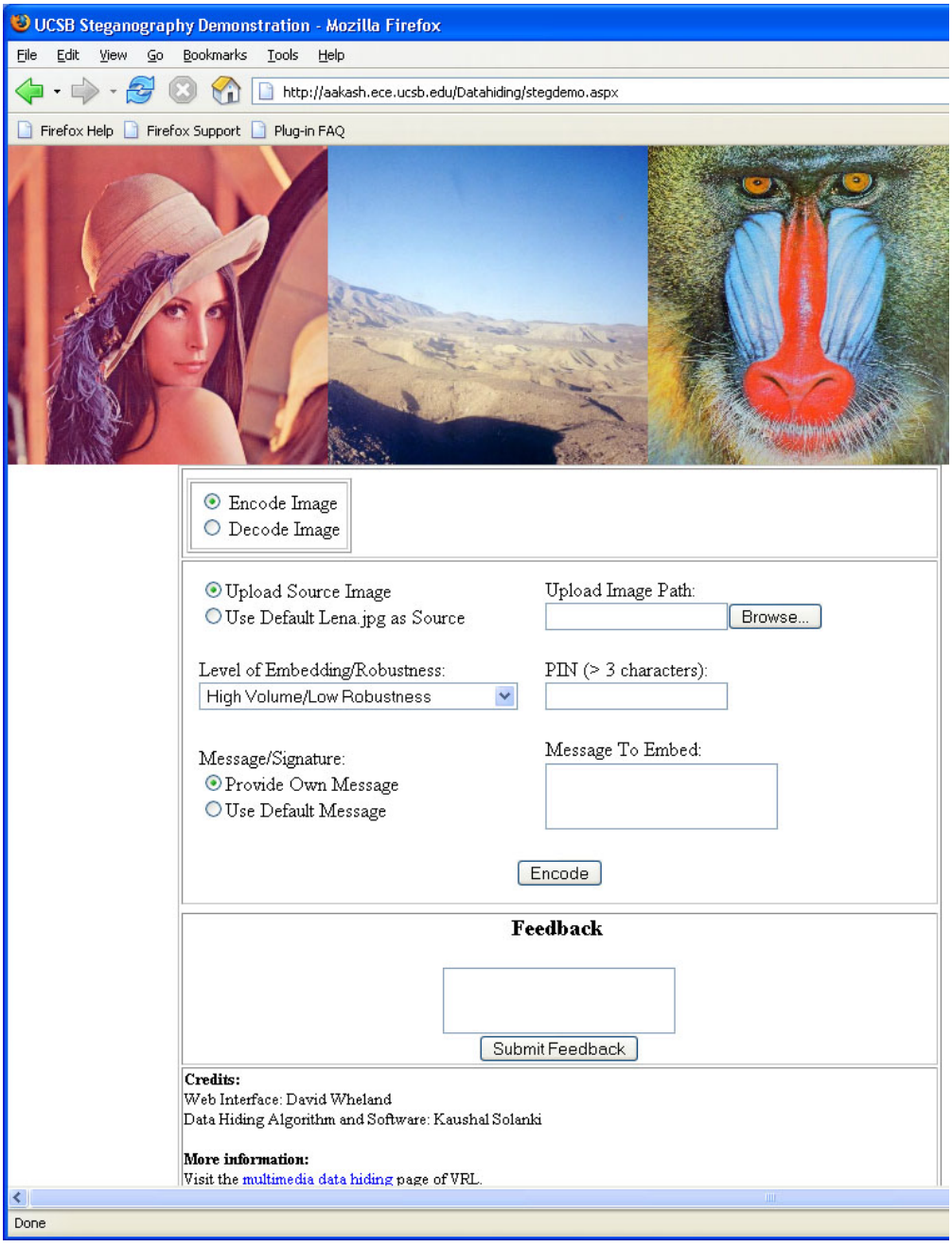


Figure 3.15: A screen-shot of the online demonstration of the high-volume data hiding system proposed in this chapter.

3.8 Discussion

The hiding methods presented in this paper are geared towards high volume embedding while preserving the perceptual quality and achieve robustness against JPEG attacks. It should be noted that we use ET scheme with RS coding mainly to explain our ideas of local adaptation and coding framework, while in most practical scenarios, the RA coded SEC scheme is used. The RA-SEC system provides a better performance in terms of robustness and perceptual quality. This is because the turbolike RA codes operate operate very close to the capacity, and the SEC scheme provides a better control on 'where to hide data'. Soft decision decoding of the RA codes is performed for AWGN attack, and hard decision decoding is performed otherwise.

While the AWGN attack is not common in the watermarking literature, it has been shown in information-theoretic studies ([23],[77]) to be the worst-case attack in certain idealized game-theoretic settings, where the mean squared distortion due to the attack is constrained. The information-theoretic "goodness" of our schemes is therefore demonstrated by our numerical results that show that, by appropriate use of soft decisions, we do approach the information-theoretic hiding capacity (with scalar quantization) under AWGN attacks. Of course, from a practical point of view, hard decisions must be employed for attacks (such as compression) whose statistics are difficult to quantify. Also, there are many attacks that induce large mean-squared distortion, but little perceptual distortion. Examples include Stirmark random bending [85], rotation, cropping, and print-scan. These geometric attacks tend to de-synchronize the decoder. Modifications to the

current hiding framework so that it allows re-synchronization of the decoder for these attacks is an avenue of future work.

It can be seen that the proposed hiding schemes survive wavelet based compression and image resizing attacks. This is because these attacks do not entirely destroy the low frequency DCT coefficients where the majority of bits have been hidden. Note that wavelet-based compression does not change the image mean squared error drastically (as opposed to geometric attacks). Hence, based on the arguments of the previous paragraph, it is not surprising that the hidden bits survive this attack. The same arguments hold true for the image resizing attack when the original image size is known to the decoder, or if the attacker scales the image back to its original size. In spite of this restriction, the presented results are significant because they indicate that the hidden bits can survive errors caused due to interpolation.

The image-in-image hiding presented here uses the fact that we can send a high volume of data with robustness against JPEG compression using uncoded SEC scheme. The signature image is compressed into a sequence of bits and these bits are hidden into the host (disregarding the actual meaning of the bits). The system is designed for the worst anticipated attack. In practice, the attack level is seldom known apriori, and if the actual attack is less severe than the design attack, we are still stuck with the design signature image quality. Ideally, we would like an image-in-image hiding scheme that results in graceful improvement in the image quality with less severe attacks. Such schemes require *joint* source-channel coding, which has been studied for the Gaussian channel (see, for example, [17, 103]). Development of similar techniques for data hiding is an important research area. A

first attempt at building such gracefully improving image-in-image hiding system is presented in the next chapter, where a hybrid digital-analog (joint source-channel) coding scheme is proposed. It leverages the current image-adaptive hiding framework for sending digital data and involves transmission of the analog residues using a new method.

Chapter 4

Joint Source-Channel Hiding

In several applications, the signature signal, which is to be hidden into a media host, is also a media data such as an image, video, audio, or speech. Examples include embedding an image into another image, or hiding video in video [15]. The need to embed a media signal arises in applications such as steganography, in which an image need to be conveyed to another party without revealing the existence of communication. Another application in which image-in-image hiding comes up naturally is when a logo is to be embedded into another image or video.

More recently, data hiding has been applied for *error concealment* of video and images. With commercialization of wireless video, and high-quality video webcasts, there is an increasing push for video coding techniques that can provide good-quality video without annoying artifacts in the presence of packet loss during transmission. Several authors have used data hiding to embed a low resolution version of the same video into the original video which is to be concealed ([5, 6, 113]). Then, at the receiver, the embedded low resolution version can be

recovered in the presence of packet loss during transmission, and the appearance of annoying artifact can be avoided. Many authors report results better then conventional error concealment systems [5, 6, 113]. In these applications too, the signature signal is a media data.

For hiding media signature signals, it is not needed to recover the signature perfectly. In this case, the signature data need to be received only with a fidelity criteria, and some error in the received signature signal is acceptable. When designing practical systems for hiding media in media (using conventional separate source and channel coding), the signature signal must be compressed to a size less than the number of bits that can be embedded into the host (or the message carrying *capacity* of the host). These compressed bits of the signature signal are then embedded into the host using appropriate channel coding. The message carrying capacity of the host is determined by the strength of the attack that is anticipated.

Obviously, in the above scenario, for a system that must survive strong attacks, fewer bits can be embedded, and hence, the signature signal need to be heavily compressed before hiding. In real-world scenario, the attack strength is seldom known beforehand, and hence, a practical system must be designed keeping the worst-case attack in mind. Thus, even when actual attack is very mild, one has to live with the poor heavily compressed signature signal quality, which was designed for the worst-case attack. It is highly desirable to have a system that can allow better quality of received signature data when the actual attack is mild. Design of such schemes require *joint* source-channel coding, which we study in this chapter.

4.1 Introduction

We consider the problem of image-in-image hiding in this chapter, where, the basic design criteria are as follows: (a) the degradation to the host image is imperceptible, (b) it should be possible to recover the hidden, or signature, image under a variety of attacks, and (c) the quality of the recovered signature image should be better if the attack is milder. In recent work [19, 33, 51, 108, 109], it has been shown that digital data can be effectively hidden in an image so as to satisfy criteria (a) and (b) by hiding in the choice of quantizer for the host data. The main idea is to view the data hiding problem as communication with channel side information ([22, 24, 79]): the channel experienced by the data comprises of the host interference and the attack, and the channel side information is the knowledge of the host. Therefore, recent advances in source coding and channel coding can be leveraged for developing data hiding schemes.

Unfortunately, these schemes do not satisfy the design criterion (c) - they exhibit the threshold effect: if the actual attack is more severe than the attack the scheme was designed for, there is a catastrophic failure in recovering the hidden image, while if the actual attack is less severe, then we are still stuck with the design attack image quality. In practice, the attack level is seldom known apriori, and ideally, we would like a scheme that results in graceful improvement and degradation in the image quality with less and more severe attacks respectively. Such schemes require *joint* source-channel coding, which has been studied for the Gaussian channel in [17, 72, 73, 123]. However, to the best of our knowledge, such schemes have not been studied for the data hiding channel.

Having provided the motivation, let us now summarize the main factors that led us to investigate joint source-channel codes for information embedding.

1. In many applications, the information to be hidden is a media data (e.g. images, video, speech and audio). These signals are inherently *analog*, or in more technical terms, *continuous alphabet sources*. For these signals, perfect recovery is not required, and receiving the signal with predetermined fidelity criteria is enough.

2. Since the attack strength is seldom known beforehand, it is desirable to have a system that allows recovering better quality signature data when the attack is mild. This way, we can construct robust data hiding systems, which are designed for severe attacks, but would enable us to receive better quality signature if the composite signal undergoes a milder attack.

3. If the embedded data is meant for more than one receivers with different channels (i.e., the broadcast scenario), it is desirable to have a system that can provide better quality signature signal for the receivers seeing mild attacks.

In this chapter, we present a hybrid digital-analog (joint source-channel) coding scheme for image-in-image hiding. It leverages an earlier digital scheme based on image-adaptive criteria and turbo-like repeat-accumulate (RA) codes, presented in Chapter 3 (also published in [51, 109]), and involves the transmission of the analog residue using a new method, which is similar in flavor to the quantization index modulation commonly used in digital schemes. At the decoder, we focus on JPEG attacks. The proposed scheme shows (perceptual as well as

mean-square error) improvement over the purely digital scheme in [51, 109] as the level of the JPEG compression attack decreases.

The rest of the chapter is organized as follows. In Section 4.2, we provide a background of joint source-channel coding for the data hiding problem. In Section 4.3, we describe our method for transmitting the analog residue and derive the minimum mean-square error estimator (MMSE) for the analog signature under uniform quantization attack. We assume that the quantization matrix of the JPEG attack is known to the decoder. In Section 4.4, we describe our hybrid digital-analog scheme and present the results. We present the conclusions in Section 4.6.

4.2 Joint Source-Channel Hiding

In this section, we develop the concept of joint source-channel data hiding, and provide an overview of the system that is employed in this chapter. We start with a discussion on joint source-channel coding for the classical communication systems in Section 4.2.1. After that we analyze the theoretical limit for the performance of any joint source-channel hiding scheme (Section 4.2.2). A prior approach for graceful improvement is briefly described next (Section 4.2.3), followed by a big picture overview of the employed system (Section 4.2.4).

4.2.1 Joint Coding for Classical Communication Systems

A number of joint source-channel coding methods have been proposed for the Gaussian channel ([17, 72, 73, 103, 123]). In [17], codes based on chaotic systems

have been proposed, which recently were shown to have optimal scaling properties in the high signal-to-noise regime in [123]. In [73, 103], hybrid digital-analog codes have been proposed. For the data hiding channel (communication with side information about the channel state at the encoder), joint source-channel codes have not been studied so far and a number of issues are open.

4.2.2 Theoretical Limit

Let us first describe some fundamental limits for a common model for the data hiding channel ([24, 79]). The hider is at most allowed to introduce a mean-square error D_1 per host symbol. Further, we assume a Gaussian attack (which simply adds i.i.d. Gaussian noise), which introduces an additional distortion of at most D_2 per host symbol.

An information theoretic analysis of the Gaussian data hiding channel, reveals that the maximum possible rate of data transmission over this channel (the capacity of the channel) can be achieved by hiding in the choice of the host vector quantizer ([24], [19]). Motivated by these results, a number of practical schemes have been developed in the literature using recent advances in source and channel coding (see, for example, [109], [22]).

Now, we consider an embedding scenario in which there is a continuous alphabet signature source, which is to be embedded into the host signal, with the same hider and attacker distortion constraints of D_1 and D_2 respectively. At the receiver, we are interested in recovering the signature with distortion of D_3 per signature symbol. Note that, in general, the host and the signature have different sizes, and so we assume that ρ channel uses per source symbol are allowed. We

are interested in finding an answer to the following question: What is the smallest D_3 that can be achieved for a given D_1, D_2, ρ? Here, we answer this question for a Gaussian signature source with zero mean and variance σ^2. To obtain distortion D_3, from rate distortion theory ([25]), we know that at least

$$R(D) = \frac{1}{2}\log_2\left(\frac{\sigma^2}{D}\right) \text{ bits/source symbol} \tag{4.1}$$

have to be transmitted. On the other hand, we know from [24] that at most

$$C = \frac{1}{2}\log_2\left(1 + \frac{D_1}{D_2}\right) \text{ bits/channel use} \tag{4.2}$$

can be transmitted over the above described data hiding channel. Since we are allowed ρ channel uses per source symbol, we have $D_3 \leq \rho C$, which yields,

$$D_3 \geq \frac{\sigma^2}{\left(1 + \frac{D_1}{D_2}\right)^{\rho}} =: D_{min}. \tag{4.3}$$

Thus we get an expression for the lower bound on the distortion that is incurred by the signature source. Given D_1, D_2 and ρ, the smallest feasible distortion above can be approached in principle by separate source and channel coding; the source encoder aims to optimally compress the source to within distortion D_3 and the channel encoder transmits the compressed source reliably over the channel.

It should, however, be note that the separation theorem for communication with encoder side information [49], holds only asymptotically, i.e., for infinitely long codewords. Moreover, a separate coding scheme has the following threshold behavior.

- Even if the Gaussian attack channel introduces a distortion less than D_2, we suffer distortion D_{min}, even though in principle we can have smaller distortion.

- If the Gaussian attack channel introduces a distortion more than D_2, the channel does not have enough capacity to transmit the source, and channel decoder makes mistakes most of the time.

The goal of joint source-channel coding is to smoothen out this threshold behavior. Moreover, a much simpler code can potentially be designed, which can match the performance of much more complicated separate source and channel codes.

4.2.3 Prior Art: Multi-bit Hiding

While joint source and channel coding for the data hiding channel has not be analyzed in the literature so far, here, we briefly overview a prior approach, that aims to receive data with better fidelity for less severe attacks.

Wu and Lui [136, 137] propose the concept of multi-bit embedding with the goal of receiving some bits for strong attacks, and receive more bits when the attack is mild. Here, bits are embedded into both low and high frequency bands, so that those bits that are hidden in the low frequency bands are received for lower quality factor JPEG attacks[1], and the bits in the mid frequency bands will also be received when the attack quality factor is higher.

This system though achieves some basic graceful improvement, it is quite naive in its design. It is not derived from the vast literature on joint source-channel codes. Only a limited number of levels of improvement are possible with this design. Also, it is not straightforward to use this kind of implementation for embedding media signature data. Note that a similar approach was briefly

[1]Note that lower quality factor JPEG attack means that the compression is higher.

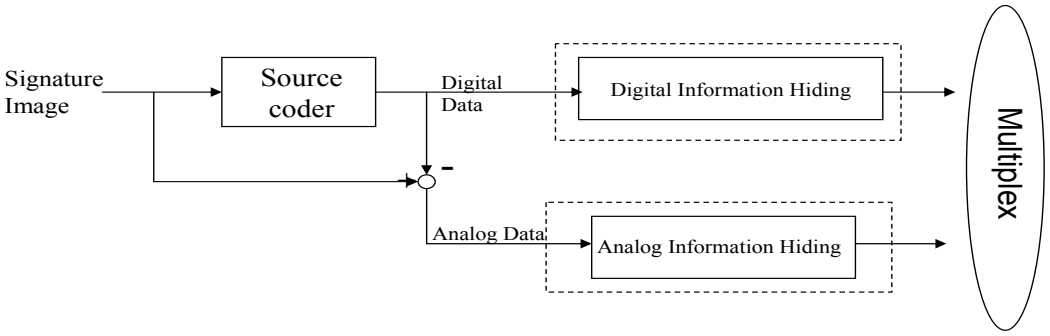


Figure 4.1: The proposed hybrid digital-analog joint source-channel coding scheme.

suggested in [16] too. We now describe a much more powerful and flexible system, which is based on hybrid digital-analog joint source-channel coding.

4.2.4 Proposed System: Hybrid Digital-Analog Hiding

We exhibit a practical hybrid digital-analog scheme for image-in-image hiding, which is similar to the scheme proposed in [103] for the Gaussian channel. A block diagram of the proposed hybrid digital-analog system is shown in Figure 4.1. The idea is to compress the signature image efficiently into a sequence of bits, which is hidden using a digital hiding scheme proposed in last chapter (also published in [109, 51]). The residual error between the original and compressed signature image is then hidden using an analog hiding scheme (proposed in Section 4.3). With practical issues in mind, we focus our attention to JPEG compression attacks instead of the Gaussian attack. We chose to develop a hybrid digital-analog scheme for the following purposes.

1. It allows us to exploit advantages of the digital scheme in [109, 51], which hides high volume of data using image-adaptive criteria and turbo-like codes,

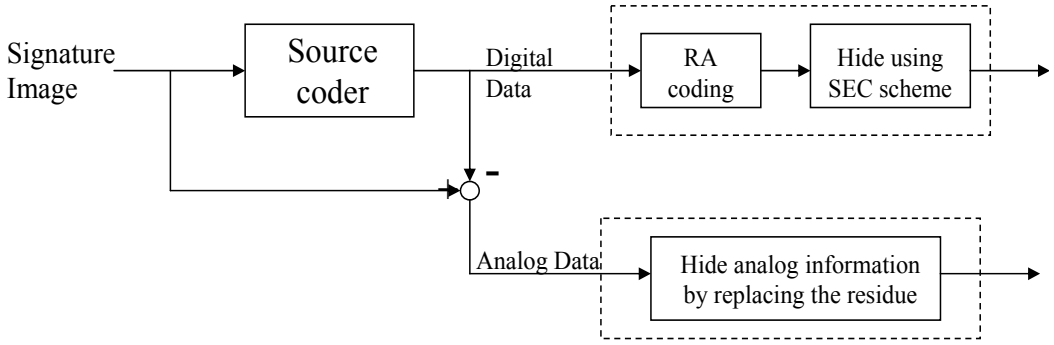


Figure 4.2: The hybrid scheme employed in this chapter: SEC scheme with RA encoding is used for digital transmission, and a new analog information hiding scheme is proposed.

and is also robust against a variety of attacks.

2. Due to the limited dynamic range of the analog residue, it is feasible to send them reliably over a limited number of host symbols.

With above observations, we now present a more refined block diagram of the proposed hybrid scheme in Figure 4.2. An important ingredient of our joint source-channel coding scheme is a new method to embed analog residue into the host, which is described in the following section.

4.3 Hiding Analog Information

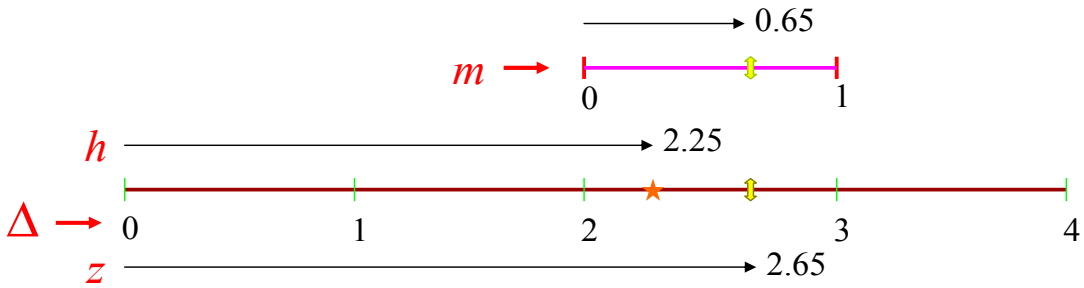
In this section, we propose a strategy to hide an analog number into a host sample. The hiding strategy involves quantization of the host followed by replacing the residue with the appropriately scaled source and is given in Section 4.3.1. The MMSE decoder is derived in Section 4.3.2.

4.3.1 Hiding using scalar quantization of the host

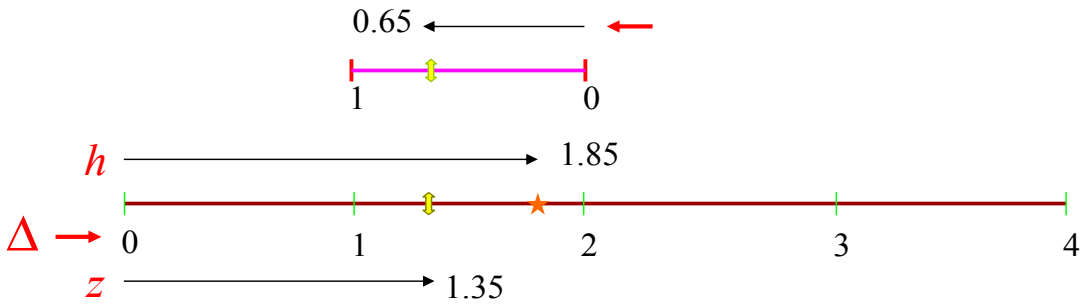
To hide an analog number m into a host sample h, we first quantize the host h using a quantizer of step size Δ, and then replace the residue with the source m, which has been *companded* or scaled to lie in the interval $(0, \Delta)$. Let us consider an example, shown in Figure 4.3 (a), where $\Delta = 1$ and the host symbol is, say, 2.25. We want to send a source symbol whose value is 0.65 (a real number $\in (0, \Delta)$) through the hiding channel. The encoder first determines that the host symbol lies between 2 and 3 (an interval $(n\Delta, (n+1)\Delta)$), then it sends the source symbol directly within that interval, i.e., it just sends 2.65. In practice, we use a hiding strategy that always *measures* the message m from an even reconstruction point of the host. This is shown in Figure 4.3 (b), in which the host symbol is 1.85, and we again wish to send the source whose value is 0.65. The encoder determines that the host value is between 1 and 2, and hence, sends 1.35 (which is 0.65 *measured* from 2). This is done to avoid catastrophic error when a hidden coefficient switches to a different integer interval as a result of attack. Thus, the symbol y to be sent for hiding a message m into a host symbol h is given by,

$$\begin{aligned} y &= \Delta(\lfloor h/\Delta \rfloor) + m, \text{ if } \lfloor h/\Delta \rfloor \text{ is even,} \\ &= \Delta(\lfloor h/\Delta \rfloor + 1) - m, \text{ if } \lfloor h/\Delta \rfloor \text{ is odd.} \end{aligned} \tag{4.4}$$

Here, $\lfloor \cdot \rfloor$ denotes the `floor` operation (defined as the largest integer smaller than or equal to the given number).



(a) Embedding message $m = 0.65$ into host symbol with value 2.25, and $\Delta = 1$.



(b) Embedding message $m = 0.65$ into host symbol with value 1.85, and $\Delta = 1$.

Figure 4.3: Analog information hiding: data is hidden simply by quantizing the host, and replacing the residue by the analog signature data after scaling or companding. As seen in (b) above, the host value is between 1 and 2, the message is always *measured* from the even reconstruction point (i.e., 2).

4.3.2 JPEG attacks and MMSE decoding

The JPEG compression performs uniform quantization of the discrete cosine transform (DCT) coefficients of 8×8 blocks of the image. Hence we derive the MMSE decoder for the above hiding scheme under uniform quantization attack, when the reconstruction points of the attack quantizer are known to the decoder, but not to the encoder. In this section, we use bold italics to represent random variables; their realizations are denoted by corresponding italic letters.

We consider the case of hiding a uniform random variable $\boldsymbol{m} \sim U[0,1]$ using (4.4) into an independent host coefficient $\boldsymbol{h}$ to obtain $\boldsymbol{y}$. In practice, even if $\boldsymbol{m}$ is not $U[0,1]$, it can be transformed in to an uniform random variable by applying the inverse of its distribution function. Without loss of generality, we assume $\Delta = 1$. In this analysis, we restrict our attention only to attacks with quantization interval less than or equal to the design interval. Note that, in practice, the design interval will be an entry in the design JPEG quantization matrix, which will be chosen to be the worst case attack. Denoting the attack quantization interval by $\delta \leq 1$, the received symbol $\boldsymbol{z} = Q(\boldsymbol{y})$, where $Q(\cdot)$ denotes the uniform quantization with an interval δ, and with zero as one of the reconstruction points. Note that all JPEG quantizers have zero as one of its reconstruction points. Thus, $\boldsymbol{z} \in \{..., -2\delta, -\delta, 0, \delta, 2\delta, ...\}$. The MMSE decoder is simply the conditional expectation $E[\boldsymbol{m}|\boldsymbol{z} = z]$. In the following, we consider various cases depending upon z, and find the conditional expectation by identifying the conditional density of $\boldsymbol{m}$ given $\boldsymbol{z} = z$.

If $z = a\delta$ is received, then y necessarily lies in the interval $[(a - 1/2)\delta, (a + 1/2)\delta)$, which we call its *ambiguity* interval (see Figure 4.4). Let us consider the

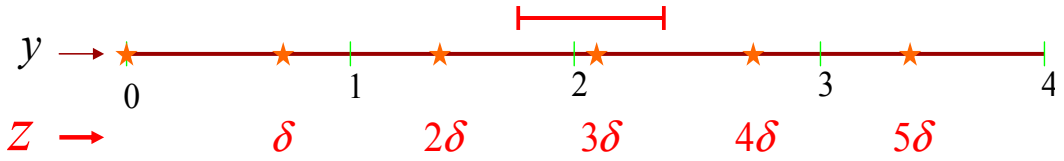


Figure 4.4: Ambiguity interval: If $z = a\delta$ is received, then the sent symbol, y, necessarily lies in the interval $[(a - 1/2)\delta, (a + 1/2)\delta)$, which is termed its ambiguity interval.

