CONTENT ADAPTIVE STEGANOGRAPHY – DESIGN AND DETECTION

BY

VOJTĚCH HOLUB

MS, Czech Technical University, Prague, 2010

DISSERTATION

Submitted in partial fulfillment of the requirements for the degree of
Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate School of Binghamton University
State University of New York
2014

Accepted in partial fulfillment of the requirements for the degree of

Doctor of Philosophy in Electrical and Computer Engineering

in the Graduate School of

Binghamton University

State University of New York

2014

April, 2014

Jessica Fridrich, Chair and Faculty Advisor

Department of Electrical and Computer Engineering, Binghamton University

Scott Craver, Member

Department of Electrical and Computer Engineering, Binghamton University

Siwei Lyu, Member

Department of Computer Science, University at Albany

Lijun Yin, Outside Examiner

Department of Computer Science, Binghamton University

Abstract

Currently, the most successful approach to steganography in empirical objects, such as digital images, is to embed the payload while minimizing a suitably defined distortion function. This powerful concept allows the steganographer to evaluate distortion caused by embedding changes based on local image content, hence the name content adaptive steganography. The design of the distortion is essentially the only task left to the steganographer since efficient practical codes exist that embed near the payload–distortion bound.

One of the contributions of this dissertation is a novel approach to steganography of JPEG images. Instead of attempting to preserve an inherently incomplete heuristic model of DCT coefficients, we design a simple distortion in the better-understood spatial domain and use it for computing the distortion caused by modifying JPEG coefficients. Besides avoiding the difficult task of modeling dependencies among JPEG coefficients, JPEG steganography using spatial domain distortion provides a superior security with respect to other current state-of-the-art methods. Virtually all current steganographic methods are implemented with additive distortion functions. However, embedding changes naturally interact when executed in nearby pixels (or when modifying DCT coefficients in the same JPEG block). This dissertation also looks into this difficult issue and points out some new open problems as well as potential advantages of embedding with non-additive distortion functions capable of capturing mutual interaction of embedding changes.

Improving steganographic schemes would not be possible without studying feature-based steganalysis for empirical images. This dissertation, among other contributions, presents a method for capturing dependencies among neighboring pixel residual values using random projections in order to improve steganalysis in the spatial domain. Moreover, a low complexity feature set for JPEG steganalysis using undecimated DCT is introduced. These features greatly improve detection of the most secure JPEG steganographic schemes, which were previously best detected by spatial domain features. We embrace this paradox and discuss these cross-domain steganalytic features and steganographic distortion functions in detail.

To the best of the author's knowledge, this dissertation presents currently the most secure steganographic schemes for grayscale images in spatial domain, JPEG domain, and JPEG domain with side-information. Furthermore, it also presents state-of-the-art feature sets for detection of modern spatial and JPEG steganography.

Acknowledgements

First and foremost, I would like to express my deepest gratitude to Jessica Fridrich, my PhD adviser. It was my honor and pleasure to work with her and I can not express how grateful I am that she brought me to this field, nursed me through uneasy beginnings, and gave me research freedom when I needed it. She was always able to create a friendly yet professional environment that motivated every student to work hard and think for themselves. I will always value our long discussions, hiking trips, Thanksgiving evenings, and our friendship.

I am very grateful to my former and current colleagues from the Digital Data Embedding laboratory – Tomáš Filler, Jan Kodovský, and Tomáš Denemark. Their personal and professional help at the beginning and during my studies was invaluable, not mentioning the amount of their time they spent answering my endless questions.

I am greatly indebted to my amazing family who supported me both emotionally and materially when I decided to study in the United States. Their love and support made my stay so far away from home much easier.

I want to thank to Kevin Hexemer, my good friend and fellow engineer, for our long nerdy discussions and few English corrections in this dissertation.

I would also like to thank to the owner and amazing staff of Cafe Oasis, a very friendly and relaxing place where I spent countless hours thinking and where many of my research ideals originated.

Last but not least, I would like to express my thanks to the Air Force Office of Scientific Research for financial support of my research in steganography and steganalysis. This work would not exist without their support, specifically without grants FA9550-08-1-0084, FA9550-09-1-0147, and FA9950-12-1-0124.

Vojtěch Holub

Binghamton, 2014

Contents

List of Tables

List of Figures

Chapter 1

Introduction

Content of any communication can be kept hidden from an unauthorized observer using standard encryption methods. However, a lot of situations require that the very existence of the communication channel remains secret. Such messages can be incorporated into other inconspicuous communication channels to avoid being detected – this is the goal of steganography. Mankind has always utilized methods of secret communication throughout its past, consequently developing many historical steganographic methods. These historical methods assumed that the adversary is not suspicious about ongoing secret communication, thus the principle of *security through obscurity*. More about history of steganography can be found in Section 1.1.