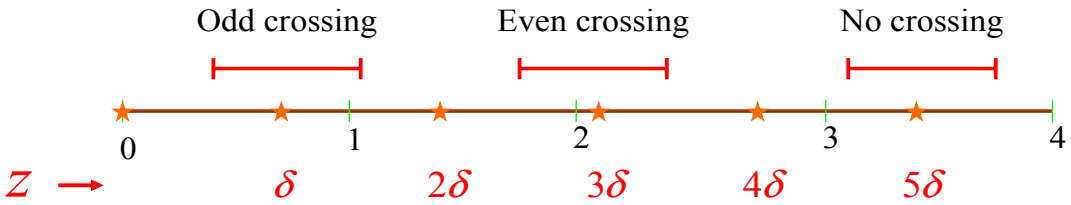


Figure 4.5: The three cases of ambiguity interval.

integer interval in which z is received, say $[n, n + 1)$. As shown in Figure 4.5, there are three possibilities with the ambiguity interval:

(i) No crossing: The ambiguity interval for y does not cross into another integer interval, that is,

$$z - \frac{\delta}{2} \geq n \text{ and } z + \frac{\delta}{2} < n + 1. \tag{4.5}$$

(ii) Even crossing: The ambiguity interval crosses an even integer, that is,

$$z - \frac{\delta}{2} < n \text{ and } n \text{ is even, or,}$$
$$z + \frac{\delta}{2} \geq n + 1 \text{ and } (n + 1) \text{ is even.}$$

(iii) Odd crossing: The ambiguity interval crosses an odd integer, that is,

$$z - \frac{\delta}{2} < n \text{ and } n \text{ is odd, or,}$$
$$z + \frac{\delta}{2} \geq n + 1 \text{ and } (n + 1) \text{ is odd.}$$

Now we proceed to find the MMSE estimates of the message $\boldsymbol{m}$ for all the three cases.

(i) No crossing: In this case,

$$f_{\boldsymbol{m}|\boldsymbol{z}}(m|z) = U\left[(a-1/2)\delta, (a+1/2)\delta\right).$$

The corresponding MMSE estimate is,

$$\hat{m} = \begin{cases} z-n & \text{if } n \text{ is even,} \\ (n+1)-z & \text{if } n \text{ is odd.} \end{cases} \tag{4.6}$$

(ii) Even Crossing: As mentioned above there could be two cases for even crossing, each involving either n or $(n+1)$ being even. The analysis is similar in both the cases and hence we just consider the first case (n even). Let us define $R_1 = n-(z-\delta/2)$ and $R_2 = (z+\delta/2)-n$ as the distances between the even crossing point n, and, the lower and upper points of the ambiguity interval respectively. Note that $R_1 + R_2 = \delta$. Defining the events $A := \{\boldsymbol{y} \in [n-R_1, n)\}$ and $B := \{\boldsymbol{y} \in [n, n+R_2)\}$, we have,

$$\begin{aligned} f_{\boldsymbol{m}|\boldsymbol{z}}(m|z) = {} & f_{\boldsymbol{m}|\boldsymbol{z},A}(m|z,A)\cdot P(A|z) \\ & + f_{\boldsymbol{m}|\boldsymbol{z},B}(m|z,B)\cdot P(B|z) \end{aligned}$$

where,

$$\begin{aligned} P(A|z) &= P(\lfloor \boldsymbol{h} \rfloor = (n-1), \boldsymbol{m} \in [0,R_1] | \boldsymbol{z} = z) \\ &= \frac{P(\lfloor \boldsymbol{h} \rfloor = (n-1), \boldsymbol{m} \in [0,R_1], \boldsymbol{z} = z)}{P(\boldsymbol{z}=z)} \\ &= \frac{P(\lfloor \boldsymbol{h} \rfloor = (n-1))\cdot P(\boldsymbol{m} \in [0,R_1])}{P(\boldsymbol{z}=z)} \\ &= \frac{P(\lfloor \boldsymbol{h} \rfloor = (n-1))\cdot R_1}{P(\boldsymbol{z}=z)}. \end{aligned} \tag{4.7}$$

Similarly,

$$P(B|z) = \frac{P(\lfloor \boldsymbol{h} \rfloor = n) \cdot R_2}{P(\boldsymbol{z} = z)} \tag{4.8}$$

where,

$$P(\boldsymbol{z} = z) = P(\lfloor \boldsymbol{h} \rfloor = (n-1)) \cdot R_1 + P(\lfloor \boldsymbol{h} \rfloor = n) \cdot R_2.$$

Note that, for a slowly varying host distribution, we have, $P(\lfloor h \rfloor = (n-1)) \approx P(\lfloor h \rfloor = n)$, so that, (4.7) and (4.8) can be approximated as $P(A|z) = R_1/\delta$, and $P(B|z) = R_2/\delta$.

Since the event $A \cap \{\boldsymbol{z} = z\} = \{\boldsymbol{m} \in [0, R_1)\}$, we have $f_{\boldsymbol{m}|\boldsymbol{z},A}(m|z, A) = U[0, R_1]$. Hence, the MMSE estimate is,

$$\hat{m} = \frac{R_1}{2} P(A|z) + \frac{R_2}{2} P(B|z).$$

Again, for a slowly varying host distribution, after some simplifications, we get,

$$\hat{m} = \frac{\delta}{2} - \frac{R_1 R_2}{\delta}. \tag{4.9}$$

(iii) Odd crossing: Following the analysis of the even case, define R_1 and R_2 as distances between the crossing point and lower and upper points of the ambiguity interval respectively. Here, we get the MMSE estimate for the general case as,

$$\hat{m} = \frac{2 - R_1}{2} P(A|z) + \frac{2 - R_2}{2} P(B|z)$$

and for the slowly varying host distribution, we get,

$$\hat{m} = 1 - \Big(\frac{\delta}{2} - \frac{R_1 R_2}{\delta}\Big). \tag{4.10}$$

Hence, we have the MMSE estimate for all the cases which can be used for decoding when decoder knows the JPEG compression quantization matrix.

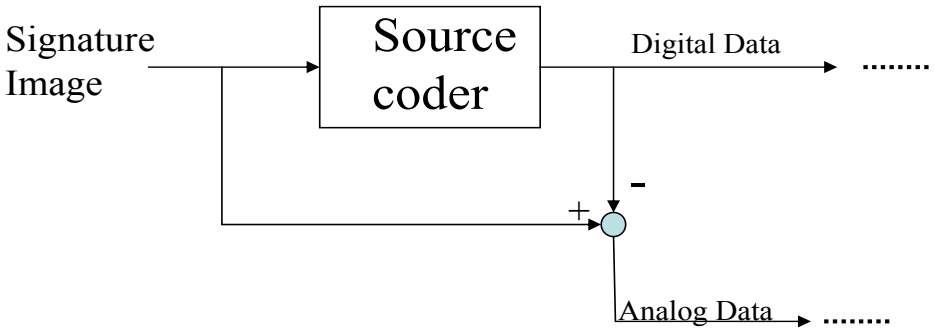
4.4 Image-in-Image Hiding

In this section we describe the actual implementation of the entire system for image-in-image hiding. The encoding process can be divided into following parts.

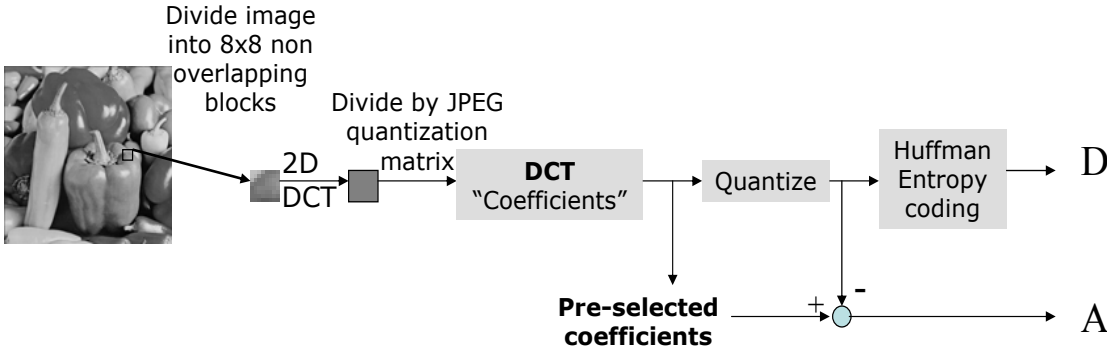
Processing the signature image: This step involves separating the signature image into digital and analog parts. We use a JPEG-based implementation as illustrated in Figure 4.6. Note that a block-DCT approach is used here just to illustrate our ideas, and in general, any compression mechanism could be employed. As shown in Figure 4.6 (b), the image is compressed using JPEG to generate a bitstream, which constitutes the digital part. The analog part is obtained by computing the residual errors of pre-selected DCT coefficients after the quantization based on design *signature* quantization matrix. Note that, the design quality factor, and the number of analog residues chosen to send, are predetermined at the design stage.

Allocating the channels: Here, we allocate the host coefficients (i.e., channel) for the digital and analog parts respectively. A few low frequency coefficients (other than the DC coefficient) of each 8×8 host block are reserved for the analog channel. Remaining low and/or mid frequency coefficients are dedicated to the digital channel. An example allocation is presented in Figure 4.7. The allocation of the digital and analog channels is done beforehand at the design stage. Thus the decoder would know where to look for analog and digital data respectively.

Hiding the digital part: The digital bitstream is hidden into its allocated channel using the RA-coded Selectively Embedding in Coefficients (SEC) scheme of [109, 51]. The bitstream to be hidden is coded using turbo-like RA code at a low



(a) The signature-image processing block.



(b) Conversion of signature image into digital part and analog residue.

Figure 4.6: Processing the signature image into digital part and analog residue: It can be seen that the particular implementation used here is based on JPEG compression. It should be noted that, in general, any compression method can be employed.

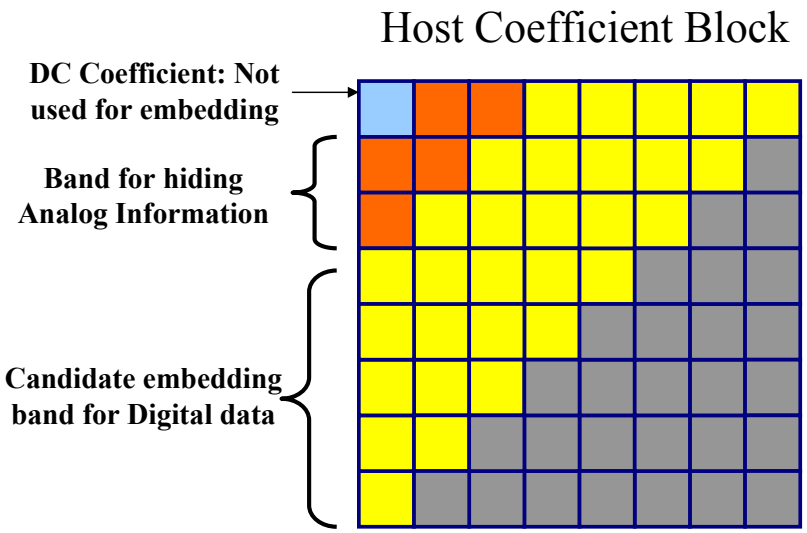


Figure 4.7: An example allocation of the host coefficient block for hiding the digital and analog parts.

rate. This coded bitstream is hidden into the host coefficients such that a code symbol is *erased at the encoder*, if the floor of its magnitude is smaller than or equal to a predetermined integer threshold. The decoder uses the same threshold criteria to estimate the erasure locations. The RA code rate is designed in such a way that one can also deal with the additional errors and erasures due to attack.

Hiding the analog part: The analog residues of selected low frequency coefficients are sent through its allocated channel using the hiding scheme of Section 4.3. Since the residue always lies in $[0, \Delta_{sig})$, where Δ_{sig} is specified by the design quantizer, we simply scale it to lie in $[0, 1)$.

The decoder decodes the analog and digital parts separately and adds them together to give an estimate of the sent signature image. The decoding of the analog part is done using the knowledge of attack δ, and assuming a slowly varying host distribution (Section 4.3.2). The digital part is iteratively decoded using

sum-product algorithm.

4.5 Results

Now we present three example implementations to show that there is an improvement in perceptual quality as well as the mean-squared error (MSE) for the received signature image as the attack becomes milder. Note that though we present a few specific examples here, the scheme is applicable to any image-in-image hiding scenario.

Example 1: We hide a 128×128 image into a 512×512 image, with the design quality factor of 25. Figure 4.8 shows the recovered signature images when the host image undergoes JPEG compression at varying levels, starting from the worst case QF of 25. Table 4.1 shows the observed MSE per coefficient for these images at various attack quality factors. The signature image is JPEG compressed at QF = 10 to form the digital part and the residues of 16 low frequency coefficients make up the analog part. We use one coefficient from each 8×8 host block for transmitting the analog data. 34 coefficients constitute the *digital channel*.

Table 4.1: Example 1: MSE per coefficients for varying levels of attacks. A 128×128 peppers image has been hidden in a 512×512 harbor image.

QF	25	35	45	55	65	75	85	95
comp.	93.5%	90.4%	88.7%	87.2%	85.0%	81.9%	75.8%	57.7%
MSE	0.0286	0.0321	0.0193	0.0149	0.0119	0.0060	0.0043	0.0025

Example 2: A 256×256 image is hidden with a design QF of 50. Figure 4.9 shows the recovered signature images when the composite image undergoes varying levels of JPEG compression attacks. Table 4.2 shows the corresponding MSE

(a) attack QF = 25 (b) attack QF = 35 (c) attack QF = 45

(d) attack QF = 55 (e) attack QF = 65 (f) attack QF = 75

(g) attack QF = 85 (h) attack QF = 95 (i) No attack

Figure 4.8: Example 1: Hiding a 128×128 peppers image into a 512×512 harbor image (not shown here). The signature images received after various levels of JPEG compression are shown along with the corresponding observed MSE per coefficient.

of the received image. The signature image is JPEG compressed at QF=18, and residues of 12 low frequency coefficients constitute the analog part. 3 coefficients per host block are used for sending analog residue and another 32 coefficients form the candidate embedding band for the digital data.

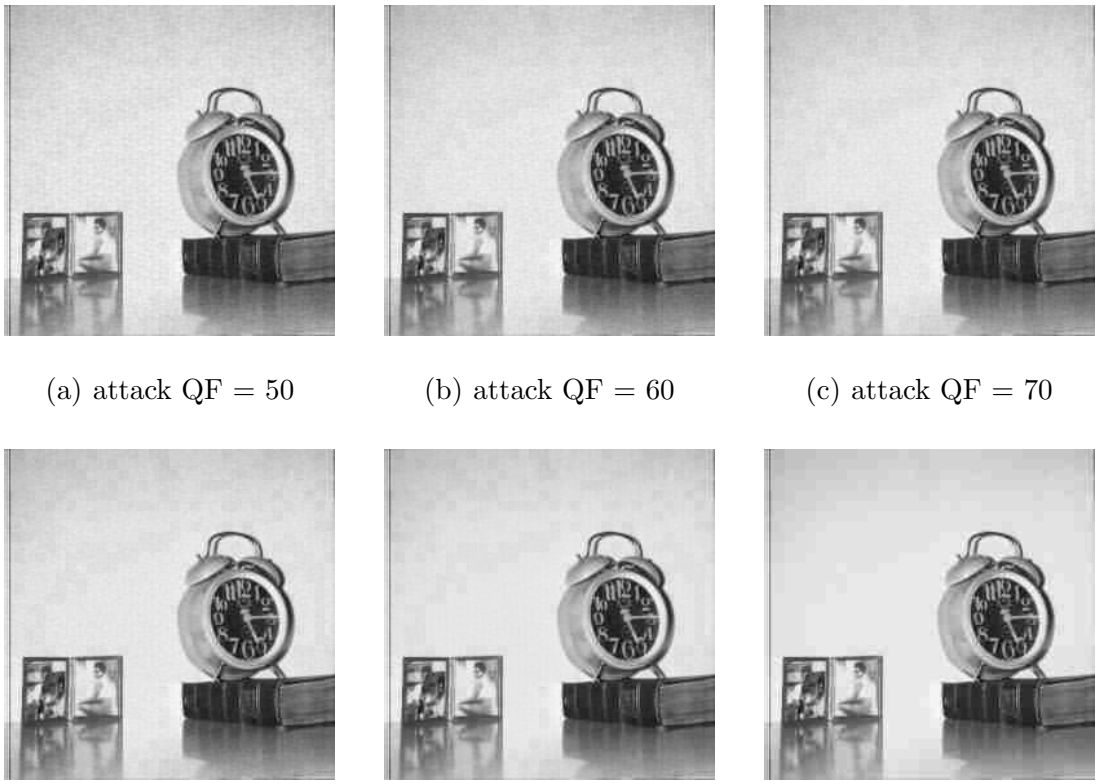
Table 4.2: Example 2: MSE per coefficients for varying levels of attacks. A 256×256 clock image has been hidden in a 512×512 bridge image.

Attk. QF	50	60	70	80	90
compr.	84.2%	81.9%	78.3%	72.5%	60.0%
MSE/coeff.	0.0335	0.0374	0.0266	0.0146	0.0046

Example 3: A 256×256 Lenna image is hidden with a design QF of 50 into a 512x512 Bridge image. Figure 4.10 shows the recovered signature images when the composite image undergoes varying levels of JPEG compression attacks. Table 4.3 shows the corresponding MSE of the received image. The signature image is JPEG compressed at QF=12, and residues of 12 low frequency coefficients constitute the analog part. 3 coefficients per host block are used for sending analog residue and another 32 coefficients form the candidate embedding band for the digital data.

Table 4.3: Example 3: MSE per coefficients for varying levels of attacks. A 256×256 Lenna image has been hidden in a 512×512 Bridge image.

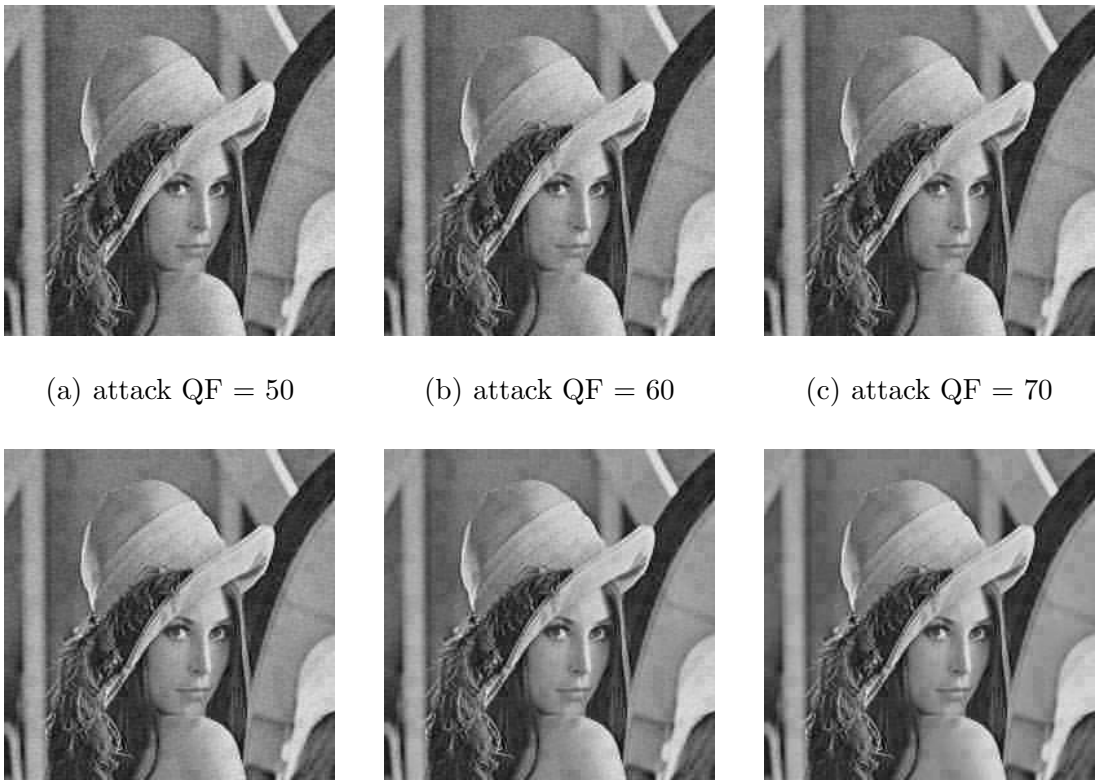
Attk. QF	50	60	70	80	90
compr.	84.3%	81.9%	78.3%	72.5%	60.00%
MSE/coeff.	0.0267	0.0371	0.0254	0.0140	0.0046



(a) attack QF = 50 (b) attack QF = 60 (c) attack QF = 70

(d) attack QF = 80 (e) attack QF = 90 (f) No attack

Figure 4.9: Example 2: Hiding a 256×256 clock image into a 512×512 bridge image (not shown here). The signature images received after various levels of JPEG compression are shown. The corresponding MSE per coefficient is shown in Table 4.2



(a) attack QF = 50 (b) attack QF = 60 (c) attack QF = 70

(d) attack QF = 80 (e) attack QF = 90 (f) No attack

Figure 4.10: Example 3: Hiding a 256×256 Lenna image into a 512×512 Bridge image (not shown here). The signature images received after various levels of JPEG compression are shown. The corresponding MSE per coefficient is shown in Table 4.3

4.6 Summary

In this chapter, we proposed a simple joint source-channel coding framework for achieving graceful improvement when hiding a media signature signal. This is practically demonstrated by a hybrid digital-analog scheme for image-in-image hiding. As the JPEG attack quality factor increases, we recover the signature image with better quality. It should be noted that, with appropriate design, the framework can be applied for any media signature or host signals. While the results show improvement over a purely digital hiding strategy, much more further work remains in exploring the huge space of possible joint source-channel coding strategies.

We have discussed high-volume embedding schemes so far, which achieve robustness against distortion constrained attacks such as compression and additive noise. In the next chapter, we focus on more robust techniques that can survive several attacks including printing-and-scanning.

Chapter 5

Print-Scan Resilient Hiding

The advent of digital age with the internet revolution has made it extremely convenient for users to access, create, manipulate, copy, or exchange multimedia data. This has created an urgent need for protecting intellectual property in both the digital and the print media. Digital watermarking is a technology being developed, in which, copyright information is embedded imperceptibly into the host in a way that is robust to a variety of intentional or unintentional attacks. The ease with which images can be converted from print to digital form and vice versa makes it necessary that the embedded digital watermark is resilient to the print and scan operation.

Strong deterrents against forgery of important documents, such as passports, driving licenses, and ID cards need to be developed at this time, when the concerns over security are higher than ever before. Print-scan resilient data hiding provides a viable solution to this problem: security information (such as fingerprints, signature, or passport number) can be imperceptibly embedded into a

picture in the document. Only specific devices, which have access to a secret key, can decode and authenticate the hidden information. Forgery of such documents become extremely difficult because the embedded data is inseparable from the picture.

Another potential application of print-scan resilient hiding is in protecting thousands of pictures that appear on magazines and newspapers everyday. With availability of inexpensive high resolution scanners, the image can be conveniently converted into a digital form and the ownership of the image may be claimed by someone else. To counter this, information can be hidden into these images before they are printed and the ownership can be verified in the digital format. A visible watermark would not be helpful in this case because it can be easily removed using any image processing software.

5.1 Introduction

In this chapter, we present methods for hiding information into images in a manner that is robust to printing and scanning. The proposed methods are *blind*, i.e., the original image is not required at the decoder to recover the embedded data. Using these techniques, several hundred information bits can be embedded into images with perfect recovery after the print-scan operation, which is a significant improvement over the state of the art. An important contribution of this chapter is a systematic analytical modeling of the print-scan process by breaking it down into simpler sub-processes, which is appropriately complemented by extensive practical experiments. The analytical and experimental findings form the basis of

the proposed embedding schemes, in which data is hidden in dynamically chosen transform coefficients, with synchronization and error correction using powerful turbo-like channel codes. Also proposed is a novel approach for estimating the rotation that an image might undergo during the scanning process, by exploiting knowledge of the digital halftoning scheme employed by the printer.

There has been a growing interest among researchers in the area of print-scan resilient embedding, but little progress has been made because of the complex nature of the problem. One of the first approaches was by Lin and Chang [61], who model the print-scan process by considering the pixel value and geometric distortions separately. There are some watermarking methods [93, 105, 10] that were not specifically designed for the print-scan attack, but they do report robustness against the print-scan operation under specified experimental setup. Ruanaidh and Pun [93] propose a watermarking method based on log-polar map of discrete Fourier transform (DFT) magnitudes (i.e., the Fourier-Mellin or FM transform). Lin and Chang's approach [61] also uses the FM transform to hide information. Technique proposed in [105] involves DFT magnitudes as well, but the watermark itself is made circularly symmetric so that the log-polar coordinate transformation is not required. Bas et al [10] use geometrically invariant feature points to embed the watermark. A few approaches focus on hiding in halftone images [92, 42], wherein, the halftone cells of the host image are shifted based on the data to be hidden, and a composite halftone image is given out directly. More recent related works include Voloshynovskiy et al [125], and Mikkilineni et al [70], who focus on document security in general rather than specifically considering printing and scanning of digital images.

Most of the above methods embed only a single bit (or a few bits) of information, as they assume the availability of the watermark sequence at the decoder. In our recent work on print-scan resilient hiding [110, 111, 106], an improvement over these methods is achieved in terms of volume of embedding. We are able to hide several hundred bits into images against the print-scan attack with blind decoding. We propose a model for the print-scan process which is comprised of three main components: geometric transformations, non-linear effects, and colored noise. We infer from the model that data must be embedded into high magnitude coefficients in a band of low frequencies. This is also found to be true in a series of practical experiments done to understand the effect of the print-scan process.

Two methods for hiding information resilient to print-scan operation are proposed. The first technique, named *selective embedding in low frequencies* (SELF), hides data in the magnitude of dynamically selected low-frequency DFT coefficients. This is in contrast to previous DFT-magnitude based approaches (e.g., [61, 105]), in which a predefined set of mid frequency coefficients are used for embedding. The second method is for hiding data in the phase spectrum of the host image. In this technique, data is embedded by quantizing the difference in phase of adjacent frequency locations. The method is accordingly termed *differential quantization index modulation* (DQIM), drawing from QIM, now-famous class of data-hiding methods proposed by Chen and Wornell [19]. Note that, because of the perceptual constraints, the volume of data hidden using the DQIM embedding in the phase spectrum is lesser than that using the SELF scheme for magnitudes.

We employ turbo-like error and erasure correcting codes in a novel fashion to

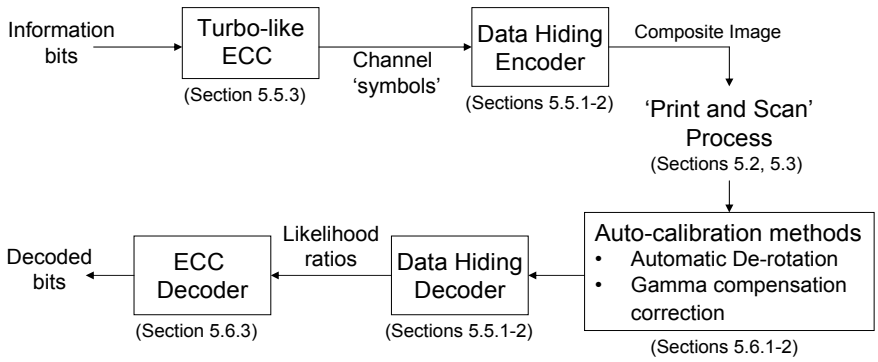


Figure 5.1: Outline of how various parts of the embedding schemes fit into the big picture. Below the block, we list the particular section(s) of the chapter that discusses it. Note, ECC stands for 'error correcting code'.

counter the synchronization problem caused due to image-adaptive hiding. This also provides robustness to the hidden data against a variety of other attacks such as those in *Stirmark* [85], e.g., heavy JPEG compression, scaling or aspect ratio change, Gaussian or median filtering, rows and/or columns removal, and to a lesser extent, random bending.

Prior to decoding, the scanned digital image is preprocessed by an automated algorithm for estimating and undoing the rotation caused by random placement of the printed image in the scanner. The method is based on the fact that laser printers use an ordered digital halftoning algorithm for printing. The employed derotation method is completely different from the previously used approaches, in which rotation invariance is typically achieved by using FM transform [93, 61]. The advantage of the proposed technique for print-scan resilient hiding is that there is no penalty in hiding rate for achieving robustness against rotation.

A big picture with the various components of our embedding techniques is provided in Figure 5.1. The figure also presents how various sections of the paper

are interconnected. The paper is organized as follows. We start, in Section 5.3, with an intuitive and analytical study of the print-scan process. Here, we lay out the three main components of the print-scan model: cropping, non-linear effects, and colored noise. We then move on to practical experiments, and list the observations made in Section 5.4. Based on the analytical and experimental findings, in Section 5.5, we propose practical methods to hide data resilient to the print-scan operation. The recovery of the embedded data is discussed in Section 5.6, where we describe a method to estimate and undo rotation undergone by the image during scanning. Numerical results are presented next (Section 5.7), followed by the concluding remarks in Section 5.8.

5.2 The Print-Scan Channel

In this section, we present a brief background of the printing and scanning process. Let us start by noting that we are dealing with two representations of an image: the digital form stored in a computer which is to be displayed on a monitor, and the analog (printed) form on a paper. Eyes are the ultimate *consumers* of pictures, and hence, directly or indirectly, the human visual system acts as a calibration for devices such as printers, scanners, monitors, and cameras. Obviously, if *perfect* printers and scanners existed, the printed picture would be *exactly* same as the one displayed on the monitor, and the problem of print-scan resilient data hiding would be very simple. In reality, however, the devices alter the image in a highly nonlinear fashion, making it extremely difficult to hide information resilient to the print-scan operation. Even then, the hope for

the data hider is that, because printers and scanners try to reproduce the image details perceptually, some image features would be preserved during the print-scan operation, where data can be embedded. In the following, we study the printing and scanning processes individually.

5.2.1 The Printing Process

When an image is printed, it undergoes a continuous-tone to bilevel conversion, known as *digital halftoning*. Digital halftoning is required because almost all printers are bilevel devices. It banks on the fact that human visual system can be coarsely approximated as a low-pass filter. Thus, the printed halftone image, which is only in black and white, would be perceived as a grayscale image when viewed from a distance. Several algorithms have evolved for digital halftoning over last decades, which can be classified into three main types: point algorithms (screening or ordered dithering, [121]), neighborhood algorithms (error diffusion, [60]), and iterative algorithms (such as direct binary search or DBS, [54]). Readers are referred to [121, 60] for an extensive discussion of various digital halftoning approaches.

5.2.2 The Scanning Process

In a scanner, the picture to be scanned is illuminated and the reflected intensity is then converted into electrical signal by a sensor, which is then digitized. Images are scanned into a computer for display on a monitor and for storage in digital media. A significant process that happens at the time of scanning is

gamma correction. Every computer monitor has an intensity to voltage response curve which is a power function with parameter γ. This means that if we send a computer monitor a message that a certain pixel should have intensity equal to $x \in (0, 1)$, it will actually display a pixel which has intensity equal to x^{γ}.

In order that the scanned image is correctly displayed on a monitor, the image data generated at the scanner is 'gamma corrected' (ie raised to a power $1/\gamma$). The correction applied at the scanner depends on the gamma of the monitor or the screen on which the image is to be displayed. The display driver in Macintosh systems apply partial monitor correction at 1/1.45. Cathode ray tube (CRT) monitors natively have gamma 2.50, and hence, the gamma of an uncalibrated Macintosh is accurately 2.5/1.45=1.72. Windows systems do not adjust the display path so the gamma-space of uncalibrated PC system is 2.50. The default compensation is placed at 2.2, a value between to two, as defined in the sRGB standard for the Internet images. Note that the scanner software usually allows users to set the gamma correction that is to be applied for an image.

5.3 Modeling the Print-Scan Process

We now present a model for the print-scan operation by breaking it down into simpler sub-processes and study how they distort the image when it is printed and scanned. We know from the watermarking literature that, for robust embedding, data must be hidden in the transform domain. Therefore, in our model, we specifically analyze the effect of the print-scan process on the DFT coefficients. Before proceeding with a detailed study, let us briefly list the most interesting

findings of this section.

1. *Frequency bands*: Most components of our print-scan model tend to affect high frequency coefficients more than the low and mid frequency ones.

2. *Effect on DFT magnitude spectrum*: High magnitude DFT coefficients are preserved better than the low magnitude ones.

3. *Effect on DFT phase spectrum*: The difference in phase of adjacent frequency locations is preserved during the print-scan operation (for the high magnitude coefficients).

Printing followed by scanning involves conversion of from digital to analog, and back to digital form. This is inherently a very complex process. The problem is compounded by the fact that a variety of printing and scanning devices are available in the market, which work on one of many different existing technologies. Obviously, constructing a unified model will be extremely difficult, if not impossible. Hence, we limit ourselves to laser printers and flatbed scanners.

However, even when only laser printers and flatbed scanners are considered, constructing a complete or near complete model would require so many parameters that the resulting model will no longer remain very useful practically. Instead, we just aim to dissect the print-scan process into simper sub-processes. We hope that analyzing these sub-processes would then inspire the construction of embedding schemes that survive the print-scan process.

There have been a few approaches that discuss individual models for printers and scanners. Several models for laser printers that aid the design of digital

halftoning methods have been proposed (for example, [54], and [126]). In [54], a model for the electrophotographic (EP) process (the technology employed by laser printers) has been proposed, in which various steps involved in the EP process are analyzed mathematically. This model is then used to design an iterative halftoning method, called direct binary search (DBS). In [126], a physical model is used to train a signal processing model for the printer, which can then be used for halftoning techniques. There have been a few efforts in modeling the scanner as well (e.g., [101, 104]). In [101], the goal is to calibrate the scanner without using calibration targets. Scanner modeling using specifically designed test targets was done in [104]. In this study, the aim was to perform efficient optical character recognition (OCR).

The only prior work, that we know of, in modeling the print-scan process as a whole is by Lin and Chang [61]. In this work, the authors separate the print-scan distortions into two categories: pixel value and geometric. The model proposed for the pixel value distortion involves a number of parameters, which must be determined experimentally. Due to this reason, it may be difficult to deploy this model practically. In our approach, instead of detailed modeling of the print-scan operation as a whole, we divide it into simpler sub-processes, and specifically study the *bottlenecks* components in detail, the ones that induce the greatest distortion.

Let us now walk through the kinds of distortions an image undergoes when it is printed and scanned, as outlined in Figure 5.2. At the beginning, we have a digital image stored on the computer in which data is to be hidden. The image, which may come from one of many possible capturing devices (such as a scanner,

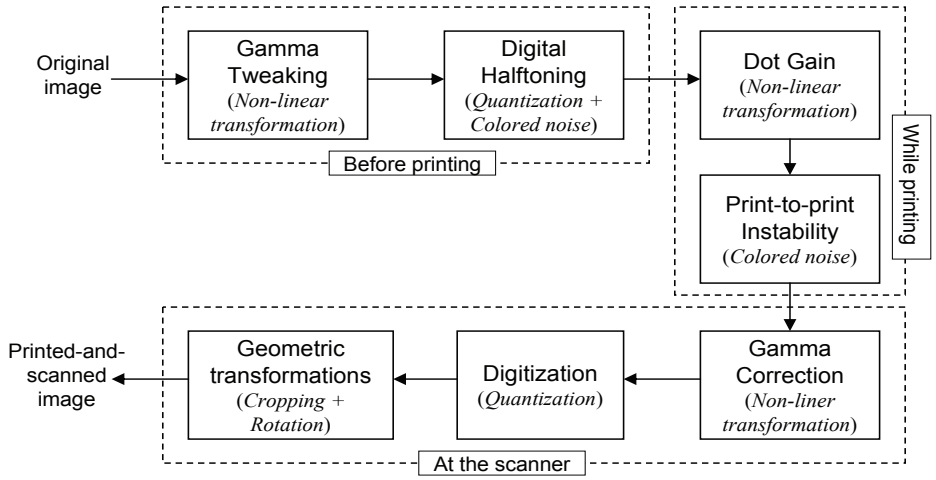


Figure 5.2: Various processes that distort the image when it undergoes printing followed by scanning.

a digital camera, or a video camera), should have been *gamma adjusted* when (or after) it was generated to make sure it looks fine when viewed on a monitor. In the following, we briefly describe how the various blocks of Figure 5.2 distort this image.

- **Gamma tweaking**: In order to make sure the printed images appear the same as on a monitor, many printer vendors change the transfer characteristics of the printer to resemble that of an uncalibrated monitor. This adjustment, called *gamma tweaking*, is the first non-linear transformation that an image undergoes during the print-scan process.

- **Digital halftoning**: The image is converted to a digital halftone before it is printed. Halftoning algorithms essentially quantize the image into a binary one. The halftoning methods tend to put the quantization noise into high

frequency spectrum of the image, which is a source of colored *high-frequency noise* that gets added to the image.

- **Dot gain**: The digital halftone of the image is then printed dot-by-dot on a paper. When it is printed, the image suffers from a phenomenon called *dot gain*: the images tend to appear darker than expected due to several reasons (such as spreading of the colorant on the medium, and also optical or electrostatic causes). Dot gain is a non-linear transformation, but it can be roughly approximated by a piecewise-linear curve. Many digital halftoning algorithms incorporate a model for the printer dot gain in their design.

- **Print-to-print instability**: Uncertainties during the printing process can lead to correlated noise. An example of print-to-print instability is *banding*, which stands for horizontal imperfections appearing in the printouts.

- **Scanner gamma compensation**: When the image is scanned, it must be compensated to make sure it looks fine to us when viewed on a monitor. The scanned image pixel values are raised to a power of $1/\gamma$, where γ is the assumed system gamma of the monitor on which the image is to be viewed.

- **Digitization**: The scanned image must be digitized before storing, which invariably leads to quantization errors. Since it follows non-linear adjustment of the previous step, the effect of quantization noise may get amplified.

- **Geometric Transformations**: At the time of scanning, the image can be subjected to a number of geometric transformations, such as cropping, rotation, and scaling. These effects must be explicitly taken into account

because, even with most careful scanning procedure, one cannot completely avoid such geometric transformations.

In the above discussion, we have identified, roughly, various processes that distort the image when it is printed and scanned. The extent to which each of these processes affect the image would depend on the particular devices and the settings used while printing and scanning. Thus, we can now model the print-scan processes by analyzing these individual sub-processes for some specific printers and scanners. However, in addition to the complexity issues, a detailed model just for some particular devices would not be very useful. What we would like to do instead is to understand the bottleneck processes in detail and apply the findings to build resilient embedding schemes. Hence, we simplify our study by grouping *similar* processes together, and divide the distortions into three broad categories: geometric transformations, non-linear affects, and colored noise. Cropping (in combination with scaling) and rotation are the major geometric distortions that an image undergoes during print-scan process. There are several sources of non-linear effects, such as gamma tweaking, dot gain, and scanner gamma compensation. Colored noise gets added to the image as a result of digital halftoning and print-to-print instability.

We now describe the individual components of our model in more detail (Sections 5.3.1 - 5.3.3). As stated before, rotation and cropping are the main geometric distortions that an image undergoes during scanning. Since we have a method to estimate and undo rotation (to be discussed in Section 5.6), we do not consider rotation for a detailed study here. In the following, we study the effects of image cropping.

5.3.1 Cropping

Some mild cropping is inevitable during the scanning process, when the image is cropped from the background either manually or automatically. As a result, the effects due to cropping cannot be ignored in the design of a print-scan resilient hiding method. One more point to note is that, in general, it is very difficult to achieve perfect registration between the original and the attacked image due to presence of cropping, however mild. When the images are analyzed, this imperfect registration might be the reason for the observation of higher noise near the edges within the image (as in [61]). Instead of specifically modeling this noise, we find it more appropriate to consider cropping separately, and not worry about the registration issue.

Cropping can be thought of as a multiplication of a masking rectangle with the image. In the frequency domain, this is equivalent to convolution of a two dimensional sinc-like function with the spectrum of the image. This causes blurring of the image spectrum. The blurring would significantly affect low-magnitude coefficients whose neighboring coefficients are of a higher magnitude.

Consider an image $f(n_1, n_2)$ with N_1 rows and N_2 columns, so that it is defined over the domain $\Omega = \{0, 1, ..., N_1 - 1\} \times \{0, 1, ..., N_2 - 1\}$. Cropping of the image can be thought of as a multiplication with a masking window. Assuming that the image is cropped to new dimensions of $M_1 \times M_2$ (with $M_1 \leq N_1$, and $M_2 \leq N_2$), the masking window $r(n_1, n_2)$, also defined over Ω, can be written as,

$$r(n_1, n_2) = \begin{cases} 1 & \text{if } M_{1a} \leq n_1 < M_{1b}, \text{ and } M_{2a} \leq n_2 < M_{2b}, \\ 0 & \text{otherwise.} \end{cases}$$

Here M_{1a} and M_{1b} define the top and bottom cropping locations respectively, so

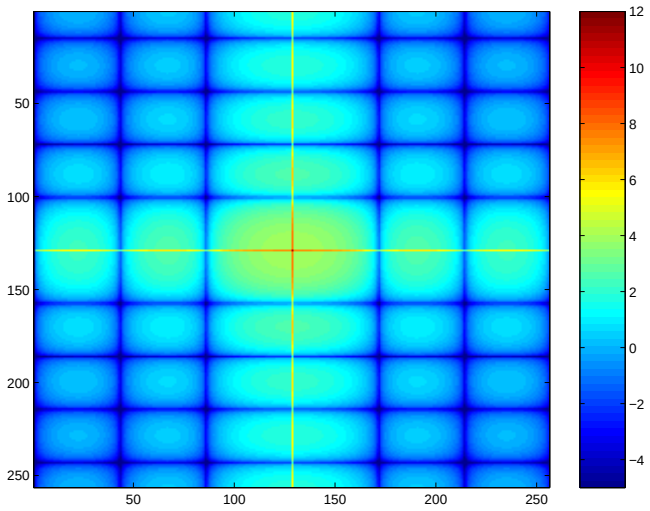


Figure 5.3: Mild Cropping: Natural logarithm of the magnitude spectrum of the mask, $r(n_1, n_2)$. The size of image is $N_1 = N_2 = 256$, and the cropping window size is $M_1 = 248$, and $M_2 = 250$. Notice that most of the energy is concentrated on the $(0, 0)$ or the DC coefficient. Note that the numbers shown here do not include the $1/N_1N_2$ scaling in computing the DFT.

that $M_1 = M_{1b} - M_{1a}$. Likewise, $M_2 = M_{2b} - M_{2a}$. We can now define the cropped image $c(n_1, n_2)$ as,

$$c(n_1, n_2) = f(n_1, n_2) \times r(n_1, n_2) \qquad \forall\{n_1, n_2\} \in \Omega$$

This product is equivalent to circular convolution in the DFT domain. Defining $F(k_1, k_2)$, $R(k_1, k_2)$, and $C(k_1, k_2)$ as the 2D DFT of $f(n_1, n_2)$, $r(n_1, n_2)$, and $c(n_1, n_2)$ respectively, the circular convolution can be written as,

$$C(k_1, k_2) = \sum_{l_1=0}^{N_1-1} \sum_{l_2=0}^{N_2-1} F(l_1, l_2) \cdot R(\langle k_1 - l_1 \rangle_{N_1}, \langle k_2 - l_2 \rangle_{N_2}) \tag{5.1}$$

Here, $\langle \cdot \rangle_N$ denotes the modulo N operator. The DFT of the masking window $r(n_1, n_2)$ would be a sinc-like function[1], with its shape being a function of M_1

[1]Note that $R(k_1, k_2)$ is a discrete function, which does not strictly follow the sinc definition.

and M_2, and a phase shift that depends on the location of the masking window, i.e., M_{1a} and M_{2a}. When the cropping is mild, $N_1 - M_1$ and $N_2 - M_2$ are small, and the sinc-like function would be quite *narrow*. Figure 5.3 shows the 'fftshifted' magnitude spectrum of an example masking window. For mild cropping, most of the energy of $R(k_1, k_2)$ is concentrated on the $\{0, 0\}$ coefficient, along with low frequency part of the first row and first column. Thus, the blurring of the original image spectrum will be mild for those DFT coefficients whose magnitude is high or of the same order as its neighbors. However, for coefficients whose magnitude is significantly lower than its neighbors, the blurring will cause its magnitude to increase. This will affect low magnitude coefficients in all the frequency bands - high, mid, or low. This is the first significant inference regarding the effect of print-scan process on the DFT coefficients: the high-magnitude coefficients are better suited for embedding information as compared to the low-magnitude ones.

Let us continue focusing on mild cropping, and investigate its effect on the magnitude and phase of the DFT coefficients. It should be noted that the cropping window, $r(n_1, n_2)$, is not known to the decoder, and hence, we cannot simply use deconvolution to estimate the original DFT coefficients. However, under the assumption of mild cropping, and considering only those coefficients that do not have significantly lower magnitude than their neighbors, we can write the convolution expression (5.1) with only two dominant terms.

$$C(l_1, l_2) = R(0, 0) \cdot F(l_1, l_2) + R(l_1, l_2) \cdot F(0, 0) + \text{other terms} \qquad (5.2)$$

Once the size of the masking window is fixed (i.e., M_1, M_2 fixed), the magnitude of $R(k_1, k_2)$ does not change with the actual location of the masking window

It still has a shape similar to the sinc function (Figure 5.3), and hence we call it *sinc-like*.

(determined by M_{1a}, M_{2a}, M_{1b} and M_{2b}). Furthermore, the blurring caused by mild cropping is not significant for high magnitude coefficients. In summary, for the magnitude spectrum, the contribution from all terms in (5.2) other than the first one would be small and also, the variation in exact location of the masking window would not make significant difference to high magnitude coefficients. This leads to the second important inference: embedding data directly into the magnitudes would work.

The phase of $R(k_1, k_2)$ would vary as the location of the masking window (i.e., M_{1a}, M_{2a}) changes. Looking at the phase shift between the original image spectrum, $F(k_1, k_2)$, and the scanned image spectrum, $C(k_1, k_2)$, from (5.2), we see that the first term does not cause a phase shift, but the second term does. The amount of shift depends on the phase of $R(l_1, l_2)$, which, as discussed above, varies with the location of the masking window, but is fixed for a particular instance of the cropped image. Also, since the phase of R varies slowly, the shift seen by nearby frequency locations is approximately the same. Thus, for the phase spectrum, there is an unknown phase shift between corresponding original and cropped image DFT coefficients, which varies slowly across the spectrum for mild cropping. This unknown shift can be canceled by taking difference in phase of adjacent frequency locations. This leads to another inference: data embedding in the phase difference of adjacent DFT coefficients might work.

5.3.2 Non-linear Effects

The main sources of non-linear effects during the print-scan process are gamma tweaking, dot gain, and gamma compensation. While gamma tweaking and dot

gain occur at the printer, gamma compensation occurs at the scanner. The final effect we see between the original and the scanned image is actually a combination of these three non-linear transformations happening at different stages. Things are worsened by the fact that these non-linear transformations are followed by quantization of some sort, which amplifies the affect of quantization noise. As seen before, while gamma tweaking is followed by quantization due to digital halftoning, dot gain and scanner gamma compensation are followed by digitization at that scanner.

We conducted experiments to understand the effect of non-linear transformations on the DFT coefficients. It was observed that these non-linear transformations affect the mid and high-frequency coefficients more than the low frequency ones. Further, we see that the in the low frequency band, only the coefficients with low magnitude were affected. This leads to another inference regarding the print-scan process, that, low frequency coefficients are more suited for data embedding than the high frequency ones.

If the devices are under control of the data hider, using profiles to calibrate the devices would reduce the distortion due to non-linear transformations. Under controlled conditions, the non-linear effects can be modeled more precisely, and an embedding scheme can be designed that can survive these transformations. In this chapter, however, we do not attempt to do this. Constructing an embedding method with a higher capacity and resilience to print-and-scan operation for some specific class of devices (having known characteristics) would be an interesting avenue of future work. We believe that many security related applications would fit this scenario.

In applications such as copyright protection and e-commerce of digital images, we must assume that the devices are not under our control. In such cases, one must deal with non-linear affects that are varied, and the design must be conservative so as to survive heavy non-linear transformations. Sometimes the devices are only partially under designer's control. This would be the case when one must work with commercially available devices. The hardware vendors usually give only a partial control on the devices to the users. For example, most printer driver software do not provide any way to get around gamma tweaking.

Dealing with non-linearity would require us to calibrate the devices, and/or learn the transfer characteristics experimentally. We do not take this up in the current work mainly because the non-linear effects are not a significant impairments for low-frequency coefficients. We do, however, present a practical way to get around incorrect gamma compensation happening at the scanner. The technique, described in Section 5.6.2, can be employed to correct any discrepancy in scanner gamma compensation which may happen when the devices are not calibrated.

5.3.3 Colored Noise

Before an image is printed, it is converted into a digital halftone. Digital halftoning algorithms tend to put the quantization noise in high frequencies [121] since the human visual system is not very sensitive to high-frequency noise. This introduces high-frequency noise into the image. Another source of colored noise is the printing process itself. Uncertainties during the printing operation, or print-to-print instability, adds correlated noise which varies every time a printout is

taken.

Addition of the colored noise due to halftoning heavily affects the high frequency DFT coefficients, so that these coefficients cannot be used for data embedding. The effect of this component of our model, however, is mostly limited to high and mid frequency coefficients. Since this component of our model does not significantly affect low frequency bands, we do not analyze this component in more detail here.

Inverse halftoning (see, for example, [67],[138]), to some extent, can allow us to reduce the affect of colored noise coming from the halftoning process. This may cause slight blurring of the image. Thus, using inverse halftoning, we may be able to embed in a larger frequency band and possibly improve the volume of embedding. Leveraging the inverse halftoning literature to mitigate the effect of colored noise, and hence embed at a higher capacity, is an interesting avenue of future work. This, however, is out of scope of the this chapter.

5.3.4 Discussion on Modeling Issues

Of all the three components of our print-scan model, only cropping contributes to distortion in all the frequency bands equally. The other two components tend to affect mid and high frequency coefficients more than the low frequency ones. This makes low frequency coefficients more suitable for data embedding. In their model, Lin and Chang [61] also consider cropping to be an important factor. They view it as an additional source of noise. Moulin and Briassouli [75] consider cropping as well, although not in the context of print-scan. Similar to our observation, they view cropping as causing blurring in the frequency domain.

In the print-scan model proposed by Lin and Chang [61], low-pass filtering (or blurring) of the image has been considered via a couple of point spread functions. Voloshynovskiy et al [125] also view the printing process as causing blurring of the image. Here, authors specifically consider error diffusion halftoning method, which has been modeled as combination of two filters [57]. It should be noted that in [61] too, authors use inkjet printers in their experiments, which typically employ error diffusion halftoning. Since our focus in this work is on laser printers rather than inkjet printers, we do not consider image blurring in our current proposal of the print-scan model. Also, in the printing scenario we consider, the images are printed at high resolutions. For example, a 512×512 image is printed at 600 dpi printer resolution on an *letter* paper with 72 pixels per inch (ppi), so that the size of the image on the paper is 7.11"×7.11". In this case, it turns out that on an average, a block equivalent to 8.33×8.33 printer dots is used for every pixel of the image. At this resolution, the image does not get significantly blurred during the printing process.

Having studied the print-scan operation from an analytical perspective, we now move on to practical experiments in the following section.

5.4 Experiments

We conducted a series of experiments involving printing and scanning of a number of images in order to practically understand the effect of print-scan process on the transform coefficients, and to determine invariants, in which data could be embedded.

The devices involved in this scenario, the printer, the scanner, and the monitor, must be calibrated before use to provide the best results because without calibration, we cannot trust the color or intensity produced by these devices. There is huge body of literature available on the world wide web and elsewhere on how to calibrate these devices (e.g., the International Color Consortium or ICC profiles). However, we note that most of the devices used by common users are uncalibrated. Hence, to mimic a real world scenario, we do not explicitly calibrate the devices we use in our experiments. Also note that, for simplicity, we limit ourselves to grayscale images. Below we describe our experimental setting followed by the observations that were made.

Several images were printed and scanned using commercially available laser printers and flatbed scanners[2]. The images were printed at resolutions varying from 300 to 1200 dpi. In the typical printing scenario, 512×512 images were printed with 72 pixels per inch (ppi) setting on letter papers, so that the size of the image on the paper is 7.11"×7.11". Widely used Xerox recycled papers (for copiers and laser/inkjet printers) were used for printing. At the time of scanning, the images were cropped and resized using bicubic interpolation to their original size. The resolutions typically used for scanning were 300 to 1200 spi.

Various parameters (such as printer and scanner resolutions, scanner gamma correction, and print image size) were varied and its effect on several image features were studied in order to find features that are invariant to the print-scan operation. No effort was made to explicitly register the scanned and original image or their features in the experiments because our goal is to build a blind

[2]Laser printers used in our experiments: Lexmark Optra S 1620, Sharp, HP, and HP . Scanner used: CanoScan N670U flatbed scanner.

system where the original image would not be available at the decoder. The DFT coefficients were identified for a more detailed study.

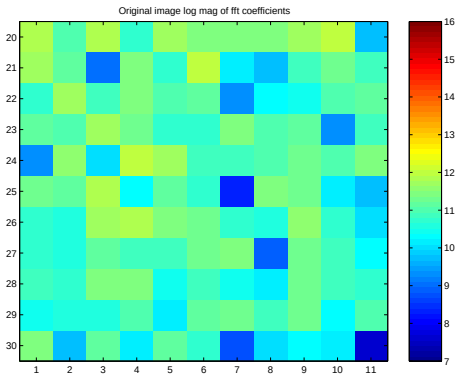
5.4.1 Effect on DFT Magnitudes

Below are the experimental observations for the effect of printing followed by scanning on the DFT coefficient magnitudes. Note that, unless otherwise stated, we refer to natural logarithm of the DFT coefficient magnitudes in the following.