Steganography, as most any other invention, can be used both for good and evil. Multiple examples of modern steganography, mostly of digital images, are presented in Section 1.2. Modern steganography is based on the famous prisoners' problem defined by Simmons [101] in early 1980's. Finally, Section 1.3 explains how this problem influenced the research field so that steganographers stopped relying on algorithm secrecy

1.1 Ancient history

The word *steganography* is of Greek origin – *steganos* means "covered," and *graphia* "writing." The term steganography was first used by Johannes Trithemius in his trilogy *Polygraphia*. The first two volumes were about cryptography, while the third volume *Steganographia* (1499) was mostly about magic and occultism – it was forbidden by Catholic Church. Interestingly, in 1996 and 1998, multiple hidden mundane steganographic messages were found in the text [94, 28].

Greek historian Herodotus [52] was the first to document the usage of steganography to send messages. A slave was sent by his master to deliver a secret message tattooed on his scalp. After the message was tattooed, the slave waited until the his hair grew back and concealed the message. When he arrived to the recipient, regent of Miletus, the slave's head was shaved again to reveal the message, an was important information that encouraged the regent to rebel against the Persian king.

The most popular steganographic methods between the 13th and 16th century involved written text. One method used a mask, a paper with holes, shared between the sender and recipient. The mask was simply put over the text and the message was revealed. Francis Bacon [3] realized that two different fonts for each letter can be applied to embed binary representations of messages. Given the state of typography at that time, it was relatively inconspicuous.

Brewster [13] devised a very original technique in 1857, which was later used in several wars. He proposed shrinking the message to such an extent that it resembled dirt or dust particles, however,

it could still be read if magnified. After the technology was properly developed, Germans used these "microdots" in World War I. Microdots were also recently proposed in the form of dust for identification of car parts [1].

More details about history of steganography can be obtained in Chapter 1.1 of Ref. [38]. Security of most of the previously mentioned methods was achieved only by assuming ignorance of the adversary – this is sometimes pejoratively called *security through obscurity*. The adversary did not attempt any targeted attack in the sense of modern steganalysis, instead they trained spies and secret services to obtain the necessary information by other means.

1.2 Two faces of modern steganography

With its barely controlled growth since the early 90's, the Internet is a perfect carrier of information. However, the decentralized infrastructure enables third parties to inspect and read the transmitted data. Although this problem can be solved with standard well developed cryptographic methods, some countries ban all encrypted messages. Steganography can be used to sustain private communication in censored countries by hiding the messages in digital media, such as digital images and videos. A documented long-term usage of digital image steganography to avoid censorship of encrypted emails was presented at the 4th International Workshop on Information Hiding [98]. Given the sheer number of multimedia sent to Internet every minute, sustaining a communication channel by an occasional image upload is not likely to raise any suspicion. Therefore, employing steganographic techniques as a means of private communication in censored countries helps preserve the freedom of speech.

Steganography, on the other hand, was also historically used for malicious purposes. The problem, by definition, is that no one will ever detect successful use of steganography so the spread of its usage is unknown. However, there are some rumors and confirmed cases. An article in 2001 *USA Today [62],* accused Muslim extremists of using steganography for planning terrorist activities. The suspicion that steganography was partly used to execute 9/11 attacks was brought up by several media – *The New York Times* [78], for example. The *NBC News* also reported [27] that Al-Qaeda uses steganographic techniques, and even the magazine Technical Mujahid [16] encouraged extremists to use them. Furthermore, Indian media [25] suspected the use of steganography behind the bombings on July 11, 2006.

Not only terrorists are suspected of using steganography; it can also be a part of illegal business practice. The British *The Independent [14]* mentioned that steganography was used for distributing child pornography. The people involved were caught in 2002 by an international effort led by the United Kingdom's National Hi-Tech Crime Unit. In 2008, a group of South American drug dealers used photographs of Hello Kitty to communicate the transit and shipment information[1].

An operation code-named *Shady Rat* was an industrial espionage ongoing between 2006 to 2011, reported by *McAfee*[2] and rumored to have originated in China. It used Trojan horses hidden in various document file types that were sent in personally written emails to specific individuals and companies. The remote site communicated with these viruses using commands steganographically embedded in digital images in order to avoid company firewalls. More than 70 companies and agencies working in crucial industries were hit worldwide.

In June 2010, FBI uncovered the largest Russian spy network in the United States since the end of the Cold War. As a result of this operation, ten Russian spies were expelled from the United States. According to legal documents of U.S. Department of Justice,[3] digital images posted on the Internet were used to conceal steganographic communication with the Russian intelligence agency. It was the most publicized use of steganography covered by reputable media, such as *The Washington Post* [84].