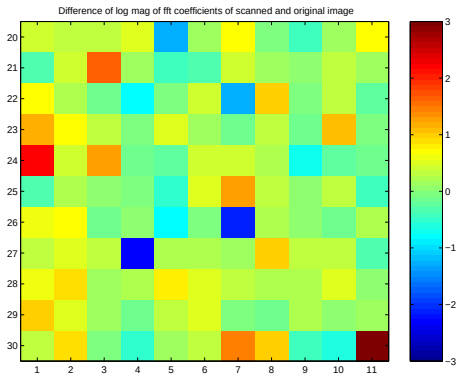
1. The low and mid frequency coefficients are preserved much better than the high frequency ones. In general, the lower the frequency, the better its chances of surviving the print-scan process.

2. In the low and mid frequency bands, the coefficients with low magnitudes get washed out, while those with high magnitudes are preserved much better. It can be seen from Figure 5.4 that the coefficients with low magnitudes are hit more severely than their neighbors with higher magnitudes. This is a significant characteristic of the channel and has been observed consistently for different images and various printer or scanner resolutions.

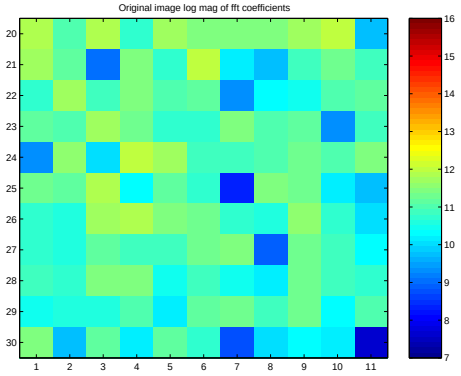
3. Coefficients with higher magnitudes (which do not get severely corrupted) see a gain of roughly unity (with the default gamma correction). Roughly speaking, if the print-scan operation is approximated as a linear filter (for large enough coefficients and low enough frequencies), then the magnitude gain is unity after application of gamma correction. One possible explanation is that the printing operation in itself does not cause blurring, since several printer dots are dedicated to each pixel of a printed image.



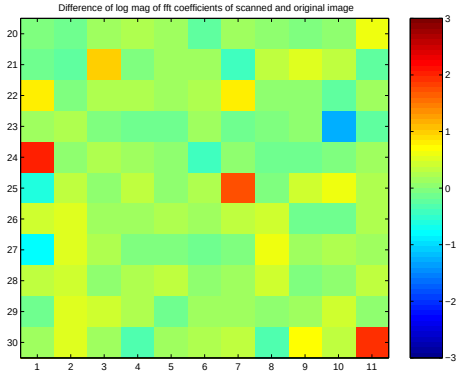
(a) Original image spectrum in log domain



(b) Difference in log DFT magnitudes of scanned and original image



(c) Original image spectrum in log domain



(d) Another instance of scanned image: diff. in log DFT magnitudes

Figure 5.4: Print-scan channel: Almost all *dark blue* coefficients in the original image magnitude spectrum of (a) and (c) correspond to *dark red* points in the log transfer function of (b) and (d), e.g., (24,1),(25,7),(30,11), and so on. It indicates that the error is high for all coefficients that have low magnitudes. Note that the image in (d) has been printed and scanned with higher resolutions than the one in (b).

4. Slight modifications to the selected high magnitude low frequency coefficients does not cause significant perceptual distortion to the image.

5.4.2 Effect on Phase Spectrum

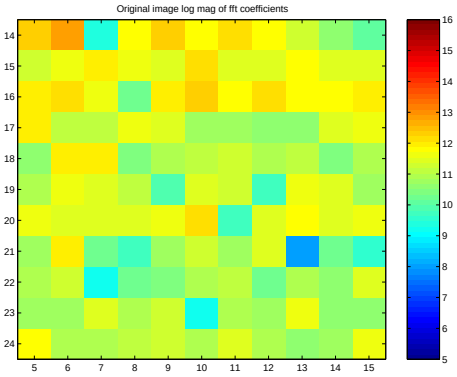
Our analysis of the model for the print-scan process (in Section 5.3.1) suggest that the difference in phase of adjacent frequency locations would be preserved during the print-scan process. Here we practically investigate the effect on phase difference of neighboring frequency locations. Following are the observations made.

1. The phase difference for the high frequency locations see a very high noise.

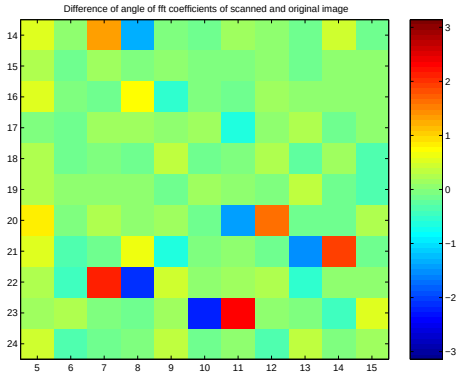
2. For the low frequency coefficients, the phase difference of adjacent locations is preserved for coefficients whose magnitude is high. Figure 5.5 shows the difference in the phase difference for original and scanned images for two different instances of printed-and-scanned image. It is observed that phase difference for coefficients with lower magnitude are severely corrupted. Note that since we are taking difference of two frequency coefficients, as seen in the figure, a high error in one gets carried to the next location as well.

5.4.3 Experimental Observations and the Print-Scan Model

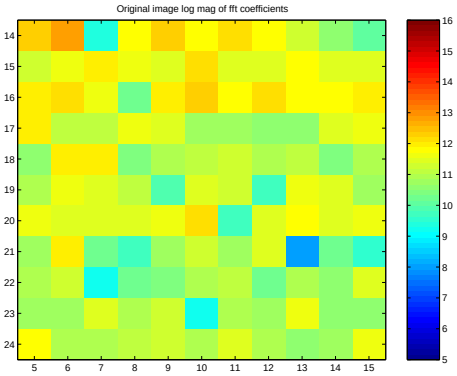
We conclude this section by noting that the experimental observations of this section are quite consistent with the analytical inferences made from the model. Our investigation of colored noise and non-linear effects suggests that high frequency coefficients are not good for embedding data, which indeed turns out



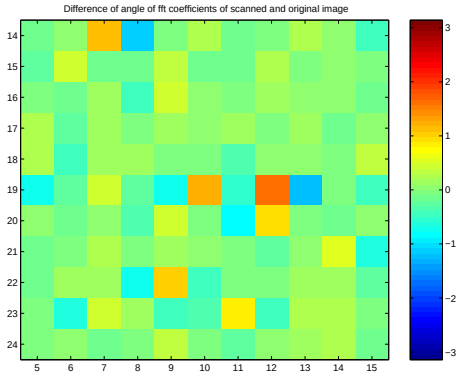
(a) Original image spectrum in log domain



(b) Difference in the difference of phase of adjacent frequency locations for scanned and original image.



(c) Original image spectrum in log domain



(d) Another instance of scanned image: Difference in the difference of phase of adjacent frequency locations.

Figure 5.5: Effect on phase spectrum during print-scan: The phase difference of adjacent frequency locations is preserved except for those coefficients whose magnitude is lower than their neighbors, e.g., (14,7), (22,7), (23,10), and so on. The exact effect also varies for different instances of scanned images.

to be the case practically. In the experiments, we observe that low magnitude coefficients are affected much more than their high magnitude neighbors, a phenomenon that was also predicted by our analysis of the effect of cropping. For the phase spectrum, the analysis suggested that difference of adjacent frequencies is likely to be preserved, which, again, is observed practically as well. Based on all these findings, we now propose practical print-scan resilient embedding methods in the following section.

5.5 Print-Scan Resilient Embedding

Before discussing the embedding schemes in detail, let us first re-visit the big picture provided in Figure 5.1. We now redraw the block diagram with more specific details of the employed embedding mechanism in Figure 5.6. The system is divided into three main *layers*: auto-calibration at the receiver, data hiding layer, and the coding framework. We study these layers as we proceed in the paper. In the rest of this section, we discuss the hiding methods and the coding framework.

Two practical embedding schemes are proposed. The first is the selective embedding in low frequencies (SELF) scheme that embeds data into the magnitude spectrum of the host image, and the second is differential quantization index modulation (DQIM) method for hiding in the phase spectrum. We now describe these methods in detail next, followed by a coding framework employed to counter synchronization problem caused due to image-adaptive hiding.

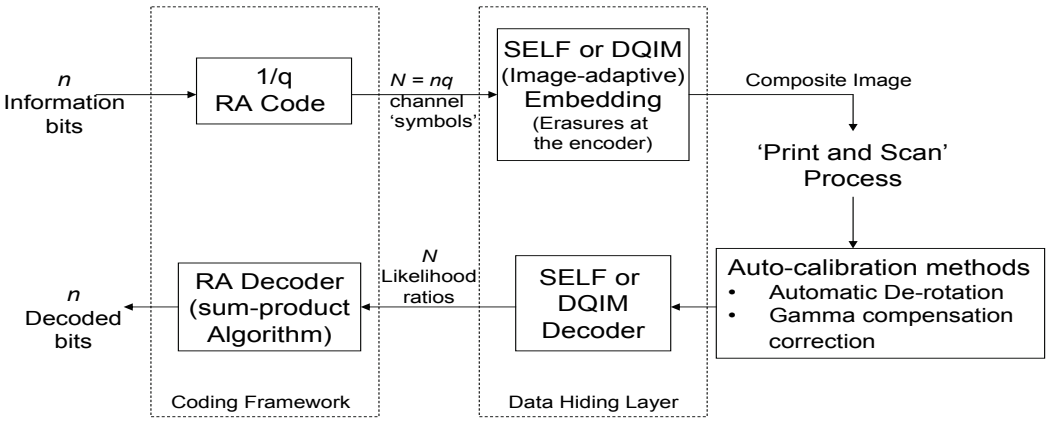


Figure 5.6: An overview of how various parts of the embedding schemes fit into the overall system.

5.5.1 SELF:Selective Embedding in Low Frequencies

Based on the experimental and analytical modeling of the print-scan process described in the previous sections, we propose an image-adaptive hiding method that achieves robustness against the print-scan operation. The model as well as the empirical observations suggest two ideas: embed in low frequency coefficients, and avoid hiding in low magnitude coefficients. With this in mind, we propose a hiding method, in which information is hidden into dynamically selected high-magnitude low-frequency coefficients. Hence the name: selective embedding in low frequencies (SELF).

Figure 5.7 shows a block diagram of the SELF embedding methodology. Consider an N×N host image in which data is to be hidden. Let us denote the natural logarithm of the magnitudes of 2D DFT of the whole image by c_{ij}, $0 \leq i, j \leq N-1$. We embed in a given coefficient c_{ij} only if it lies in a predetermined frequency band and also exceeds a threshold t_{ij}. Let us define the band as an indicator

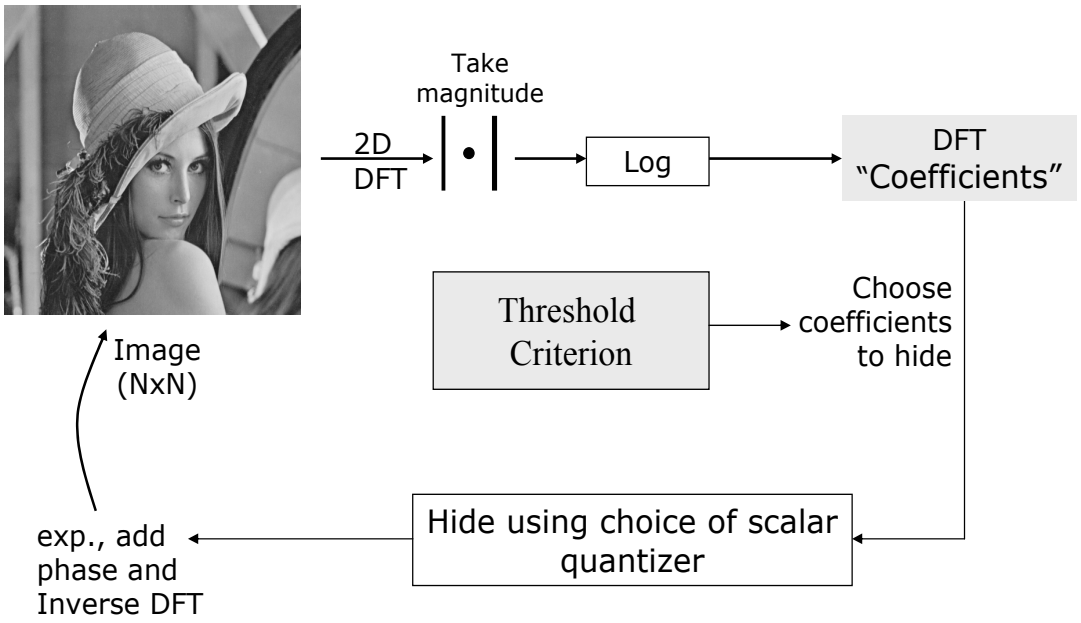


Figure 5.7: Hiding methodology for the SELF scheme.

function b_{ij}, such that if $b_{ij} = 1$, the coefficient c_{ij} lies in the band. Note that b_{ij}, t_{ij} and the quantization interval Δ are design parameters that are shared between the encoder and the decoder. Embedding is done using choice of scalar quantizers. We send either $Q_1(c_{ij})$ or $Q_0(c_{ij})$ depending on the bit to be hidden. Thus, the modified coefficient, d_{ij} can be given as

$$d_{ij} = \begin{cases} Q_{b_l}(c_{ij}) & \text{if } b_{ij} = 1, \text{ and } c_{ij} > t_{ij}, \\ c_{ij} & \text{otherwise.} \end{cases} \tag{5.3}$$

Also note that symmetry of the DFT coefficients is maintained during the hiding process by modifying two symmetric coefficients in the same manner so that the inverse DFT gives real values. Finally, taking exponential, adding phase, and taking inverse Fourier transform gives the hidden image intensity values.

The choice of the candidate embedding band, the threshold(s), and Δ is done

empirically through experimentation with several images. The goal is to hide as much information as possible without causing perceptual distortion to the image while maintaining a low error rate. The value for Δ we use in our experiments is 0.4 to 0.5. Using a higher value for Δ causes perceptual distortion to the image, while using a lower value increases the error rate significantly. Perceptual considerations influence the our choice of the candidate embedding band b_{ij} as well. Choosing a smaller band reduces the hiding rate but gives a good quality composite image, while using a larger band may cause greater distortion to the image. Using a larger candidate embedding band may also increase the error rate since the noise level increases as we go on to the higher frequencies.

The threshold varies with respect to the frequency band, which follows the same trend as the image spectrum itself. It is known that images have significant low frequency component, and in general, the magnitude of the coefficients decrease as we move to the higher frequencies. The coefficient threshold t_{ij} is chosen such that it also reduces with the band. A typical (example) band along with the threshold values is shown in Figure 5.8. Since we dynamically chose the embedding locations, we must deal with the synchronization problem inherent to image-adaptive hiding schemes, which we discuss later in the Section 5.5.3. Let us now move on to DQIM hiding scheme for phase spectrum.

5.5.2 Differential Quantization Index Modulation

Quantization index modulation (QIM), proposed by Chen and Wornell [19], are a class of information hiding methods, in which data is embedded into the host sample by the choice of quantizer. Here, we propose a new quantization-

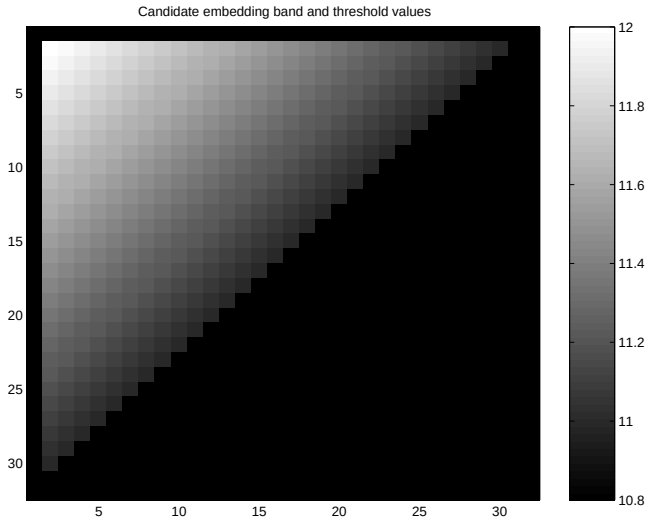


Figure 5.8: Typically used candidate embedding band and threshold values: Only one quadrant is shown here with the *black* part indicating that the coefficients are not in the band. Threshold values are shown for the coefficients that are inside the band. Notice how the threshold value decreases as we go towards higher frequencies. Note that the numbers shown here are for a 512×512 image and do not include the $1/N^2$ scaling in computing the DFT.

based method for data hiding with the goal of surviving mild cropping and the print-scan process. Instead of just quantizing the host signal, we embed data by quantizing the difference of two adjacent host samples. The idea of hiding in difference of adjacent locations is analogous to 'differential phase shift keying' (DPSK), used to combat the effect of unknown channel phase shifts in wireless communication. We employ similar nomenclature, and term the proposed method *differential quantization index modulation* (DQIM).

We use DQIM to embed information in the phase spectrum of the images to counter unknown phase shift induced due to mild cropping. As discussed in Section 5.3.1, cropping is equivalent to circular convolution of the image spectrum with a sinc-like function. This leads to a phase shift between original and scanned

image, which varies slowly across the spectrum of the image. This unknown shift can be canceled by embedding data in the difference of adjacent frequency locations. This inference has also been observed in our practical experiments (Section 5.4.2). Below we describe how embedding in the phase differences is practically implemented.

We first scan the image phase spectrum row-wise. Note that only those coefficients that lie in a predefined band are used for embedding information. Let us denote the row-wise scanned original image phase values by ϕ_n, where n is the index ($n \in \{0, 1, 2, ..., N_{max}\}$), and the quantized values by θ_n. Then, the embedding function is,

$$\theta_n = \langle Q_b(\phi_n - \theta_{n-1}) \rangle_{2\pi} \forall n \in \{1, 2, ..., N_{max}\}$$

Note that since we are dealing with phase, we must output the modulo-2π values after the quantization $Q_b(\cdot)$ of the difference is done. Also note that we use the quantized values θ_n to compute the phase difference for the next coefficient. This is done to maintain consistency for the decoder, which just finds these differences, and determines which of the two quantizers was used.

As discussed before (Section 5.3.1), the assumption of slowly varying phase shift is not valid for those coefficients whose magnitude is significantly lower than it neighbors. Hence, we avoid hiding in these locations, and use turbolike repeat-accumulate (RA) codes to counter the synchronization problem caused due to adaptive hiding, as discussed below.

5.5.3 Coding Framework for Synchronization

An erasure and error correction coding framework is used to counter the desynchronization problem caused due to the fact that the proposed methods dynamically choose the embedding locations. Readers are referred to our previous work [109], [51] for a detailed account of the coding framework, in which a local adaptive criteria was used to preserve the perceptual quality of the hidden image. Here we briefly discuss the main ingredients of the framework, and describe how it is adapted for the proposed methods.

Both the methods, the SELF hiding scheme for embedding in magnitudes, and the DQIM method for embedding in phase are image-adaptive methods, in which, the encoder selects DFT coefficients to embed based on a threshold criteria. The decoder does not have explicit knowledge of the locations where data is hidden, but employs the same criteria as the encoder to guess these locations. The distortion due to attacks may now lead to insertion errors (the decoder guessing that a coefficient has embedded data, when it actually does not) and deletion errors (the decoder guessing that a coefficient does not have embedded data, when it actually does). In principle, this can lead to desynchronization of the encoder and decoder.

An elegant solution based on erasures and errors correcting codes is provided to deal with the synchronization problem caused by the use of local adaptive criteria. The bit stream to be hidden is coded, using a low rate code, assuming that all host coefficients that lie in the candidate embedding band will actually be employed for hiding. A code symbol is *erased at the encoder* if the local adaptive criterion (i.e., the threshold criterion) for the coefficient is not met. Specifically,

we use repeat-accumulate (RA) codes [31] in our experiments because of their simplicity and near-capacity performance for erasure channels. A rate 1/q RA encoder involves q-fold repetition, pseudorandom interleaving, and accumulation of the resultant bit-stream. Decoding is performed iteratively using the sum-product algorithm [58].

Let us consider an example wherein we want to hide in a 512×512 image. The candidate embedding band is a design parameter known to both encoder and decoder. Let us assume that the band spans 1000 coefficients. Suppose we want to hide 200 bits into the image. We would use a 1/5 RA code (i.e., q = 5), which gives a codeword length of 1000. This codeword is now hidden using the adaptive criteria such that if a coefficient does not pass the threshold test, the corresponding code symbol is erased (i.e. not hidden). Note that the RA code rate and the number of bits hidden are predetermined at the design state, and are chosen in such a way that the codeword length is equal to, or slightly greater than the number of candidate embedding coefficients. When the codeword length is greater than the size of the band, the excess code symbols are erased at the encoder.

5.6 Recovery of Embedded Data

We now discuss how the embedded data is recovered and decoded. Before decoding, the scanned digital image is pre-processed by an automated algorithm to estimate and undo rotation. In the following, we describe this approach. Next, we present a method to deal with incorrect gamma compensation that might happen

at the time of scanning. After that we briefly discuss the decoding strategy.

5.6.1 Estimating and Undoing Rotation

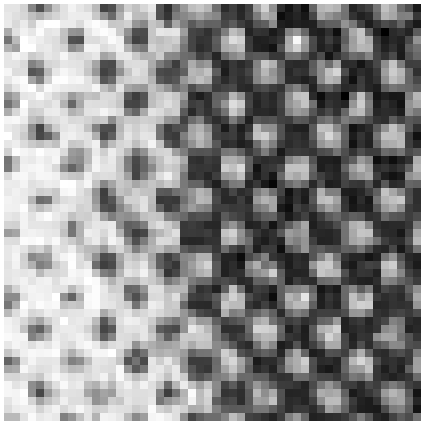
A novel method to estimate the rotation that an image might undergo during the scanning process is proposed in this section. The method is based on the fact that laser printers use an ordered digital halftoning algorithm for printing. An advantage of the proposed technique for print-scan resilient hiding is that there is no penalty for estimating and undoing rotation, which is unlike previous approaches [93, 61] that typically use FM transform to achieve rotation invariance. It should be noted that the proposed derotation technique cannot be applied to a general rotation attack (e.g., if the image is rotated digitally) since it uses the printer halftone screen to estimate the rotation angle.

As stated before, laser printers employ an ordered halftoning algorithm to generate the binary image. In most laser printers, the cells lie in a deterministic periodic array oriented at an angle of 45 degrees for grayscale images. This is because there is a sharp minimum in perceptual sensitivity for spatial frequencies oriented at 45 degrees from horizontal. Note that some modern printers use different orientation angle (33 degree) when printing at certain specific settings. In order to illustrate our ideas, we restrict ourselves to a printer that uses a 45 degree halftone screen for grayscale images. It should, however, be noted that the algorithm and the results presented here would remain perfectly valid when an angle other than 45 degrees (such as 33 degree) is used. The idea is to capture the halftone pattern by high resolution scanning, which is then used to estimate the rotation angle as described in the following section.

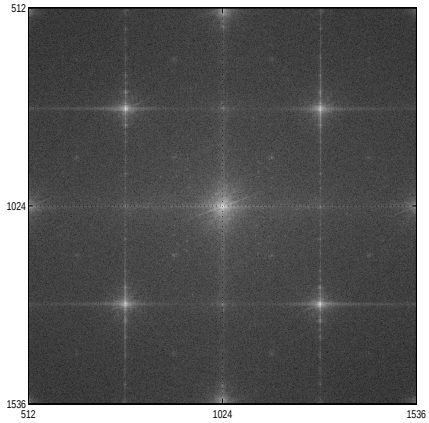
The angle by which an image gets rotated during the scanning process can be estimated using the fact that the halftone cells in the printout (of the image) are oriented at a 45 degree angle with the horizontal. Figure 5.9 (a) and (c) show magnified portions of a printed and scanned image without rotation and with rotation during scanning. Figure 5.9 (b) and (d) show the magnitude spectrum of the images in Figure 5.9 (a) and (c) respectively. Due to the orientation of the halftone cells, a peak can be seen at an angle of 45 degrees for the image without rotation. When the image gets rotated during the scanning process, the peaks also get rotated as in 5.9 (d). Note that a number of secondary peaks are observed, but only a part with the *primary* peaks is displayed here. The angle of the peak can be used to estimate the rotation and the image can be derotated before the hidden data is decoded.

It should be noted that the Fourier transform is symmetric such that out of the four quadrants, the values are same for a pair of quadrants (for the displayed fft-shifted spectrum, quadrants I and III have same values and so do quadrants II and IV). The rotation angle can be estimated by measuring the angle of the peak in any of the four quadrants in the magnitude spectrum.

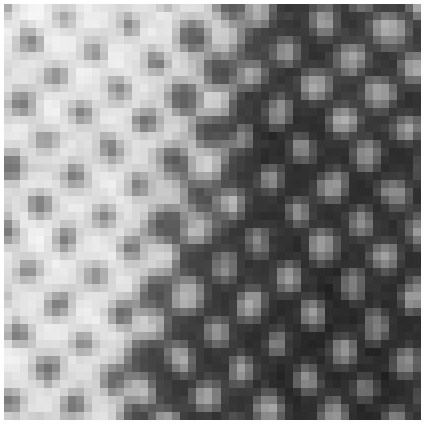
It is observed that the size of image on the printout is not exactly same as that in the digital form. For example, when a 512×512 image is printed with 72 pixels per inch, the height measured on the printout turns out to be about 0.05 inches longer than its width. Due to this discrepancy, the angle measured for a peak in the first quadrant of the Fourier magnitude spectrum is slightly different from that in the second quadrant. In practice, we use average of the two angles as an estimate of the rotation angle.



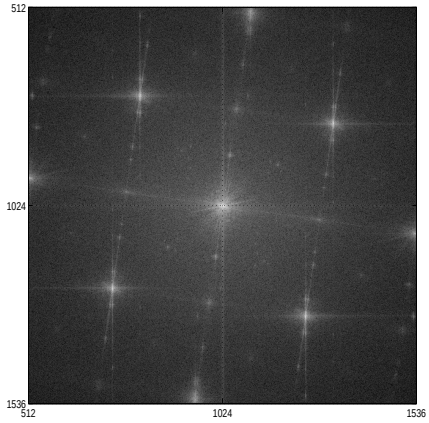
(a) No rotation



(b) Spectrum of (a)



(c) With rotation



(d) Spectrum of (c)

Figure 5.9: Zoomed printed-and-scanned images and their Fourier spectra.

In the following we describe the algorithm used in estimating and derotating an image after scanning (at 600 dpi resolution).

1. Crop a block of 2048×2048 pixels from the center of the scanned image and take its DFT.

2. Find peaks (location of the maximum values) in the magnitude spectrum for the first and second quadrants. Let these angles (in degrees) be denoted by θ_1 and θ_2.

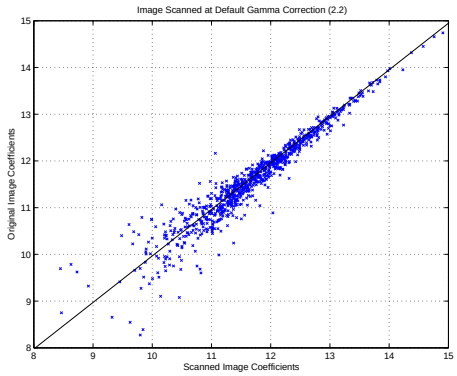
3. Compute the estimate of the rotation angle as $\theta_r = (\theta_1 + \theta_2)/2 - 45$ and use bicubic interpolation to rotate the image by θ_r.

4. The image is then cropped from the background by finding the edges with largest magnitudes of transition (first order difference) in intensity values.

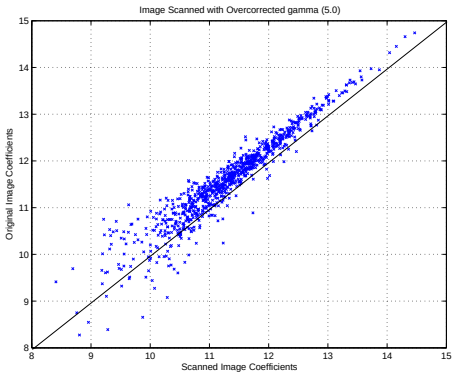
Using the above algorithm, we can estimate the angle by which the scanned image has been rotated. The image is then derotated and cropped automatically. As it can be seen in Section 5.7, automatic derotation outperforms the best manual placing of the printout on scanner flatbed.

5.6.2 Dealing with Incorrect Gamma Compensation

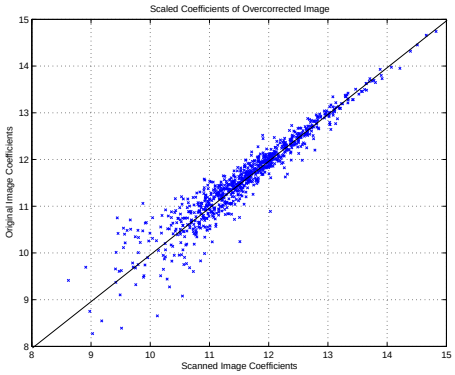
When the printout of an image is scanned, it undergoes gamma-correction, as discussed before. Different computer systems may have different system gamma (e.g., Macintosh computers use a gamma of 1.72, while the gamma for PCs is 2.5) and it is important to apply the right gamma correction at the receiver. We experimented with various gamma correction values at the scanner in order



(a) Image scanned with a gamma correction of 2.2.



(b) Image scanned with a gamma correction of 5.0 (overcorrection).



(c) The coefficients of the overcorrected scanned image of (b) are scaled by 1.023.

Figure 5.10: Effect of gamma correction: Logarithm of low frequency DFT coefficient magnitudes of original 512×512 peppers image are plotted against those of the same image after printing and scanning. $1/N^2$ scaling has not been applied in computing the DFT. It can be seen that the plot is spread around the x=y line for the gamma correction of (a). If the image is overcorrected at the scanner (b), the response shifts. However, a plot spread around x=y can be achieved by scaling of the coefficients (c).

to find a way to deal with incorrect gamma compensation. As in all previous experiments, we study the logarithm of DFT coefficient magnitudes here.

In the experiments, we observed that when the gamma correction is varied at the scanner, the logarithm of DFT coefficient magnitudes of the scanned image are scaled by a constant factor. Figure 5.10 plots the original and scanned image DFT coefficients (or the *input/output* characteristics) for the default gamma correction (monitor gamma = 2.2) and for overcorrection (monitor gamma = 5.0). Figure 5.10 (c) shows the same plot when the scanned image DFT coefficient magnitudes are scaled. It can be seen that the plot in (c) is quite close to the unity gain line.

The gamma correction applied, in general, depends on the system gamma. If the devices are not calibrated, there could be some mismatch. We can, however, deal with incorrect gamma compensation simply by scaling the log DFT coefficient magnitudes. The scaling factor may be determined experimentally for a particular scanner and monitor pair, or the decoder can try a few scaling factors and use the one which works best.

5.6.3 Decoding

Once the image is automatically derotated and the gamma compensation is corrected using the above algorithms, it is then used to demodulate and decode the embedded information. Readers are referred to our prior work [109] for a detailed discussion on decoding for the employed coding framework. Here we just provide an overview.

The receiver takes the DFT of the image coefficients and scans the coefficients in the same order as the encoder. It employs the same threshold criteria as the

encoder to estimate the locations where data has been embedded. Hard-decision decoding of the embedded channel symbols is performed. This is because it is difficult to quantify the statistics of the print-scan attack. For those coefficients that do not pass the threshold test, an erasure is passed to the *channel* decoder. Finally, the sum-product algorithm [58] is used to decode the hidden information bits leading to error-free recovery of the hidden data in spite of the strong attacks. The use of powerful channel codes provides robustness to the embedded data against a variety of other attacks as well.

5.7 Results

We now present the performance of our embedding schemes in this section. Note that the setup for evaluating the hiding techniques remains same as that in our experimental setting (Section 5.4). Images with hidden data are printed and the digital scanned image is fed to a receiver that decodes the hidden data after undoing the rotation using the automated algorithm of Section 5.6.1. We have evaluated the hiding schemes for several images and for many different printers. Note that when scanning at higher resolutions (300 samples per inch or more), the choice of scanner does not make much difference in the performance of the embedding schemes.

For each hiding scheme, we present the maximum number of bits that can be hidden and recovered perfectly for five selected sample images. These images were chosen based on varying detail and texture content so as to study their embedding capacities. Note that though we present results for these particular

images, we have conducted experiments with several images and observed similar performance for the other images as well (which depends on the detail and texture content in the images). We believe that presenting the maximum number of bits embedded for these selected images are enough to illustrate the performance of our schemes, and listing these numbers for more images would not provide new insights. In the experiments, the number of bits embedded into the images are increased (in steps of 25 bits), until we fail to recover the hidden data. The bits reported in the Tables (5.1, 5.2, and 5.4) below are the number of bits that can be embedded in that particular image with perfect recovery after scanning. Having discussed the details of the setup, we start the presentation of the results with the SELF hiding scheme.

5.7.1 Surviving Print-Scan with Automatic De-rotation

Figures 5.11-5.13 shows three example images at various stages of embedding, attack, and decoding (for the *Baboon*, *Man*, and *Couple* images). The embedded bits can be recovered from the images after they are printed and scanned, even when the images get rotated during the scanning process. For example, Figure 5.11 (a) and (b) show the original *man* image and the composite image with 500 bits embedded. Figure 5.11 (c) shows the printed-and-scanned image which got rotated during the scanning process. Figure 5.11 (d) shows the automatically derotated image (using the algorithm proposed in Section 5.6.1). Figure 5.11 (e) shows the image after the background is automatically cropped. Similarly, Figure 5.12 show the *baboon* image example, and Figure 5.13 show the intermediate-stage images for the *couple* image.

(a) Original 512×512 Man image

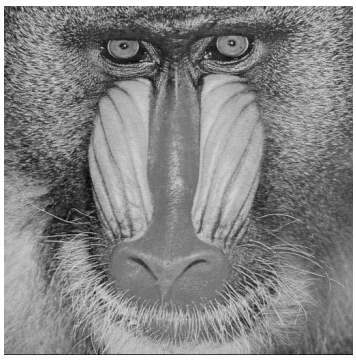
(b) Image with 500 bits hidden

(c) Printed and scanned image

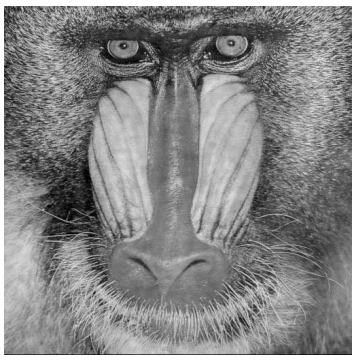
(d) Automatically derotated image

(e) The derotated image cropped automatically

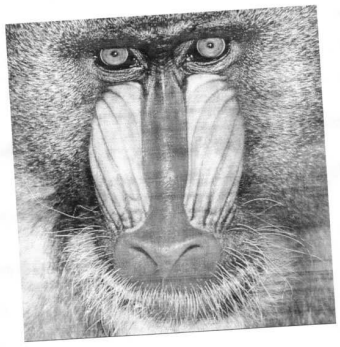
Figure 5.11: Images at various stages of embedding, attack, and decoding for the 512×512 Man image. All the 500 embedded bits have been recovered successfully at the decoder.



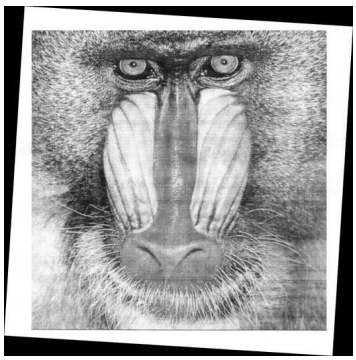
(a) Original 512×512 Baboon image



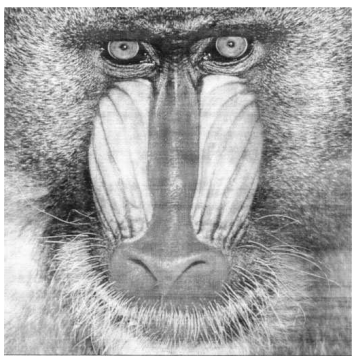
(b) Image with 475 bits hidden



(c) Printed and scanned image



(d) Automatically derotated image



(e) The derotated image cropped automatically

Figure 5.12: Images at various stages of embedding, attack, and decoding for the 512×512 Baboon image. All the 475 embedded bits have been recovered successfully at the decoder.

(a) Original 512×512 Couple image

(b) Image with 300 bits hidden

(c) Printed and scanned image

(d) Automatically derotated image

(e) The derotated image cropped automatically

Figure 5.13: Images at various stages of embedding, attack, and decoding for the 512×512 Couple image. All the 300 embedded bits have been recovered successfully at the decoder.

Table 5.1: Number of information bits hidden along with RA code parameters used for various 512×512 images for the print-scan attack. The images with listed number of hidden bits also survive attacks such as 3×3 Gaussian filtering, 4×4 median filtering, heavy JPEG compression (QF = 10), 17 row and 5 columns removal, and aspect ratio change (by 0.8×1.00).

Image	# of bits hidden	RA code rate (1/q)	# of coeff. in band
Peppers	250	1/4	870
Baboon	475	1/6	2450
Bridge	250	1/7	1560
Man	500	1/5	2450
Couple	300	1/6	1560

Table 5.2: Comparison of number of information bits hidden in various 512×512 images in two scenarios: (i) automatic derotation at the decoder, and (ii) careful manual placing of the image printout on the scanner flatbed.

	Number of bits hidden	
Image	Auto. derotation	Manual placing
Peppers	250	225
Baboon	475	350
Bridge	250	200
Man	500	400
Couple	300	275

Table 5.1 shows the number of information bits hidden for various 512×512 images along with the RA code rate and number of candidate embedding coefficients. The listed number of bits were recovered perfectly after the images were printed and scanned with varying degrees of rotation.

Table 5.2 compares the number of information bits hidden in various 512×512 images with automatic derotation at the decoder and with careful manual placing of the image on the flatbed of the scanner to avoid rotation. It can be seen that more information bits can be hidden when automatic derotation is performed at the decoder as compared to careful manual placing without automatic derota-

Table 5.3: Performance of the proposed SELF hiding scheme against various attacks.

Images	# bits hidden	Attacks: Overall error percentage					
		Print-Scan	JPEG compr. QF=10	3×3 Gauss. filter	4×4 Median filter	17 rows 5 cols removed	Asp. rat. change 0.8×1.0
Barbara	367	7.63%	1.77%	0%	2.72%	2.45%	0.27%
Man	1076	15.75%	8.59%	0.09%	3.86%	5.62%	0.09%
Couple	364	10.03 %	4.81%	0%	1.64%	1.24%	0.55 %

tion. It shows that automatic derotation outperforms the best human effort at preventing rotation.

5.7.2 Other Attacks

The images with data hidden using SELF hiding scheme also survive several other attacks included in *Stirmark* [85], e.g., Gaussian or median filtering, rows and/or columns removal, heavy JPEG compression, and aspect ratio change. The number of bits listed in Table 5.1 and 5.2 survive these attacks as well. In Table 5.3, we show the percentage of errors encountered against various attacks for an uncoded transmission. This gives us an idea of the amount of protection needed via error correction codes to deal with those errors. It can be seen that the print-scan process is most severe among all the attacks. Hence, a system with sufficient redundancy to survive the print-scan process would also work against all other attacks. This is consistent with our observation that the images that are designed to survive print-scan process using the SELF hiding scheme survive all the attacks listed in Table 5.3.

It should be noted that much less data can be hidden against the *Stirmark* random bending attack. For example, 73 bits are hidden in Peppers image (with-

Table 5.4: DQIM embedding in phase: Number of information bits hidden along with RA code parameters used for various 512×512 images for the print--scan attack.

Image	# of bits hidden	RA code rate (1/q)	# of coeff. in band
Peppers	125	1/5	576
Baboon	275	1/6	1444
Bridge	250	1/6	1444
Man	225	1/7	1444
Couple	150	1/6	784

out the channel coding) and received with 20 % error. Note that this performance may still be good for watermarking applications, where the watermark sequence is known to the decoder and can be correlated with the hidden data to *detect* the watermark. So far we have discussed the performance of the our SELF embedding scheme (for hiding in magnitude spectrum). Let us now move on to the DQIM hiding method, which embeds data into the phase spectrum of the images.

5.7.3 DQIM Hiding in Phase

For our DQIM hiding in phase method, we are able to embed several hundred bits against the print-scan attack. Table 5.4 shows the number of information bits hidden for various 512×512 images along with the RA code rate and number of candidate embedding coefficients. Here too, all the embedded bits are recovered after the print-scan attack. The volume of embedding depends on the host image, which turns out to be lesser than that of the SELF hiding scheme for embedding in the magnitudes. This is especially true for images such as Peppers and Couple that have many smooth regions within, so that a smaller candidate embedding bands must be used in order to preserve the perceptual quality. Since DFT phase

is known to have more *information* about the image then the magnitudes [47], it is that much more difficult to embed data in the phase spectrum without inducing much perceptual distortion.

5.8 Summary

We have successfully demonstrated print-scan resilient data hiding methods with potential applications such as document authentication and image copyright protection. The robustness of the methods are based on three key components of our approach: choice of embedding strategy based on analytical and experimental modeling of the print-scan process, the use of powerful turbo-like channel codes, and automated algorithms for derotation and correcting gamma compensation at the receiver. In the analytical modeling, we get around the complexity involved by dividing the print-scan operation into simpler sub-processes and identifying the bottlenecks, which are then studied in detail.

There is still much left for future investigation. One can focus on some specific printers and scanners, and analyze the non-linear transformations in more detail so as to design hiding schemes with higher capacities. Another interesting avenue for future work is to leverage the inverse halftoning literature for reducing the affect of colored noise. This way, we can possibly improve the embedding capacity by using the mid (or high) frequency coefficients along with the low frequency ones for hiding.

In the next chapter, we shift the focus from surviving the attacks to evading the detection of the presence of embedded data. The main goal is covert commu-

nication, and the approach is to achieve a small (or zero) divergence between the original and the hidden image distributions using *statistical restoration*.

Chapter 6

Secure Steganography via Statistical Restoration

Steganography, the art and science of communicating in a manner that the very presence of communication is not known to a third party, has a rich history (e.g., [119], and references in [86] and [117]). In 1983, Simmons [102] introduced the modern version of the problem: Alice and Bob are in jail, and want to hatch up an escape plan, but all their communication pass through Willie, the warden. Hence, the communication should be hidden, so that it does not incite the suspicion of Willie. The challenge in the design of steganographic systems is to communicate at high rates without being detectable via statistical, or perceptual analysis.

A general framework for steganography problem is shown in Figure 6.1. Here, the problem is described in terms of the above mentioned prisoner's problem, in which the warden monitors the communication between the prisoners. The steganalyst has to determine whether the sent signal is cover or stego.

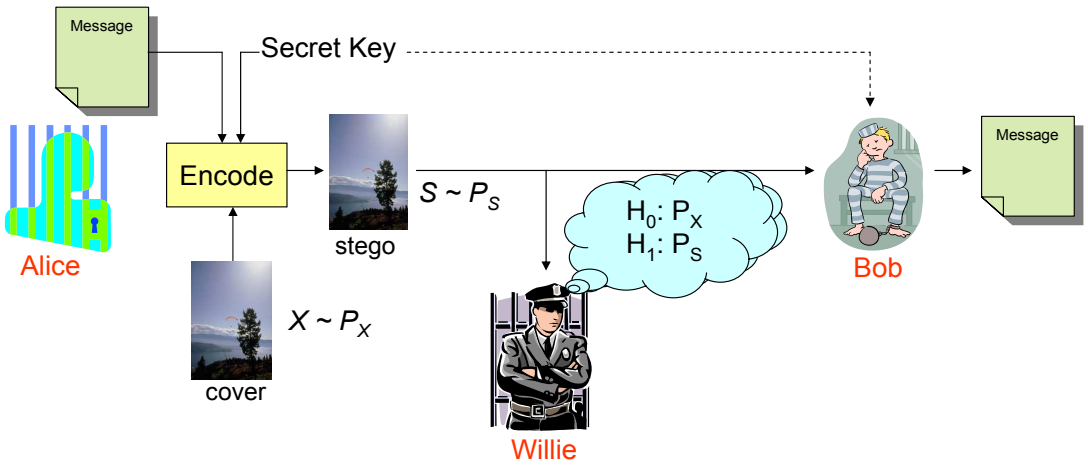


Figure 6.1: General framework of steganography: the prisoner's problem.

6.1 Introduction

In recent years, there has been a great deal of activity in developing data hiding techniques, which have classical applications to steganography, or covert communication, as well as to watermarking for digital rights management. The typical objective in high-volume data hiding is to embed data in a *host* or *cover*, in a manner that is resistant to a number of natural and malicious attacks, and is imperceptible to the casual observer. However, the resulting *stego* signal can be subjected to increasingly sophisticated *steganalysis* techniques for detecting the presence of hidden data.

In this chapter, we propose a framework for the design of embedding schemes that can evade statistical steganalysis while hiding at high rates, and achieve robustness against attacks. We are motivated by the notion of ϵ-secure steganography proposed by Cachin [12], in which the relative entropy (also called Kullback-Leibler or K-L divergence) between the cover and stego distributions is less than

or equal to ϵ. Our approach for achieving a small ϵ is to employ *statistical restoration*, wherein a portion of the data-hider's "distortion budget" is spent in repairing the damage done to the image statistics by the embedding process. To ensure that the restoration does not interfere with decoding, a fixed percentage of host symbols are set aside for restoration, while the rest are used for embedding. A secret key, shared between the encoder and the decoder, determines the embedding and compensation locations. While we focus on hiding in images in this chapter, the approach itself applies to general host signals.

One of the first popular steganalysis tools proposed in the literature was *Stegdetect* [90], which uses a chi-square statistic on the histogram of transform coefficients to detect least significant bit (LSB) hiding. Stegdetect can be improved upon by more sophisticated detection-theoretic approaches [29]. Such methods, which are based on the histogram of the host coefficients, have spurred the development of hiding techniques that make as little change to the histogram as possible. Provos' Outguess algorithm [89] was an early attempt at histogram compensation for LSB hiding, while Eggers et al [32] suggest a more rigorous approach to the same end, using histogram-preserving data-mapping (HPDM). In turn, steganalysis tools that counter such histogram-preserving hiding methods have been developed, such as detection, for image-based hiding, of block-DCT embedding by evaluation of the increase in blockiness due to hiding [39, 128].

Unlike most of the steganographic approaches discussed above, our framework allows design of schemes that can have perfect security by achieving zero Kullback-Leibler (K-L) divergence between the cover and the stego signals. One can match continuous statistics using the proposed approach, not just discrete (or quantized)

statistics. Only a couple of prior schemes, to the best of our knowledge, can potentially achieve zero KL divergence for continuous host statistics: Gullion et al [48], and Wang and Moulin [129, 75]. Both the approaches, however, have some serious issues that limit their practical applicability. Guillon et al [48] suggest transforming the source to get a uniform PMF source. The message is hidden in this with the quantization hiding scheme, which is known not to change the PMF of uniform sources. Therefore, the PMF after transforming back is also the same as the original. This method, however, is not likely to be robust, and also, there is no way to control the distortion induced by the embedding process. Wang and Moulin [129] propose a reduced rate variant of standard QIM, called the stochastic QIM, which can be made to have zero K-L divergence. However, because of the stochastic nature of the hiding process, the method is likely to yield high error rates when embedding large volumes of data. Note that in [75], the proposed stochastic QIM technique embeds only one bit of information.

The proposed framework allows design of robust techniques that are not fragile against attacks, unlike most of the methods proposed in the literature so far. While certainly not the most important issue for steganographic systems, robustness against "natural" attacks such as compression or additive noise is highly desirable. Most of the prior schemes, such as OutGuess [89], HPDM [32], Sallee's model based methods [94, 95], and Fridrich et al's perturbed quantization [40], are fragile against any modifications to the image.

The techniques do not rely on accurate modeling of the host statistics. This is unlike Sallee's model-based steganography [94, 95], in which the hider ensures that the stego signal conforms to a given model. In the absence of a perfect model for

the host, nothing stops the steganalyzer from selecting a *better* model by spending more computational power, and hence detect the embedded data. This is indeed practically shown in [11], where Sallee's Cauchy-model based JPEG steganography is broken by using only the first order statistics. Our approach is very difficult to detect in this manner, since the stego marginals are simply restored to conform to the host's empirical density, rather than invoking a statistical model for the host's marginals.

For any statistical restoration technique, the steganalyst can always go one step further, and use higher order joint statistics[1] than those that have been compensated for, typically at the cost of higher computational complexity. Thus, hiding techniques that compensate marginals are easily detected using the cover memory. For example, a few approaches (Fridrich et al [39], and Wang and Moulin [128]) detect block-DCT embedding by modeling the increase in *blockiness* of the image due to the block-DCT hiding. We use our framework of statistical restoration to design a method that defeats this type of block-based steganalysis. In this case, the statistic to be restored is the difference of adjacent pixels values within the blocks and on the block boundaries. In general, the framework presented in the paper can be applied to restore statistics of any order.

We use supervised learning on a set of over 1000 natural images to evaluate the performance of our schemes. We find that statistical restoration severely affects the steganalysis performance of both DCT-histogram and blockiness methods. We achieve very low K-L divergence between original and cover distributions at fairly high embedding rates. The image could also survive JPEG compression or

[1]We use the term 'first order' statistics to denote the marginal statistics, and 'higher order' statistics to actually mean joint statistics with higher-order dependencies.

recompression without compromising the undetectability.

The rest of the chapter is organized as follows. In Section 6.2, we discuss the limits of steganographic systems. Next, we introduce the concept of statistical restoration in Section 6.3, in which we also present a technique for restoration with minimum mean squared error (MMSE) criteria. In Section 6.4, we extend the statistical restoration idea to a framework that can achieve perfect security by having zero KL divergence between original and stego distributions. Next, in Section 6.5, we propose to use a variable bin-size in analyzing the statistics, which provides several advantages. Based on the framework, several practical schemes are designed for image steganography in Section 6.6. The results are presented in Section 6.7, followed by a brief summary of the chapter in Section 6.8.

6.2 The Limits of Steganography

Modern steganography has become a game with escalating sophistication between the hider and the steganalyst. This is evident from our discussion in Section 2.7 of the state of the art in steganography and steganalysis. It is seen that, many times, a steganography scheme is proposed to evade a particular steganalysis technique. This in turn is detected by an improved steganalysis method. With these iterations still happening, at this point, it is not clear who, the data hider or the steganalyst, will come out to be the winner.

In the following, we discuss a method for perfectly secure communication under most stringent (idealized) assumptions. After that, we move on to more

realistic setting, and present a model for the design and analysis of stegosystems[2].

6.2.1 One-time Pad for Steganography

Consider the problem of steganography, in which, Alice wants to communicate with Bob by sending an innocuous cover signal, which is monitored by Willie, the warden. In cryptography, Shannon proposed the concept of *one-time pad* in [99], which provide means for a perfectly secure communication between Alice and Bob[3]. One-time pad provides information-theoretic security, i.e., the code cannot be broken by the cryptanalyst even if he or she has infinite computational resources. The security of the system is based only on the secrecy and randomness of the key.

Is it possible to achieve provable security for steganography? The equivalent of one-time pad in steganography, if it exists, would be a system that enables communication between Alice and Bob via an innocuous cover, when Willie, the warden, has the perfect knowledge (i.e., a deterministic model) of all the possible cover signals, and also has infinite computational resources to *try* all possible secret keys. The first assumption states that the steganalyst has access to *all* possible original cover signals, and even a small change in just one pixel is detectable. The second assumption regarding the key is similar to one made by Shannon in [99]. Under these assumptions, any modification in the cover signal by Alice is suspicious, and moreover, by trying all the possible keys, Willie can potentially

[2]We use the word *stegosystem* as a short form for a *steganographic system*.

[3]Note that the word 'security' in this sentence means *cryptographic* security (the meaning of the message is not revealed). This is not same as the steganographic security considered in this chapter, wherein the presence of communication must not be revealed.

figure out the contents of the message.

Though it may seem impossible, there is a way for Alice and Bob to communicate secretly under these idealized assumptions too. As shown in Figure 6.2, the equivalent of one-time pad in steganography would be that Alice and Bob share a secret key and a database or library of natural images[4], which is assumed to be known to Willie as well. To communicate a message to Bob, Alice sends an image from the library, which is indexed by the message, which in turn is scrambled by a secret key. This is equivalent to Shannon's idea of one-time pad, except that, instead of sending an encrypted message, Alice now sends an image from the database, which is indexed by the encrypted message.

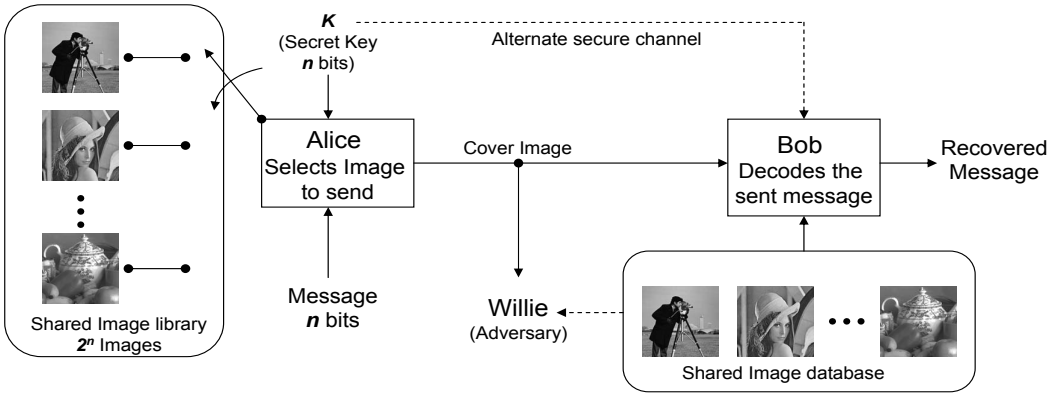


Figure 6.2: One-time pad for steganography: Perfect communication is possible between Alice (the encoder) and Bob (the decoder), even when Willie (the adversary) has the perfect knowledge of all possible cover signals. Using a n-bit secret key, and a database of 2^n images, a message of size n bits can be securely sent once.

Let us now investigate the number of bits that can be sent, i.e., the *capacity* of this stegosystem. It is clear that in order to communicate n bits, Alice and

[4]Note that the discussion presented here refers to digital images, but the system can be employed for any cover, such as audio, video, text, or in general, any signal that can be considered *innocuous* by the warden.

Bob must share a database of at least 2^n images. Since the image library is known to Willie as well, a secret key of at least n bits must be shared between the encoder (Alice) and the decoder (Bob) to index the images so as to enable perfectly secure communication. An interesting point to note is that a perfectly secure stegosystem with a finite capacity can be constructed *without* using data hiding at all.

It should, however, be noted that the above system has several limitations in deploying it practically. Similar to Shannon's one-time pad, the secret key, which has the same complexity (i.e., the number of bits) as the message itself, has to be communicated via an alternate secure channel. Also, one particular key can be used only once (hence the name, one-time pad). Moreover, it may not be feasible to share a large database of innocuous cover signals between the encoder and the decoder. If we restrict ourselves to the situation where Alice and Bob cannot share a database of images, and they must communicate through only one given image, then the capacity of such a system is $\log_2(1) = 0$. In other words, a perfectly secure communication is not possible. This is not entirely surprising because of the assumption that Willie has perfect knowledge of the all possible original cover images. In the following section, we relax this assumption.

6.2.2 A Model for Steganography

Let us now move closer to a real-world system, in which, the steganalyst does not have a perfect knowledge of the cover signals. Willie now has, at best, only a stochastic model for the cover signal instead of a deterministic one. In this case, we can expected to have a finite non-zero capacity for having perfectly secure

communication, even when one particular given image must be used as the cover. Knowing that Willie does not have the perfect knowledge of the image, Alice can now modify the image to *hide* the message.

The number of bits that can be hidden without inciting Willie's suspicion, i.e., the capacity of the system, strictly depends on the accuracy of the model at Willie's disposal. Willie's understanding of what a 'natural image' is, may consist of a perceptual aspect (suspicious visual artifacts), as well as some statistical conditions (unusual statistical observations). The requirement that the hiding process should not incur any perceptual distortion to the cover signal comes naturally. A number of steganalysis techniques also employ some statistical analysis to detect the presence of embedded data. Thus, in order to communicate without being detected, the data-hider must obey following two conditions.

1. **Perceptual constraint.** The perceptual distortion between the original and stego image should not be more than a certain maximum amount, D_1, for some perceptual distance measure.

2. **Statistical constraint.** The embedding process should not modify the statistics of the host signal more than a very small number, ϵ, for some statistical distance measure.

The above conditions are quite commonly used in the literature. Distortion constraint for limiting the perceptual distortion has long been used in the information-theoretic and game-theoretic analysis of the data hiding problem ([19, 23, 79]). The second condition, the statistical constraint, has been proposed by Cachin [12], which states that the K-L divergence between original signal dis-

tribution, P_X, and the stego signal distribution, P_S, should be less than ϵ, as given below.

$$D(P_X||P_S) \leq \epsilon \tag{6.1}$$

A *perfectly* secure stegosystem should, obviously, have zero K-L divergence.

$$D(P_X||P_S) = 0 \tag{6.2}$$

Cachin (in [12], and its extended version [13]) considers a stegosystem from an information-theoretic and cryptographic point of view, without considering any distortion constraints. Anderson and Petitcolas [8] also discuss security of steganographic systems with a similar perspective. Moulin and Wang, in [80], analyze achievable rates for a very simplified system, for a Bernoulli (equiprobable binary alphabet) source and Hamming distortion.

When the original cover and stego signals are discrete, the two conditions mentioned above, namely the perceptual and the statistical constraints, are sufficient to describe and analyze passive warden stegosystems. However, when the distributions are continuous, there could be a trivial solution to the problem of maximizing the embedding rate while inducing minimum statistical and perceptual distortion. The data can be embedded, for example, using choice of quantizer, with the quantizer step-size Δ tending to zero [5]. In other words, for passive warden case, the embedding capacity is *infinite* for continuous alphabet sources (since the number of bits hidden per host symbol can tend to infinity with the quantizer step-size tending to zero).

We note that, for cover signals such as images and video, the transform domain coefficients (such as DCT, DWT, or DFT) are generally modeled as continuous

[5] Actually, data can be hidden using any method, with a vanishing embedding "strength".

distributions. This, however, does not mean that the capacity of such signals is infinite. The transform coefficients are not *exactly* continuous, since after any modifications in the transform domain, the signal must be transformed back to spatial domain, which leads to round-off errors.

Thus, even when there is no active adversary, there are some attacks, such as the round-off errors, which must be survived. Instead of modeling the effect of round-off errors on the transform coefficients, it is easier to consider the active adversary system, in which the stego system must survive an attack causing a distortion of at most D_2. In the presence of attacks, it is not possible to get the trivial solution to the problem of maximizing rate, and achieving infinite capacity by modifying the cover with vanishing quantizer Δ. This is because, the encoder must now introduce some minimum distortion D_E, in order to have sufficiently large distortion to noise ratio (DNR, or $\frac{D_E}{D_2}$).

From the above paragraphs we know that the stego capacity of continuous cover signals for an active warden is finite. The actual problem of finding the capacity of active warden stegosystems, then, reduces to maximizing the rate of transmission with three constraints, namely, $D(P_X||P_S) = 0$ (zero K-L divergence), $d(X, S) \leq D_1$ (encoder perceptual constraint), and $d(S, Y) \leq D_2$ (attacker maximum distortion), where $d(\cdot, \cdot)$ denotes a perceptual distortion measure.

In this chapter, we do not derive the above theoretical limit, but rather focus on designing practical steganographic schemes that allow secure communication at high rates. A simplified framework is proposed, in which we separate the two problems of surviving the attack, and maintaining statistical transparency. This is done by embedding data in a predefined subset of host symbols in such a way

that they could survive the attack. The remaining symbols are used to restore the statistics of the stego to resemble that of the cover. In the following section, we study such a system in more detail.

6.3 Statistical Restoration

The discussion in the previous section suggests that it would be impossible to communicate secretly if the steganalyst has perfect knowledge of the cover signal. In the real-world scenario, the cover signal is not known to the steganalyst. Even an imprecise stochastic model for natural images is difficult to construct. Hence, certain simplified statistical models (such as, of DCT coefficients), are considered for steganalysis. This is what generates the room for the data-hider. The advantage with the data-hider is that he or she is 'informed' of the cover image, and hence its statistics. Thus, he or she can be assured of perfectly secure communication simply by sending a composite image whose statistics resemble that of the original cover. A natural way to accomplish this is to spend a part of the allocated distortion budget to *restore* the statistics. Note that we are considering the simplified statistics under scrutiny, and not the complete underlying random process.

In order to make sure the restoration process does not interfere with decoding, we allocate certain coefficients for embedding and use the rest for restoration. By separating the hiding and compensation locations, we make sure that the robustness properties of the employed embedding algorithm remain intact. This is unlike previous compensation approaches that use entropy codecs [32, 94], and

hence, are fragile against attacks. Note that in [89], Provos proposes a method to restore the DCT histogram statistics for JPEG steganography. Note that unlike this approach, we match continuous distribution (i.e., the probability density function, or the *pdf*) of the cover, rather than discrete or quantized statistics (PMF). Moreover, we use a MMSE criteria to minimize the distortion during compensation (discussed in Section 6.3.3).

6.3.1 Matching Continuous Distribution

The goal, in our framework, is to match the continuous pdf of the cover signal. Note that, in general, distributions of transform coefficients (such as DCT or DWT) are modeled as continuous. By matching the continuous probability density of the cover, we can *advertise* the stego image in an uncompressed format, such as TIFF, or BMP. Moreover, the stego image statistics would continue to match that of the original, even when it is compressed (i.e., if the DCT coefficients are quantized).

Matching the continuous statistics means that we must not leave any "gaps" in the stego image pdf. To achieve this, we must have a embedding algorithm that does not leave any gaps in the histogram, and a compensation procedure, which can correct the difference in the histogram after embedding. In our statistical restoration framework, the host symbols are divided into two *streams*: embedding stream, and compensation stream. We use QIM with dithering to embed the data into host symbols that lie in the embedding stream. By using dithering, we make sure that there are no gaps in the hidden image histogram. Next, the host symbols in the compensation stream are modified to match the original as closely

as possible.

Note that, in practice, both the data hider and the steganalyst have only the empirical density of the image coefficient values. Thus, the histograms must be studied using a bin size, denoted w. During the compensation procedure (discussed in Section 6.3.3), some host symbols are moved from one bin to another. We assume that, for small enough bin-width the distribution of the original cover signal is uniform over the bin, a commonly used assumption in signal compression literature [45]. Thus, when a host symbol is to be moved to another bin, we generate a uniform random data, which becomes the new value of the host symbol. Note that, in theory, we can always match any distribution that is within the bin. This can be done by generating the pseudorandom data according to the distribution in the particular bin to which a compensation coefficient is to be moved. We, however, follow the uniform distribution assumption for simplicity of implementation, and find it quite effective in practice.

Let $f_X(x)$ and $f_S(s)$ be the cover and stego probability density functions respectively. For I bins centered at $t[i]$, $i \in [1, I]$, with a constant width w, the expected histogram for data generated from $f_X(x)$ is as given below.

$$P_X^E[i] = \int_{t[i]-w/2}^{t[i]+w/2} f_X(x)dx \tag{6.3}$$

Similarly, $P_X^E[i]$ is obtained from $f_S(s)$ in the same way. The superscript E denotes that this is expected histogram, to discriminate it from empirical histograms computed from random realizations. In this chapter, we generally refer to these expected quantized pdfs as PMFs.

6.3.2 Rate vs. Security

The restoration process reduces the size of the message that can be hidden, which is the cost of increasing the security. We can characterize this cost by studying the amount of data that can be hidden in an idealized data source with a given probability mass function (PMF). Let $\lambda \in [0, 1)$ be the ratio of host symbols used for hiding, so $1 - \lambda$ is the ratio remaining to match the cover PMF. If $P_X[i]$ is the cover PMF, $P_S[i]$ the standard (uncompensated) stego PMF, $P'_C[i]$ and $P_C[i]$ the PMF of compensating host symbols before and after compensation respectively, and $P_Z[i]$ the PMF of the final output, our goal can be summarized as:

$$\begin{aligned} P_Z[i] &:= \lambda P_S[i] + (1-\lambda) P_C[i] \\ &= P_X[i] \; \forall \, i \end{aligned} \tag{6.4}$$

Typically P_S can be derived directly from P_X. The amount of data that can be hidden is proportional to the number of samples that can be hidden in. So to maximize the amount of data we send, we seek to maximize λ for a given cover PMF subject to the constraint in (6.4), and the constraints imposed on the compensating PMF, namely $\sum P_C[i] = 1$ and $P_C[i] \geq 0 \; \forall \, i$. Substituting $P_C[i] = \frac{P_X[i] - \lambda P_S[i]}{1-\lambda}$ from (6.4), the first constraint is true for any λ. For the second constraint we find $\lambda \leq \frac{P_X[i]}{P_S[i]} \; \forall \, i$. This gives us an upper limit on the percentage of samples we can use for hiding, or equivalently, the rate at which we can secretly embed. Since the data-hider must choose a fixed percentage of symbols beforehand, λ can not be a function of i, and hence a worst-case λ is chosen: $\lambda = \min_i \frac{P_X[i]}{P_S[i]}$. We now address the next obvious question of how to actually perform the restoration. A strategy to modify the compensation host

symbols with a minimum mean squared error (MMSE) criteria is discussed below.

Let us now study the tradeoff between embedding rate and security. Let us revisit the conditions on the embedding rate λ derived above. If we apply the constraint $\lambda = \min_i \frac{P_X[i]}{P_S[i]}$ to typical PMFs, we run into erratic behavior in the low-probability tails. The ratio $\frac{P_X[i]}{P_S[i]}$ can vary widely here, from infinitesimally small to huge. e.g. $P_X(\text{event } A) = 1 \times 10^{-9}, P_S(A) = 1 \times 10^{-6}, \lambda = 0.001$; only a tenth of a percent of the samples can be used. Since this happens only in the low probability regions in general, the effect of PMF differences in these regions on the net divergence is small. So to avoid this problem we can relax exact equality constraint and ignore a small region of low probability. That is, we do not require compensation in a small, low probability region of the PMF. So now λ is chosen as the minimum $\frac{P_X[i]}{P_S[i]}$ over the high-probability compensated region.

In addition to the divergence introduced due to the ignored region, since (6.4) is not true for all i, P_C must be normalized to satisfy the unity sum constraint, adding a small change across the PMF. Though the net effect is to introduce a small amount of divergence, λ and the corresponding hiding rate can only increase.

The tradeoff between the desired security from detection and the hiding rate can be studied by finding the rate corresponding to several different sizes of ignored (uncompensated) regions. We also note that simply embedding in fewer coefficients also reduces the detectability. However, in Figure 6.3 we see that a large decrease in divergence can be made with a small drop in rate using restoration, which is not possible by merely embedding less. This is true for both Laplacian and Gaussian PMFs over a range of variances.

An example of compensation for the Gaussian pdf is presented in Figure 6.4,

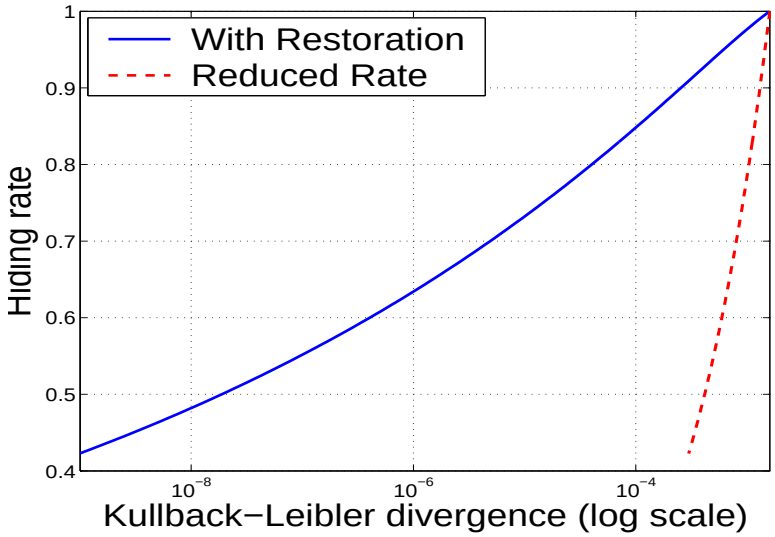
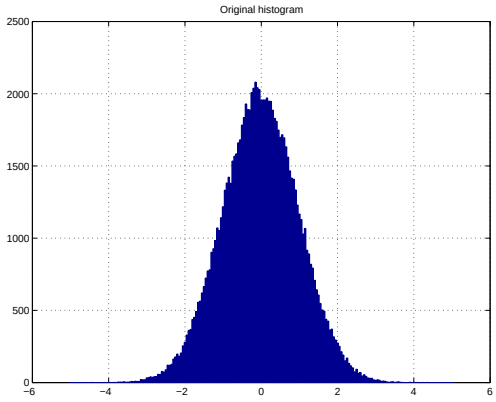


Figure 6.3: Rate, security tradeoff for Gaussian cover. As expected, compensating is a more efficient means of increasing security than simply reducing the rate.

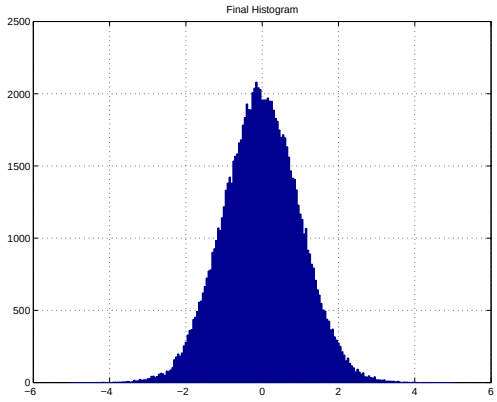
in which the proposed low-divergence achieving method is used to embed and compensate. As it can be seen, there is some difference in the low-probability tail region, which is ignored for compensation. Note that the error is quite small compared to the total number of samples used in this Monte Carlo simulation.

6.3.3 Restoration with MMSE criteria

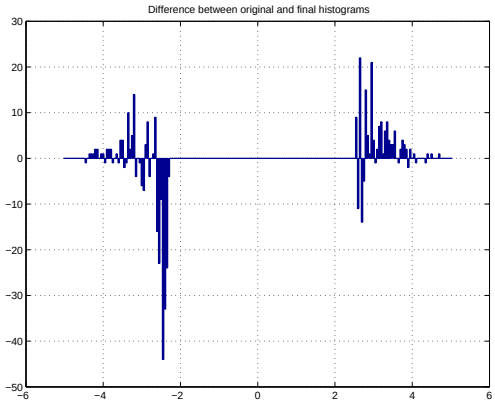
The distribution of the compensation host symbols $P'_C[i]$ must be modified to a target distribution: $P_C[i] = \frac{P_X[i]-\lambda P_S[i]}{(1-\lambda)}$. This would not be as straightforward as saying that if the embedding process modifies a host symbol from A to B, find another host symbol (in the compensation stream) with value B and modify it to A. If for example the hiding process itself modifies another host symbol from B to A, the above change would not be required. It would be very inefficient if such an approach is followed. Another situation could be when $P[B] < P[A]$ so



(a) Original histogram



(b) Final histogram after embedding and compensation



(c) Difference between original and final histograms

Figure 6.4: Low divergence compensation for a Gaussian cover: The original, and final histograms, and their differences for embedding in Gaussian cover signals. Here, the low-probability tail regions are ignored for compensation. The $\sigma/\Delta = 2$, number of samples are 100,000, the bin-width is 0.05, and the embedding rate, λ is 0.45.

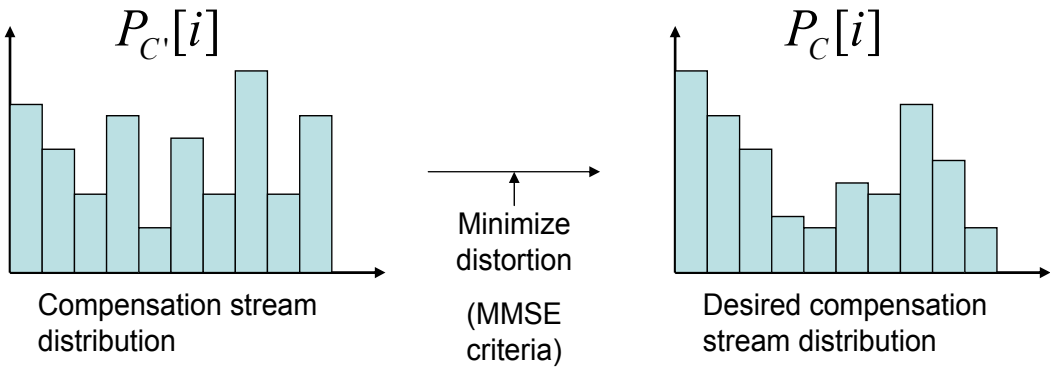


Figure 6.5: Restoration set-up: A target distribution is to be achieved using an MMSE criteria.

that one would soon run out of symbols with value B to compensate for data embedding. As shown in Figure 6.5, to efficiently use our distortion budget, we must modify the compensation stream to achieve a target distribution $P_C[i]$ with a MMSE criteria.

Histogram modification is a well studied problem in the image processing literature. The typical requirement here is that the elements of the input data with the same values must be mapped to the same output values after modification. This way, however, the target histogram can be matched only approximately. In our problem, while it is important to match the target histogram nearly perfectly, the restriction of changing same value symbols to same output values is not present. We are free to change to any values as long as the overall MSE is minimized, and the target histogram is within the ϵ divergence range.

This problem of histogram modification with MMSE criteria was first considered by Mese and Vaidyanathan [68], who propose solving an integer linear programming problem to obtain a mapping matrix. Tzschoppe et al [120] show

that a simpler solution exists, which does not require solving a linear programming problem. They prove a theorem essentially showing that to achieve a MMSE mapping, all the bins of the target histogram must be filled in an increasing order by mapping the input data with values in increasing order. This means that first the bin $i = 1$ of the target histogram must be filled with $P_C[1]$ smallest compensation host symbols. The bin $i = 2$ will be filled next with the $P_C[2]$ smallest remaining symbols, and so on. We note that the mapping would be similar even if the process is started from the last bin and filled in a decreasing order.

In the actual implementation, the above algorithm is slightly modified to ensure that the high probability regions are compensated before the low probability tail. Instead of starting the compensation from the first index (i.e., the lowest value), we separate the positive and negative sections of the histogram and perform their restorations independently. For the positive part, the restoration is done in an increasing order starting from the 'zero' bin. For the negative part, the restoration is done in the descending order starting from the the next bin smaller than zero. For the histograms centered around zero, which is the case for both the practical scenarios considered in this chapter, this procedure compensates the high probability regions first.

6.4 Achieving Zero K-L Divergence

In the previous sections, we observed that it is impossible to completely compensate the low-probability region and match the cover density exactly. The embedding rate λ has to be reduced by a huge factor, and hence, in the above

section, we simply ignore certain low-probability region for compensation. We now consider a scheme that can achieve perfect security by having zero K-L divergence.

The idea for achieving zero K-L divergence is quite simple. As seen in the previous section, since the low-probability region is hard to compensate, we just avoid embedding in that region. This way, the low-probability region need not be restored, and hence, we can potentially achieve zero K-L divergence at good embedding rates. Note that by not hiding in low probability region, we do give up some embedding rate, but we can potentially have larger $\frac{P_X[i]}{P_S[i]}$ over the region in which we are embedding. There is a trade-off between increase in embedding rate by having a larger $\frac{P_X[i]}{P_S[i]}$, and decrease in rate by giving up the low probability region for compensation.

6.4.1 Practical Considerations

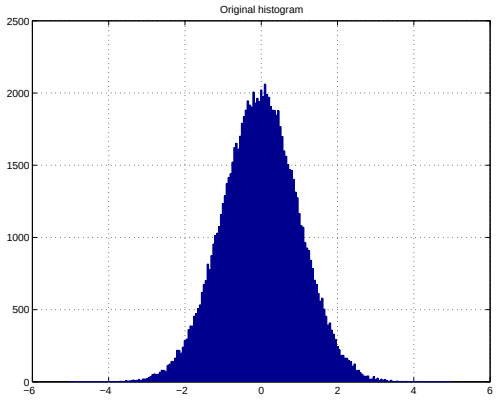
Most distributions encountered in practice, such as Gaussian, or Laplacian density functions, have low-probability tails, and it is possible to avoid embedding in the low-probability region by using a threshold. That is, the encoder would not embed in the host symbols with absolute values greater than a predetermined threshold. The decoder shares this threshold value, which then uses the same criteria to decide whether there was data hidden or not.

We choose the threshold by optimizing the rate-loss due to not embedding in low-probability region of the host distribution, and the gain in rate by minimizing $\frac{P_X[i]}{P_S[i]}$ over a smaller subset. However, the choice of threshold cannot be arbitrary, since we must make sure that the embedded data is decodable at the receiver.

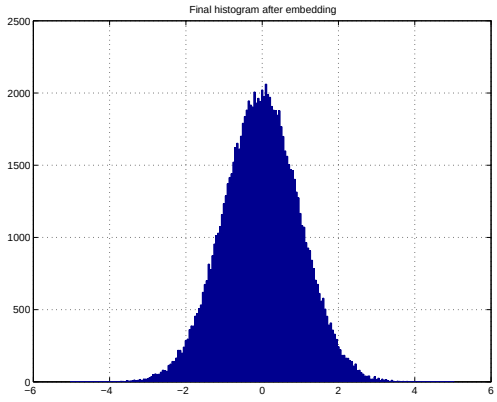
In the presence of attacks, simply using a threshold to determine the hiding locations may cause desynchronization problems at the decoder. Even if no attacks are considered, the decoder might get confused if the embedding algorithm hides in a host symbol that was below the threshold, but its value increases to a value above the threshold after hiding.

For QIM embedding, we can get around this problem by choosing the threshold to be an integer multiple of the quantization interval Δ. Then, data is embedded in all the host symbols whose absolute values are smaller than Δ (for two-sided symmetric distributions, such as Gaussian). QIM embedding would not change the coefficient beyond $t\Delta$, where t is the positive integer threshold. When dithering is used, the quantizers are shifted by the dither sequence, but it is known to the decoder as well. In the presence of attacks, some coefficients' values may increase to be above the threshold, leading to deletion of the symbol, and some coefficients may decrease causing an insertion. This insertion-deletion problem is similar to the one encountered in Chapter 3 (discussed in Section 3.4). We can employ a coding framework similar to the one used in Section 3.4.

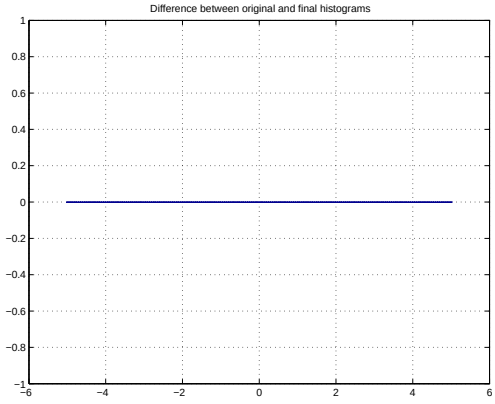
To demonstrate the practical applicability of the system, in Figure 6.6, we present a zero-divergence compensation example for a Gaussian host. As it can be seen, we can achieve exact final histogram, and hence, zero K-L divergence practically as well, at high embedding rates. In this example, we can hide 33,242 bits in 100,000 host samples with perfect restoration.



(a) Original histogram



(b) Final histogram after embedding and compensation



(c) Difference between original and final histograms

Figure 6.6: Zero K-L divergence compensation for a Gaussian cover: The original, and final histograms, and their differences for embedding in Gaussian cover signals. A threshold is used to avoid hiding in the low-probability region. The $\sigma/\Delta = 2$, number of samples are 100,000, the bin-width is 0.05, and the λ is 0.45. Due to the threshold used, the actual embedding rate is 0.33.

6.5 Variable Bin-Size

In the above framework, we have to set a fixed bin-size for the analysis of the statistics when the involved distributions are continuous. Thus, it is natural to ask what happens if the steganalyst analyzes the statistics with a finer bin size. It seems obvious that he or she will be able to detect the stego images because the observations made are *finer*. However, this statement is true only under the assumption that the number of samples are infinite. When there are finite number of samples, a finer bin-size does not guarantee a better observation, and hence, a better detection performance.

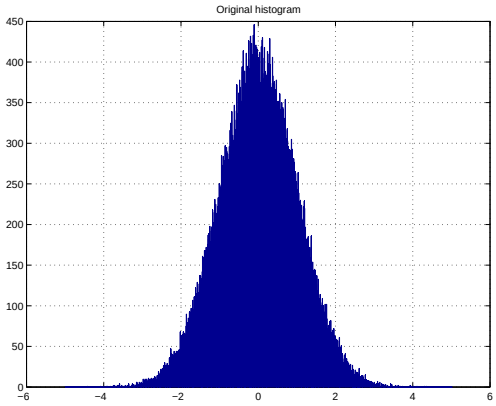
For a moment, let us assume that both the hider and the steganalyst have infinite number of samples, so that an arbitrary degree of precision can be achieved by choosing very small bin sizes. When the underlying cover distribution is known to the hider and the steganalyst, the hider is at a disadvantage because no matter how small bin size he or she uses, the steganalyst can always use an even smaller one, and potentially detect the presence of hidden data. However, since the underlying distribution is known to the encoder too, he or she can move a coefficient from one bin to another by assigning the new value simply by drawing a new coefficient based on the underlying cover distribution within the bin. Thus, the approach here is to use statistical restoration for a particular bin-size, and then *stochastic* within the bin. This way, the steganalyst cannot detect the presence of embedded data, in spite of having potentially infinite number of samples for analysis.

In reality, however, only a finite number of samples are available to the data

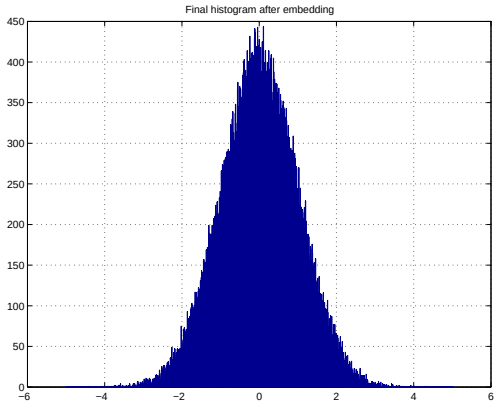
hider as well as the steganalyst. Also, the underlying histogram is not generally known (e.g., for images). In such case, it is not optimal to use as small bin size as possible [97]. If the bin-size used is too small, then the obtained histogram is too jagged, and for too large bin-size, we loose the resolution. In this case, the data hider can use the optimum bin-size recommended in [97]. However, the optimum bin-size must be determined based on the particular empirical distribution.

Another solution from the point of view of the data hider is that he or she can employ a variable bin size in a way that there are a fixed predetermined number of host symbols in every bin. This way, the bin width gets automatically adjusted so that it is finer in the high probability regions, and wider in the low-probability regions. The idea here is to match the original histogram more precisely in the high probability regions compared to the low-probability parts. Thus, the stego image will not get detected by the steganalyst even if he or she uses a very fine bin-size for analysis.

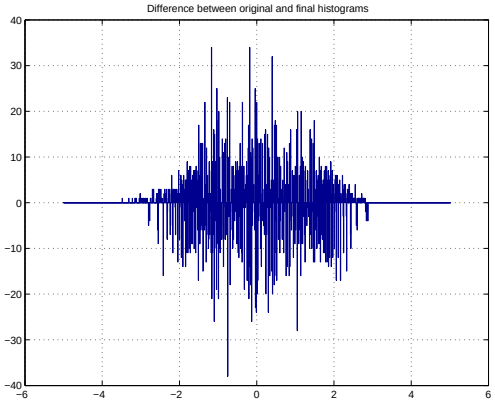
In Figure 6.7, we present and example of compensation using a variable bin-width for a Gaussian cover signal. All the bins have exactly 250 host symbols. As expected, there is some difference between the original and final distributions. However, now we need not worry about the exact bin-size used by the steganalyst to analyze the histogram. In the presented example, even when a much smaller bin-size of 0.01 is used by the steganalyst, the difference is quite small compared to the total number of samples.



(a) Original histogram



(b) Final histogram after embedding and compensation



(c) Difference between original and final histograms

Figure 6.7: Variable bin-size compensation for a Gaussian cover: The original, and final histograms, and their differences for embedding in Gaussian cover signals. The bin-size used is variable, such that all the bins have 250 host symbols. A threshold is also used to avoid hiding in the low-probability region. The $\sigma/\Delta = 2$, number of samples are 100,000, the bin-width is 0.01 (five times smaller than the examples of Figures 6.4 and 6.6.), and the λ is 0.45.

6.6 Practical Schemes

In this section, we describe several practical schemes based on the idea of statistical restoration.

6.6.1 Restoring Marginal Statistics

Several steganalysis approaches [90, 115] detect the JPEG steganography techniques by hypothesis testing on the marginal distribution of the DCT coefficients. We here propose a method that restores the histogram of the DCT coefficients so as to evade this type of steganalysis.

The host image is divided into 8×8 non-overlapping blocks and its 2-d DCT is taken. Those coefficients that lie in a low frequency band of 21 coefficients are considered to be eligible for data embedding or compensation. Now, out of all eligible coefficients, a fixed percentage (we use 25-40% in our experiments) are set aside for hiding and the rest are used for compensation. Data is embedded into the coefficients designated for hiding using dithered quantization. Finally, the compensation coefficients are modified using the algorithm described in Section 6.3.3 so that the stego image histogram closely matches that of the original cover.

The use of dithering in our design makes it possible to match the unquantized source histogram, so that even if the image is compressed or recompressed by the data-hider or an adversary, we neither lose the embedded data nor compromise the undetectability. The stego image can be advertised as any uncompressed format, (e.g. TIFF, BMP, RAW) or subsequently compressed at any quality factor and will continue to closely match the source.

The tradeoff between rate and security (as discussed in Section 6.3.2) implies that the source histogram cannot be matched exactly if we want to communicate at a reasonable rate. Also, in practice, we must work with a limited number of available compensation coefficients. Hence, depending on the chosen rate of embedding, we cannot perfectly match a part of the source histogram towards the low probability tail region. Therefore, we would expect a smart detector to perform better than just a random guess, and this partly explains the better-than-random performance of our supervised learning tests. Below we describe an implementation for quantized DCT coefficients that achieve perfect security by not embedding in low-probability regions.

6.6.2 JPEG Steganography

Here we describe an adaptation of our zero K-L divergence framework for a JPEG steganography scheme. The goal here is to embed in a JPEG compressed image at a particular quality factor, such that the stego image is also a JPEG image at the same quality factor with *exactly* the same distribution of the DCT coefficients. We employ the framework presented in Section 6.4, to achieve the same stego histogram as original, for the JPEG quantized DCT coefficients.

In the actual implementation, we again go to the block-DCT transform domain by dividing the image into 8×8 non-overlapping blocks, taking 2-d DCT, and dividing by the JPEG quantization matrix. The coefficients are quantized since the input image is assumed to be JPEG compressed. As before, those coefficients that lie in a low frequency band of 21 coefficients are considered to be eligible for data embedding or compensation. Again, out of all eligible coefficients, a

fixed percentage (say, 40%) are set aside for hiding and the rest are used for compensation. The hiding and compensation locations are pre-determined based on a secret key shared between the encoder and the decoder. We then embed data using $\pm k$ LSB steganography (with $k = 1$) into those coefficients that are in the hiding stream. Note that QIM cannot be used because the coefficients here are already quantized. Those coefficients whose magnitude is greater than a positive integer threshold, and hence are in the low-probability region, are not used for embedding information. The coefficients in the compensation stream are modified as per the MMSE algorithm presented in Section 6.3.3.

6.6.3 Defeating Block-Based Steganalysis

We now turn our attention to steganalysis schemes that use the cover memory to detect the hidden data. In particular, we focus on techniques that bank on the increase in the blockiness due to block-DCT embedding [39, 128]. It can be seen that these methods basically use a function or a subset of a two-dimensional histogram. For example, Wang and Moulin [128] use one-dimensional histograms of value differences of two populations: one within the blocks, and another along the block boundaries. We note that the value difference histogram can be derived by summing along the diagonals of the two-dimensional histogram. This way the most relevant information is kept while reducing the complexity (of a two-dimensional histogram). Here we propose a method that restores the pixel value differences within the blocks as well as along the block boundaries, so as to survive the steganalysis technique proposed in [128].

A subset of 8×8 blocks are used for data embedding and the rest are set

aside for restoring the pixel difference histograms. In the blocks designated for data embedding, data is hidden in a low frequency band comprised of 21 DCT coefficients. Next, the pixel values of the compensation blocks are modified (with MMSE criteria, as described in Section 6.3.3) so that the difference histograms are very close to the original. Note that the two histograms (within the blocks and along the block boundaries) are restored separately to match their respective originals.

6.7 Results

We now describe the performance of the proposed methods in this section. We use a supervised learning machine on a set of over 1000 natural images to discriminate between the cover and the stego images (as in [115]). The machine is trained on the statistics of hundreds of examples of distinct stego and cover images, and is then tested on its ability to correctly classify a different, unknown set of cover and stego images.

6.7.1 Continuous PDF Restoration Methods

As a first step in examining the efficacy of statistical restoration, we compare the divergence between cover and stego for standard hiding and for hiding with compensation at the same rate. Embedding at a rate of $\lambda = 0.35$ in a Gaussian cover, the divergence for statistically restored dithered-QIM hiding is 1.3×10^{-3}, roughly a five-fold improvement over the standard QIM which yields a divergence of 5.9×10^{-3}. Similar improvement is also seen for a set of real image statistics,

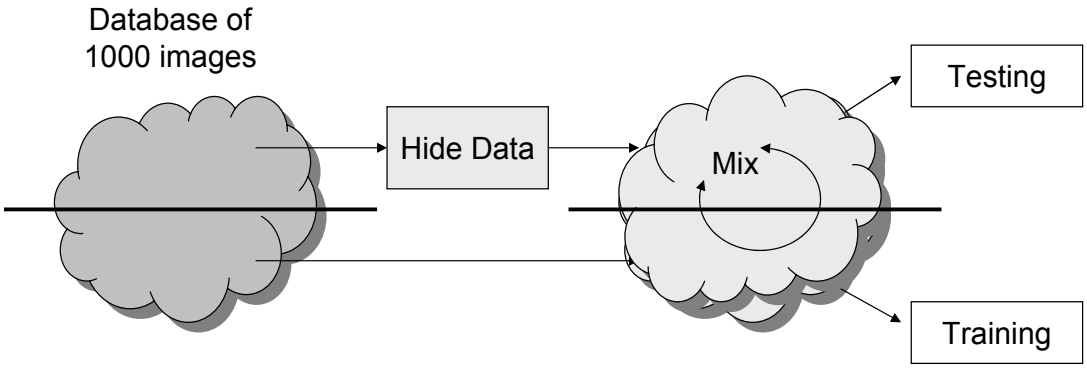


Figure 6.8: Set-up for steganalysis using supervised learning on natural images.

Table 6.1: Performance of uncompensated vs. compensated methods for over 1000 images in supervised learning tests. It is seen that restoration can severely affect the steganalysis performance.

	Dithered QIM		Adaptive dithered QIM		Blockiness based scheme	
	Un-comp.	comp.	Un-comp.	comp.	Un-comp.	comp.
P(m)	0.075	0.525	0.701	0.796	0.043	0.259
P(fa)	0.177	0.000	0.000	0.074	0.000	0.007
P(m)+ P(fa)	0.252	**0.525**	0.701	**0.870**	0.043	**0.266**

wherein, the average divergence for standard hiding is 6.5×10^{-3}, which reduces to 2.1×10^{-3} for compensated embedding. Although detection is still possible, restoration greatly increases the error probabilities of an ideal detector. For example, a steganalyst would require more than three times as many samples to achieve the same detection rates with standard hiding in images as with hiding with restoration.

Now we present the results for testing with supervised learning machine on a set of 1000 natural images. Figure 6.8 shows the set-up for steganalysis system

using supervised learning. Three embedding methods were tested: dithered QIM, adaptive dithered QIM (of [109]), and blockiness based scheme (of Section 6.6.3). For each of these schemes, we trained and tested two machines on the same sets of images and at the same rate; one with compensation, one without. Table 6.1 lists the probability of false alarm, P(fa), and the probability of missed detection, P(m), for each of these configurations. It can be seen that for the dithered QIM hiding, the detector has twice the sum of errors while detecting restored hiding as compared to standard hiding. Figure 6.9 plots the probability of missed verses the false alarm for adaptive dithered QIM scheme. Remember that in this adaptive embedding, data is not hidden into coefficients that do not get quantized to zero.

For the blockiness compensation scheme, the sum of errors is six times greater for restored hiding than for standard hiding. Figure 6.10 shows the ROC curve for the blockiness compensated hiding verses the non-restored hiding. Note that a λ of 0.35 is used in all the cases, which translates to hiding roughly 30100 bits in a 512×512 image.

6.7.2 JPEG Steganography with Perfect Restoration

We now present the results for our JPEG steganography technique presented in Section 6.6.2. Again, we use supervised learning on 1000 natural images to test the system. A support vector machine (SVM) classifier is trained and tested on the first order statistics of the DCT coefficients. We here compare the perfect restoration JPEG steganography with the standard QIM. We embed random bits into images using both the techniques, and then train and test the SVM classifier using the DCT histogram. Same rate and same images are used in both the cases.

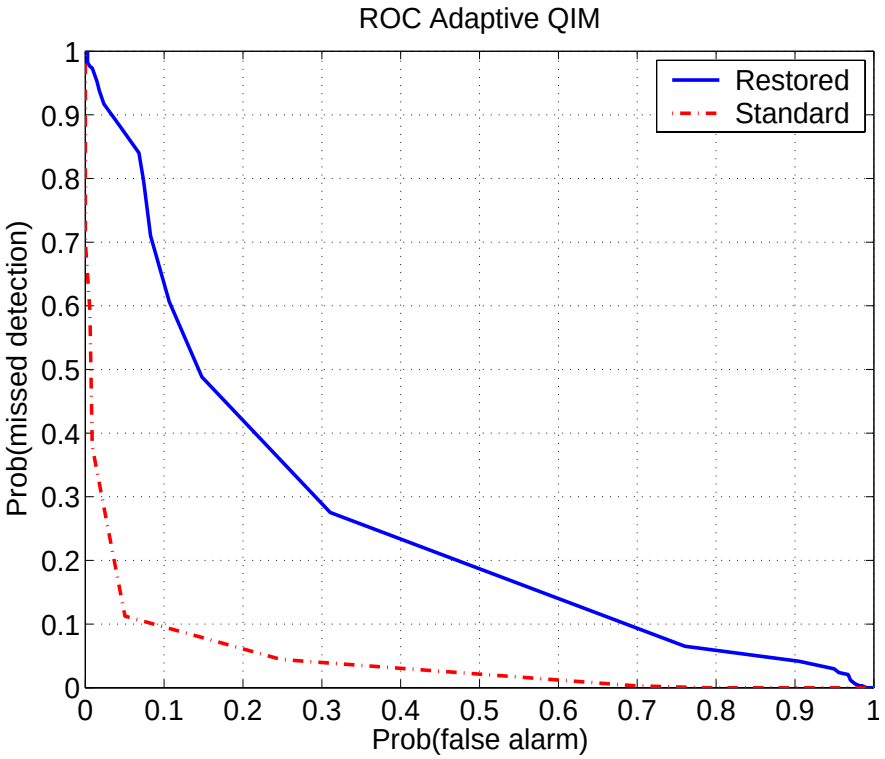


Figure 6.9: Detection of standard adaptive-QIM verses adaptive restored QIM: As expected, the restored QIM can evade steganalysis better than the standard adaptive-QIM.

Figure 6.11 plots the probability of missed detection verses probability of false alarm for both the schemes. As expected, the detector performance is random for the JPEG steganography scheme with perfect restoration.

6.8 Summary

We have demonstrated how statistical restoration can be employed for robust and secure communication. Our experiments indicate that the detectability of our statistically compensated QIM schemes is lower than the standard QIM. Our

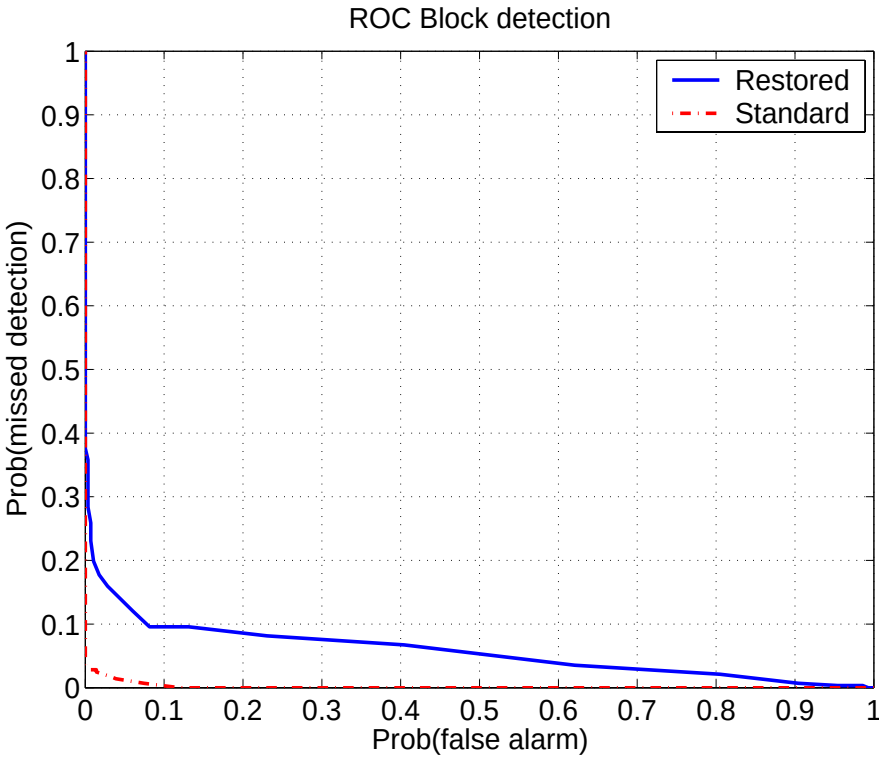


Figure 6.10: Detection using blockiness evaluation of non-restored embedding verses blockiness-restoration hiding: blockiness-restored embedding can evade steganalysis better than the non-restored hiding.

JPEG steganography scheme, which is based on the zero K-L divergence QIM framework, achieves perfect security when the DCT histograms are considered for steganalysis. This can potentially be detected by blockiness-based steganalysis techniques. To counter this, we have implemented the statistical restoration framework to restore the blockiness statistics as well. Using this scheme, we can significantly lower the detection rates for block-based steganalysis as well. The approach presented in this chapter allows design of schemes that can *guarantee* secure transmission at sufficiently high embedding rates.

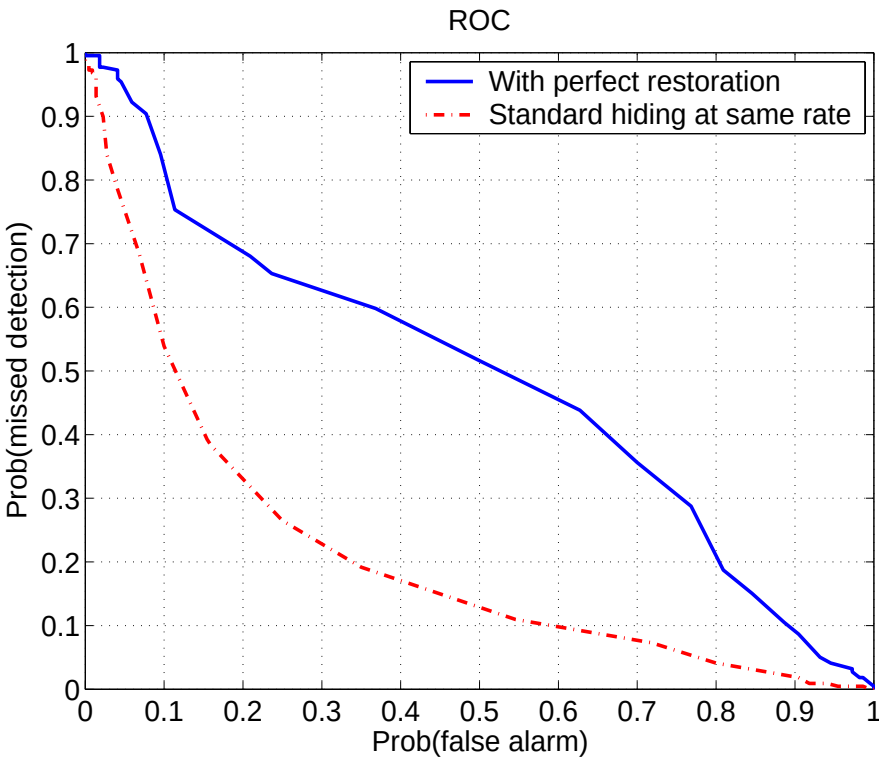


Figure 6.11: Detection of JPEG steganography with standard QIM verses perfect restoration QIM. As expected, the detection for perfect-restoration JPEG scheme is random. However, the standard QIM at same rate is detectable.

Chapter 7

Conclusions and Future Work

In this dissertation, we have addressed several aspects of the information embedding problem. In the first approach, we consider embedding large volume of information without incurring any perceptual distortion, and achieve robustness against many distortion-constrained attacks (such as compression, and additive noise). The embedding capacity we achieve is among the best reported in the literature (see a recent tutorial by Moulin and Koetter [76]). We can hide data of the order of several thousands of bits, in say 512×512 images, with robustness against a number of operations. Such an ability can be leveraged in several exciting applications, such as image annotation, seamless upgrade, error concealment, and broadcast monitoring. The flexibility provided by the employed coding framework in choosing the embedding locations can allow, for example, embedding data in regions of a medical image that are not sensitive for diagnosis. In many such disciplines, the flexibility in choosing hiding locations can allow the use of data hiding technology for annotation and tamper protection of the images

pertaining to these disciplines.

The problem of joint source-channel coding for conventional communication problem has attracted attention from the research community since a long time in spite of the fact that separation theorems for several channels have been proved (for asymptotically reducing probability of error with increasing codeword length). The primary reason for the research drive towards joint source-channel coding schemes is that it provides simplicity of encoding and decoding, and also allows graceful improvement in received quality. This is especially important now because time-varying channels have gained significance lately. With the demonstration, in this thesis, of a practical joint source-channel coding scheme for the data hiding channel (i.e., communication channel with side information about the channel state at the encoder), several new avenues for future research have opened up in both the theoretical analysis and the design of more advanced practical schemes. Moreover, deploying this framework for practical applications such as error concealment of images and video provide an interesting potential for future work.

Design of robust techniques have received the most attention from the researchers in multimedia data hiding. We present a powerful scheme in this thesis that can resist several severe manipulations, such as printing followed by scanning, random bending, heavy compression, rows and/or columns removal, Gaussian or median filtering, and aspect ratio change. While there are schemes available in the literature that can deal with these attacks individually, what we have demonstrated here, is one scheme that can survive all these attacks. Two key factors have contributed to the robustness of this scheme: first, a powerful coding frame-

work that allows dynamic choice of hiding locations, and second, embedding in robust features comprising of selected low-frequency coefficients. The success of the schemes also highlights the usefulness of experimental approach in solving complex problems (such as print-scan resilient hiding).

The problem of steganography, or secure communication is both interesting and significant. We present a practical framework for achieving perfect security by having zero K-L divergence between the original and cover distributions. Key to our efforts is the fact that we do not attempt to model the statistics, but rather match the empirical density of the cover signal. We provide a simple and easy to implement framework that can be employed to construct schemes that can match statistics of any order. Having demonstrated a practical method to achieve zero K-L divergence, it would now be interesting to investigate the capacity of general stegosystems, and see how close the proposed system is from the theoretical capacity. We now present some of the future research directions in more detail.

7.1 Future Work

In this section, we discuss several new avenues of future work, that can extend and improve upon the techniques presented in this thesis. Specifically, we discuss three areas here: (a) deeper investigation of joint source-channel coding strategies, including identifying fundamental limits (Section 7.1.1); (b) further exploration of the print-scan channel so as to increase the number of hidden bits (Section 7.1.2), and extending the work for general digital to analog and back to digital trans-

formations; (c) investigating the capacity of the steganographic systems (Section 7.1.3). Another future direction is to employ our robust data embedding methods for various disciplines and applications, such as, using our methods for image and video error concealment, embedding meta-data into bio-molecular images, and document authentication using print-scan resilient hiding.

7.1.1 Further Study of Joint Source-Channel Hiding

The efficacy of the simple joint source-channel hiding scheme proposed in Chapter 4 prompts us to ask more fundamental questions. What are the ultimate performance limits? How far are we from this limit? What would be the construction of involved embedding strategies that could perform better than the currently used method, and possibly operate close to the theoretical capacity? While a number of joint source-channel coding approaches have been studied for the Gaussian channel, joint source-channel coding for data hiding has not been studied prior to our own work, and there are a number of open issues that one can investigate.

An interesting future research direction is to analyze and compare the analog information hiding[1] scheme presented in Chapter 4 with the theoretically achievable limits. It should be noted that the theoretical limit expression derived in Chapter 4, equation (4.3) is very general and can be termed the "vector" embedding limit (termed thus because the optimal strategy involves vector quantization of the host) for data hiding. Our analog information hiding scheme, however, embeds information on a per host-symbol basis. Such "scalar" hiding has the merit

[1]Note that by '*analog* information hiding', we mean embedding *continuous* alphabet sources.

of simplicity, and it would be interesting to investigate the fundamental performance limits for analog scalar hiding. Thus we can compare the performance of our current scheme with a "scalar capacity". It should be noted that our work on digital hiding in Chapter 3 shows that, for AWGN attacks, there is roughly only a 2 dB penalty for scalar hiding. We would like to derive analogous results for analog embedded data under AWGN attacks.

Determining the performance limits, and the 'gap' between those and our current hiding scheme can allow further investigation on whether more complex embedding methods can close the gap. For this, the vast literature in joint source-channel coding for the Gaussian channel can be leveraged as appropriate.

7.1.2 Print-Scan Resilient Hiding with Higher Capacity

The print-scan resilient embedding schemes presented in Chapter 5 provide improvement over prior published methods in terms of volume of embedding. The approach used is to divide the print-scan process into simpler sub-processes, then identify the bottlenecks, which are then studied in further detail. In Chapter 5, we have identified three main components of the print-scan process, namely, geometric distortions, non-linear transformations, and colored high-frequency noise. In our study, we focus only on the geometric distortions. A detailed study of the other two components, non-linear effects, and colored noise is an avenue of future work.

In particular, one can focus on some specific printers and scanners, and analyze the non-linear transformations in more detail so as to design hiding schemes with higher capacities. Note that the exact non-linear characteristics depend on

the particular printers and scanners employed in the system. For security applications, such as authentication of documents such as passports and driving licences, it is possible to have devices that are under control of the designer.

Another interesting avenue for future work is to leverage the inverse halftoning literature for reducing the affect of colored noise. There are several effective inverse halftoning methods available, which provide very good performance in terms of the resultant image quality. As discussed in Chapter 5, the colored noise introduced during the printing process leads to distortion in the mid and high frequency coefficients. By using inverse halftoning methods, we can possibly improve the embedding capacity by using the mid (or high) frequency coefficients along with the low frequency ones for hiding. It should be noted that inverse halftoning methods are known to introduce some blurring in the image, which must be dealt with explicitly.

7.1.3 The Capacity of Steganographic Systems

In Chapter 6, we present techniques that can allow high capacity embedding with either low or even zero divergence. A model for steganography is described in Section 6.2, in which the problem of maximizing embedding capacity is set up. Deriving the theoretical capacity of secure steganographic schemes is an avenue of future work. As stated there, the problem of finding the theoretical embedding limit for an i.i.d. cover signal reduces to maximizing the embedding rate with following constraints.

(i) $D(P_X||P_S) = 0$, i.e., the K-L divergence between the original and stego signal distributions is zero.

(ii) $d(X, S) \leq D_1$, i.e., the distortion incurred by the encoder is smaller than or equal to D_1.

(iii) $d(S, Y) \leq D_2$, i.e., the stego signal must survive an attack distortion of at most D_2.

where, $d(\cdot, \cdot)$ is some measure for the perceptual distortion between two media signals. For simplicity of analysis, we can start with the mean squared distortion. For N-tuple X, and Y, the distortion can be written simply as,

$$d(X^N, S^N) = \frac{1}{N} \sum_{i=1}^{N} (X_i - S_i)^2.$$

Note that, due to the presence of attacks, the encoder must introduce certain minimum distortion D_E to the cover signal, in order to have sufficient distortion to noise ratio (DNR, or $\frac{D_E}{D_2}$). An analysis of this set-up would provide insights into the steganography problem, and may lead to the development of schemes with better capacities.

7.2 Summary

In this thesis, we have addressed several fundamental issues in multimedia data hiding, added new requirements, and proposed several schemes and frameworks that provide practical solutions to many challenging problems in this field. The experiments and results presented in this thesis are for data-sets consisting of real images, and hence, the proposed techniques can be readily deployed for practical applications.

An important fundamental contribution of this thesis is the novel use of turbo-like erasure and error correcting codes allowing the encoder to choose embedding locations dynamically. This framework was employed in Chapter 3 for schemes that can embed high-volume data with robustness against a variety of attacks. We also demonstrate a practical technique here which gets to within 2 dB from the theoretical embedding capacity of the scalar QIM. The coding framework has also been applied to techniques that are robust against several malicious attacks including printing followed by scanning (Chapter 5). Focussing on hiding media signature signals into media hosts, we propose a new embedding framework that can provide graceful improvement in received signature signal fidelity (Chapter 4). This has been made possible by the use of a new hybrid digital-analog joint source-channel coding scheme. To the best of our knowledge, such schemes had not been studied prior to our work. Our work has opened up several new avenues for future work including investigating the fundamental limits as well as devising new strategies for joint source-channel hiding. In the final part of the thesis (Chapter 6), we propose steganographic techniques that can evade statistical steganalysis while hiding large number of bits. Now that we have a practical scheme, it would be interesting to investigate the *capacity* of data hiding systems that can evade detection.

Bibliography

[1] http://aakash.ece.ucsb.edu/datahiding/stegdemo.aspx. UCSB data hiding online demonstration. Released on Mar. 09, 2005.

[2] http://www.bioimage.ucsb.edu/. Center for Bio-Image Informatics, University of California, Santa Barbara.

[3] http://www.stegoarchive.com. Steganography software archive.

[4] M. D. Adams and F. Kossentini. JasPer: A software-based JPEG-2000 codec implementation. In *Proceedings of ICIP*, Vancouver, Canada, September 2000.

[5] C. B. Adsumilli, M. C. Q. Farias, M. Carli, and S. K. Mitra. A hybrid constrained unequal error protection and data hiding scheme for packet video transmission. In *Proceedings of ICASSP*, volume 5, pages 680–683, April 2003.

[6] C. B. Adsumilli, M. C. Q. Farias, S. K. Mitra, and M. Carli. A robust error concealment technique using data hiding for image and video transmission over lossy channels. *Accepted for future publication, IEEE Trans. on Circuits and Systems for Video Technology*, 2005.

[7] M. Alghoniemy and A. H. Tewfik. Geometric invariance in image watermarking. *IEEE Trans. on Image Processing*, 13(2):145–153, February 2004.

[8] R. J. Anderson and F. A. P. Petitcolas. On the limits of steganography. *IEEE Journal on Selected Areas in Communications*, 16(4):474–481, May 1998.

[9] M. Barni and F. Bartolini. *Watermark Systems Engineering*. Marcel Dekker, 2004.

[10] P. Bas, J.-M. Chassery, and B. Macq. Geometrically invariant watermarking using feature points. *IEEE Trans. on Image Processing*, 11(9):1014–1028, September 2002.

[11] R. Bohme and A. Westfeld. Breaking Cauchy model-based JPEG steganography with first order statistics. *P. Samarati et al (Eds.): ESORICS 2004, LNCS 3193*, pages 125–140, 2004.

[12] C. Cachin. An information theoretic model for steganography. *LNCS: 2nd Int'l Workshop on Information Hiding*, 1525:306–318, 1998.

[13] C. Cachin. An information theoretic model for steganography. *Information and Computation*, 192:41–56, July 2004.

[14] J. J. Chae. *Robust Techniques for Hiding Data in Images and Video*. PhD thesis, University of California, Santa Barbara, June 1999.

[15] J. J. Chae and B. S. Manjunath. Data hiding in video. In *Proceedings of ICIP*, volume 1, pages 311–315, October 1999.

[16] B. Chen. *Design and Analysis of Digital Watermarking, Information Embedding, and Data Hiding Systems*. PhD thesis, Massachusetts Institute of Technology, June 2001.

[17] B. Chen and G. W. Wornell. Analog error-correcting codes based on chaotic dynamical systems. *IEEE Trans. on Communications*, 46(7):881–890, July 1998.

[18] B. Chen and G. W. Wornell. Dither modulation: A new approach to digital watermarking and information embedding. In *Proceedings of SPIE: Security and Watermarking of Multimedia Contents*, January 1999.

[19] B. Chen and G. W. Wornell. Quantization index modulation: A class of provably good methods for digital watermarking and information embedding. *IEEE Trans. on Information Theory*, 47(4):1423–1443, May 2001.

[20] J. Chou, L. El Ghaoui, S. S. Pradhan, and K. Ramchandran. On the duality between distributed source coding and data hiding. In *Proceedings of 33rd Asilomar Conference on Signals, Systems and Computers*, November 1999.

[21] J. Chou, S. S. Pradhan, and K. Ramchandran. A robust optimization solution to the data hiding problem using distributed source coding principles. In *Proceedings of CISS*, March 2000.

[22] J. Chou and K. Ramachandran. Robust turbo-based data hiding for image and video sources. In *Proceedings of ICIP*, October 2002.

[23] A. S. Cohen and A. Lapidoth. The Gaussian watermarking game. *IEEE Trans. on Information Theory*, 48(6):1639–1667, June 2002.

[24] M. H. M. Costa. Writing on dirty paper. *IEEE Trans. on Information Theory*, 29(3):439–441, May 1983.

[25] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. Wiley, 1991.

[26] I. Cox, J. Kilian, T. Leighton, and T. Shamoon. Secure spread spectrum watermarking for multimedia. *IEEE Trans. on Image Processing*, 6(12):1673–1687, December 1997.

[27] I. J. Cox, M. J. Miller, and A. L. McKellips. Watermarking as communications with side information. *Proceedings of the IEEE*, 87(7):1127–1141, July 1998.

[28] I. J. Cox, M. L. Miller, and J. A. Bloom. *Digital Watermarking*. Morgan Kaufmann, 2001.

[29] O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekaran, and B.S. Manjunath. Detection of hiding in the least significant bit. *IEEE Trans. on Signal Processing, Supplement on Secure Media I*, 52(10):3046–3058, October 2004.

[30] M. C. Davey and D. J. C. Mackay. Reliable communication over channels with insertions, deletions, and substitutions. *IEEE Trans. on Information Theory*, 47(2):687–698, February 2001.

[31] D. Divsalar, H. Jin, and R. J. McEliece. Coding theorems for turbo-like codes. In *Proceedings of 36th Annual Allerton Conference on Communications, Control, and Computing*, pages 201–210, September 1998.

[32] J. J. Eggers, R. Bauml, and B. Girod. A communications approach to image steganography. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents IV*, San Jose, CA, January 2002.

[33] J. J. Eggers, R. Buml, R. Tzschoppe, and B. Girod. Scalar Costa scheme for information embedding. *IEEE Trans. on Signal Processing*, 51(4):1003–1019, April 2003.

[34] J. J. Eggers and B. Girod. *Informed Watermarking*. Kluwer Academic Publishers, Boston, 2002.

[35] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In *Proceedings of the 6th Information Hiding Workshop*, Toronto, Canada, May 2004.

[36] J. Fridrich and M. Goljan. Digital image steganography using stochastic modulation. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents IV*, pages 191–202, Santa Clara, CA, USA, January 2002.

[37] J. Fridrich, M. Goljan, and R. Du. Reliable detection of LSB steganography in color and grayscale images. In *Proceedings of ACM Workshop on Multimedia and Security*, Ottawa, Canada, 2001.

[38] J. Fridrich, M. Goljan, and D. Hogea. Attacking the OutGuess. In *Proceedings of ACM Workshop on Multimedia and Security*, Juan-Pins, France, 2002.

[39] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *Lecture notes in computer science: 5th Int'l Workshop on Information Hiding*, volume 2578, pages 310–323, 2002.

[40] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal. Writing on wet paper. In *ACM workshop on Multimedia and Security*, Magdeburg, Germany, September 2004.

[41] J. Fridrich, M. Goljan, P. Lisoněk, and D. Soukal. Writing on wet paper. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII*, pages 428–445, San Jose, CA, USA, January 2005.

[42] M. S. Fu and O. C. Au. Data hiding watermarking in halftone images. *IEEE Trans. on Image Processing*, 11(4):477–484, April 2002.

[43] R. G. Gallager. Low density parity check codes. *IRE Trans. on Information Theory*, IT-8(12):21–28, January 1962.

[44] S. I. Gel'Fand and M. S. Pinsker. Coding for channel with random parameters. *Problems of Control and Information Theory*, 9(1):19–31, January 1979.

[45] A. Gersho and R.M. Gray. *Vector quantization and signal compression*. Kluwer Academic Publishers, 1992.

[46] J. D. Gibson and M. G. Kokes. Data embedding for secure communications. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Anaheim, CA, USA, October 2002.

[47] R. C. Gonzalez and R.E. Woods. *Digital image processing.* Addison Wesley, 1992.

[48] P. Guillon, T. Furon, and P. Duhamel. Applied public-key steganography. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents IV*, San Jose, CA, January 2002.

[49] C. Heegard and A. A. El Gamal. On the capacity of computer memory with defects. *IEEE Trans. on Information Theory*, 29(5):731–739, September 1983.

[50] C. Herley. Why watermarking is nonsense. *IEEE Signal Processing Magazine*, 19(5):10–11, September 2002.

[51] N. Jacobsen, K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Image adaptive high volume data hiding based on scalar quantization. In *Proceedings of the IEEE Military Communications Conference (MILCOM)*, Anaheim, CA, USA, October 2002.

[52] H. Jin, A. Khandekar, and R. J. McEliece. Irregular repeat-accumulate codes. In *Proceedings of 2nd Int'l Symposium on Turbo codes and Related Topics*, pages 1–8, September 2000.

[53] N. Johnson, Z. Duric, and S. Jajodia. *Information Hiding: Steganography and Watermarking - Attacks and Countermeasures.* Kluwer Academic Publishers, Boston, 2001.

[54] D. Kacker, T. Camis, and J. P. Allebach. Electrophotographic process embedded in direct binary search. *IEEE Trans. on Image Processing*, 11(3):243–257, March 2002.

[55] J. Kelley, USA TODAY. Terror groups hide behind Web encryption. Published on Feb. 05, 2001, available at http://www.usatoday.com/tech/news/2001-02-05-binladen.htm.

[56] M. Kesal, M. K. Mihcak, R. Koetter, and P. Moulin. Iteratively decodable codes for watermarking applications. In *Proceedings 2nd Int'l Symposium on Turbo Codes and Related Topics*, September 2000.

[57] T. D. Kite, B. L. Evans, and A. C. Bovic. Modeling and quality assessment of halftoning by error diffusion. *IEEE Trans. on Image Processing*, 9(5):909–922, May 2000.

[58] F. R. Kschischang, B. J. Frey, and H.-A. Loeliger. Factor graphs and the sum-product algorithm. *IEEE Trans. on Information Theory*, 47(2):498–519, February 2001.

[59] M. Kutter. Watermarking resisting to translation, rotation and scaling. In *Proceedings of SPIE: Multimedia systems and applications*, volume 3528, pages 423–431, November 1998.

[60] D. Lau and G. Arce. *Modern Digital Halftoning.* Marcel Dekker, 2001.

[61] C. Y. Lin and S. F. Chang. Distortion modeling and invariant extraction for digital image print-and-scan process. In *Intl. Symposium on Multimedia Information Processing*, December 1999.

[62] C.Y. Lin, M. Wu, J. A. Bloom, I. J. Cox, M. L. Miller, and Y. M. Lui. Rotation, scale and translation resilient watermarking for images. *IEEE Trans. on Image Processing*, 10(5):767–782, May 2001.

[63] S. Lyu and H. Farid. Detecting hidden messages using higher-order statistics and support vector machines. In *Lecture notes in computer science: 5th Int'l Workshop on Information Hiding*, volume 2578, 2002.

[64] D. J. C. MacKay and R. M. Neal. Near Shannon limit performance of low density parity check codes. *Electronics Letters*, 32(18):1645–1646, August 1996.

[65] H. S. Malvar and D. A. F. Florêncio. Improved spread spectrum: a new modulation technique for robust watermarking. *IEEE Trans. on Signal Processing*, 51(4):898–905, April 2003.

[66] L. Marvel, C. G. Boncelet Jr., and C. T. Retter. Spread spectrum image steganography. *IEEE Trans. on Image Processing*, 8(8):1075–1083, August 1999.

[67] M. Mese and P. P. Vaidyanathan. Look-up table (LUT) method for inverse halftoning. *IEEE Trans. on Image Processing*, 10(10):1566–1578, October 2001.

[68] M. Mese and P.P. Vaidyanathan. Optimal histogram modification with MSE metric. In *Proceedings of ICASSP*, Salt Lake City, Utah, USA, May 2001.

[69] M. Kivanc Mihcak and R. Venkatesan. Blind image watermarking via derivation and quantization of robust semi-global statistics. In *Proceedings of ICASSP*, volume 4, pages 3453–3456, Austin, TX, USA, May 2002.

[70] A. K. Mikkilineni, G. N. Ali, P.-J. Chiang, G. T. C. Chiu, J. P. Allebach, and E. J. Delp. Signature-embedding in printed documents for security and forensic applications. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VI*, pages 455–466, San Jose, CA, USA, January 2004.

[71] M. L. Miller, G. J. Doerr, and I. J. Cox. Applying informed coding and embedding to design a robust high-capacity watermark. *IEEE Trans. on Image Processing*, 13(6):792–807, June 2004.

[72] U. Mittal and N. Phamdo. Duality theorems for joint source-channel coding. *IEEE Trans. on Information Theory*, 46(4):1263–1275, July 2000.

[73] U. Mittal and N. Phamdo. Hybrid digital-analog joint source-channel codes for broadcasting and robust communications. *IEEE Trans. on Information Theory*, 48(5):1082 –1102, May 2002.

[74] P. Moulin. Comments on "Why watermarking is nonsense". *IEEE Signal Processing Magazine*, 20(6):57–59, November 2003.

[75] P. Moulin and A. Briassouli. A stochastic QIM algorithm for robust, undetectable image watermarking. In *Proceedings of ICIP*, Singapore, October 2004.

[76] P. Moulin and R. Koetter. Data-hiding codes. To appear, *Proceedings of the IEEE*, December 2005.

[77] P. Moulin and M. K. Mihcak. A framework for evaluating the data-hiding capacity of image sources. *IEEE Trans. on Image Processing*, 11(9):1029–1042, September 2002.

[78] P. Moulin and M. K. Mihcak. The parallel-Gaussian watermarking game. *IEEE Trans. on Information Theory*, 50(2):272–289, February 2004.

[79] P. Moulin and J. A. O'Sullivan. Information-theoretic analysis of information hiding. *IEEE Trans. on Information Theory*, 49(3):563–593, March 2003.

[80] P. Moulin and Y. Wang. New results on steganographic capacity. In *Proceedings of CISS*, Princeton, NJ, USA, March 2004.

[81] D. Mukherjee, J. J. Chae, S. K. Mitra, , and B. S. Manjunath. A source and channel-coding framework for vector-based data hiding in video. *IEEE Trans. on Circuits and systems for video technology*, 10(4):630–645, June 2000.

[82] J. A. O'Sullivan, P. Moulin, and J. M. Ettinger. Information-theoretic analysis of steganography. In *Proceedings of the IEEE Symposium on Information Theory*, page 297, Boston, MA, USA, August 1998.

[83] S. Pereira and T. Pun. Robust template matching for affine resistant image watermarks. *IEEE Trans. on Image Processing*, 9(6):1123–1529, June 2000.

[84] S. Pereira, S. Voloshynovskiy, M. Madueo, S. Marchand-Maillet, and T. Pun. Second generation benchmarking and application oriented evaluation. In *3rd International Workshop on Information Hiding*, Pittsburgh, PA, USA, April 2001.

[85] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Attacks on copyright marking systems. In *Proceedings of Information Hiding Workshop, IH'98, LNCS 1525, Springer-Verlag*, pages 219–239, 1998.

[86] F. A. P. Petitcolas, R. J. Anderson, and M. G. Kuhn. Information hiding — A survey. *Proceedings of the IEEE, special issue on Identification and Protection of Multimedia Information*, 87(7):1062–1078, 1999.

[87] C. I. Podilchuk and W. Zeng. Image adaptive watermarking using visual models. *IEEE Journal of Selected Areas in Communication*, 16(4):525–539, 1998.

[88] J. G. Proakis. *Digital Communications*. McGraw-Hill, 1995.

[89] N. Provos. Defending against statistical steganalysis. In *In 10th USENIX Security Symposium*, Washington DC, USA, 2001.

[90] N. Provos and P. Honeyman. Detecting steganographic content on the internet. In *ISOC NDSS'02*, San Diego, CA, February 2002.

[91] M. Ramkumar. *Data Hiding in Multimedia: Theory and Applications*. PhD thesis, New Jersey Institute of Technology, January 2000.

[92] J. Rosen and B. Javidi. Hidden images in halftone pictures. *Applied Optics*, 40(20):3346–3353, July 2001.

[93] J. K. O. Ruanaidh and T. Pun. Rotation, scale and translation invariant spread spectrum digital image watermarking. *Signal Processing*, 66(3):303–317, May 1998.

[94] P. Sallee. Model-based steganography. In *IWDW 2003, LNCS 2939*, pages 154–167, October 2003.

[95] P. Sallee. Model-based methods for steganography and steganalysis. *International Journal of Image Graphics*, 5(1):167–190, 2005.

[96] R. R. Schaller. Moore's law: past, present, and future. *IEEE Spectrum*, 34(6):52–59, June 1997.

[97] D. W. Scott. On optimal and data-based histograms. *Biometrika*, 66(3):605–610, 1979.

[98] C. E. Shannon. A mathematical theory of communication. *The Bell System Technical Journal*, 27:623–656, 1948.

[99] C. E. Shannon. Communication theory of secrecy systems. *The Bell System Technical Journal*, 28:656–715, 1949.

[100] C. E. Shannon. Channels with side information at the transmitter. *IBM Journal Research and Development*, 2:289–293, 1958.

[101] G. Sharma. Targetless scanner color calibration. *Journal of Imaging Science and Technology*, 44(4):301–307, July/August 2000.

[102] G. J. Simmons. The prisoner's problem and the subliminal channel. In *Advances in Cryptology: Proceedings of CRYPTO '83, Plenum Press*, pages 51–67, 1984.

[103] M. Skoglund, N. Phamdo, and F. Alajaji. Design and performace of VQ-based hybrid digital-analog joint source-channel codes. *IEEE Trans. on Information Theory*, 48(3):1082 –1102, March 2002.

[104] E. H. B. Smith. Characterization of image degradation caused by scanning. *Pattern Recognition Letters*, 19(13):1191–1197, 1998.

[105] V. Solachidis and I. Pitas. Circularly symmetric watermark embedding in 2-D DFT domain. *IEEE Trans. on Image Processing*, 10(11):1741–1753, November 2001.

[106] K. Solanki, O. Dabeer, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Robust image-adaptive data hiding: Modeling, source coding and channel coding. In *42ed Annual Allerton Conference on Communications, Control, and Computing*, October 2003.

[107] K. Solanki, O. Dabeer, B. S. Manjunath, U. Madhow, and S. Chandrasekaran. A joint source-channel coding scheme for image-in-image data hiding. In *Proceedings of ICIP*, pages II–743–746, Barcelona, Spain, September 2003.

[108] K. Solanki, N. Jacobsen, S. Chandrasekaran, U. Madhow, and B. S. Manjunath. High-volume data hiding in images: Introducing perceptual criteria into quantization based embedding. In *Proceedings of ICASSP*, Orlando, FL, USA, May 2002.

[109] K. Solanki, N. Jacobsen, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Robust image-adaptive data hiding based on erasure and error correction. *IEEE Trans. on Image Processing*, 13(12):1627 –1639, December 2004.

[110] K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Estimating and undoing rotation for print-scan resilient data hiding. In *Proceedings of ICIP*, Singapore, October 2004.

[111] K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. Modeling the print-scan process for resilient data hiding. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 418–429, March 2005.

[112] K. Solanki, U. Madhow, B. S. Manjunath, and S. Chandrasekaran. 'Print and scan' resilient data hiding in images. Submitted for publication, *IEEE Trans. on Information Forensics and Security*, September 2005.

[113] J. Song and K. R. J. Liu. A data embedded video coding scheme for error-prone channels. *IEEE Trans. on Multimedia*, 3(4):415–423, December 2001.

[114] Y. Steinberg and N. Merhav. Identification in the presence of side information with application to watermarking. *IEEE Trans. on Information Theory*, 47(4):1410–1422, May 2001.

[115] K. Sullivan, Z. Bi, U. Madhow, S. Chandrasekaran, and B.S. Manjunath. Steganalysis of quantization index modulation data hiding. In *Proceedings of ICIP*, Singapore, October 2004.

[116] K. Sullivan, O. Dabeer, U. Madow, B. S. Manujunath, and S. Chandrasekaran. LLRT based detection of LSB hiding. In *Proceedings of ICIP*, volume 1, pages 497–500, September 2003.

[117] M. D. Swanson, M. Kobayashi, and A. H. Tewfik. Multimedia data-embedding and watermarking technologies. *Proceedings of the IEEE*, 86:1064–1087, 1998.

[118] W. Trappe, M. Wu, Z. J. Wang, and K. J. R. Liu. Anti-collusion fingerprinting for multimedia. *IEEE Trans. on Signal Processing*, 51(4):1069–1087, April 2003.

[119] J. Trithemius. Steganographia, 1500. Digital edition can be found at: http://www.esotericarchives.com/tritheim/stegano.htm.

[120] R. Tzschoppe, R. Bauml, and J.J. Eggers. Histogram modifications with minimum MSE distortion. tech. rep., Telecom. Lab., Univ. of Erlangen-Nuremberg, December 2001.

[121] R. Ulichney. *Digital Halftoning*. The MIT Press, 1987.

[122] D. Upham. JSTEG: Modification of the independent JPEG group's JPEG software for steganography.

[123] V. Vaishampayan and S. I. R. Costa. Curves on a sphere, shift-map dynamics and error control for continuous alphabet sources. *IEEE Trans. on Information Theory*, 49(7):1658–1672, July 2003.

[124] R. G. van Schyndel, A. Z. Tirkel, and C. F. Osborne. A digital watermark. In *Proceedings of ICIP*, volume 2, pages 86–90, Austin, TX, USA, November 1994.

[125] S. V. Voloshynovskiy, O. Koval, F. Deguillaume, and T. Pun. Visual communications with side information via distributed printing channels: extended multimedia and security perspectives. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VI*, pages 428–445, San Jose, CA, USA, January 2004.

[126] A. Vongkunghae, J. Yi, and R. B. Wells. A printer model using signal processing techniques. *IEEE Trans. on Image Processing*, 12(7):776–783, July 2003.

[127] G. K. Wallace. The JPEG still picture compression standard. *Communications of the ACM*, 34(4):30–44, 1991.

[128] Y. Wang and P. Moulin. Steganalysis of block-DCT image steganography. In *IEEE workshop on Statistical Signal Processing*, St Louis, MO, USA, September 2003.

[129] Y. Wang and P. Moulin. Steganalysis of block-structured stegotext. In *Proceedings of SPIE: Security, Steganography, and Watermarking of Multimedia Contents VI*, San Jose, CA, January 2004.

[130] A. B. Watson. DCT quantization matrices visually optimized for individual images. In *Proceedings of SPIE*, volume 1913, pages 202–216, September 1993.

[131] A. B. Watson, G. Y. Yang, J. A. Solomon, and J. Villasenor. Visibility of wavelet quantization noise. *IEEE Trans. on Image Processing*, 6(8):1164–1575, August 1997.

[132] A. Westfeld. High capacity despite better steganalysis (F5 - a steganographic algorithm). In *Lecture notes in computer science: 4th Int'l Workshop on Information Hiding*, volume 2137, pages 289–302, 2001.

[133] A. Westfield and A. Pfitzmann. Attacks on steganographic systems. In *Lecture notes in Computer Science: 3rd International Workshop on Information Hiding*, 1999.

[134] S. B. Wicker and V. K. Bhargava. *Reed-Solomon Codes and Their Applications*. IEEE Press, 1994.

[135] R. B. Wolfgang, C. I. Podilchuk, and E. J. Delp. Perceptual watermarks for digital images and video. *Proceedings of the IEEE, special issue on Identification and Protection of Multimedia Information*, 87(7):1108–1126, 1999.

[136] M. Wu and B. Liu. Data hiding in images and video: Part I - fundamental issues and solutions. *IEEE Trans. on Image Processing*, 12(6):685–695, June 2003.

[137] M. Wu, H. Yu, and B. Liu. Data hiding in images and video: Part II - designs and applications. *IEEE Trans. on Image Processing*, 12(6):696–705, June 2003.

[138] X. Zixiang, M. T. Orchard, and K. Ramchandran. Inverse halftoning using wavelets. *IEEE Trans. on Image Processing*, 8(10):1479–1483, October 1999.