[1] Report: http://afp.google.com/article/ALeqM5ieuIvbrvmfofmOt8o0YfXzbysVuQ.
[2] Report: http://www.mcafee.com/us/resources/white-papers/wp-operation-shady-rat.pdf.
[3] Official document: http://www.justice.gov/opa/documents/062810complaint2.pdf.

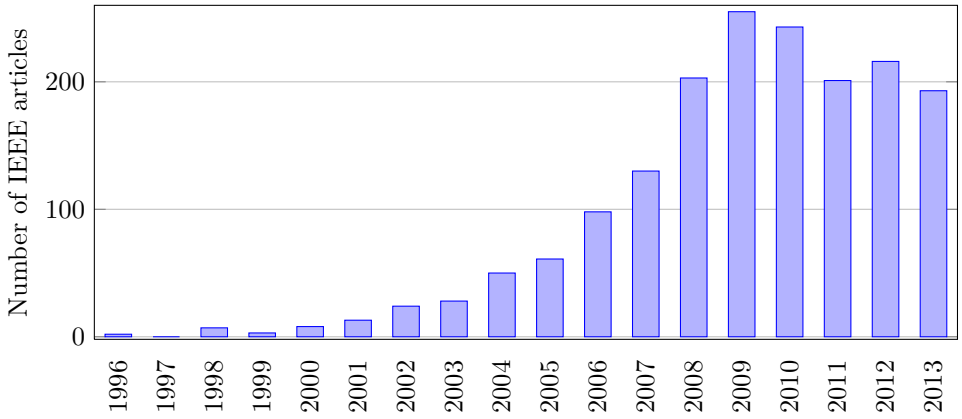

Figure 1.2.1: Annual number of IEEE journal and conference articles containing the words 'steganography' or 'steganalysis'.

The paragraphs above show that steganography can be used as a technique of choice for concealing a wide spectrum of both legal and illegal activities. Notice that the majority of the above mentioned examples used digital images as a medium for covering steganography. It is no surprise since digital images are easy to manipulate, and given the number of images on the Internet (Facebook reports 350 million image uploads per day as of 2013), it is easy to hide the modified image among the rest. Moreover, the majority of thousands steganographic tools available on-line are only for digital images, either compressed or raw [61].

Figure 1.2.1 shows the interest of the scientific community in steganography and steganalysis. The expansion of the Internet and a wide-spread use of digital images around year 2000 started the explosion of steganography related articles. However, since 2010 as the field matured and theoretical foundations were laid down, the interest of scientists appears more or less constant.

1.3 The prisoners' problem

In 1983, Simmons [101] studied a hypothetical misuse of communication channels serving for mutual inspection of nuclear missiles. These channels were provided according to the SALT disarmament treaty signed by both Cold War powers. In his research, he defined modern steganography and steganalysis using the so-called prisoners' problem.

Two steganographers, Alice and Bob, are locked in separate prison cells. They know beforehand of their separation, so they agree on a communication strategy for planning the escape. The prisoners are allowed to send messages, however, all their communication is observed by a prison warden named Eve. If the warden finds out that they are planning to escape or even suspects so, she would cut their communication and send Alice and Bob to solitary confinement.

The prisoners agreed to use steganography as the mean for secret communication. The Eve's approach of developing statistical tests for detecting secret messages is called *steganalysis*. In this scenario, the steganographic system is broken when Eve finds out that Alice and Bob are secretly communicating. In particular, the warden does not have to decode the message ,which makes steganalysis fundamentally different from cryptanalysis. Moreover, it is assumed that Eve is the so-called *passive warden* – the steganalyst passively monitors the channel but does not manipulate the messages. An *active warden* would be, for example, allowed to modify the pixels in images to destroy any potential hidden message.

Simmons also imported the Kerckhoffs' principle from cryptography – the assumption that Eve knows every possible steganographic algorithm Alice and Bob might use, except for the secret stego key. This is reasonable since the history of warfare and espionage shows that the embedding algorithm or device can easily fall into enemy's hands. Therefore, the steganographer is forced to abandon the principle of *security through obscurity* and rely solely on the stego key instead.

Chapter 2

Preliminaries

This chapter introduces the principles, notation, and elementary building blocks for the following chapters. The formalization of the steganographic channel, generally described in Section 1.3, is defined in Section 2.1. Notation consistently used in the entire dissertation is introduced in Section 2.2. This dissertation also contains many experiments, therefore, a brief description of image databases used in experiments, and the proposed steganographic methods and steganalysis feature sets can be found in Section 2.3. Finally, basic principles of the FLD-ensemble classifier used for all security evaluations appear in Section 2.4.

2.1 The steganographic channel

The goal of steganography is to communicate secret messages without revealing the very existence of the secret communication. This can be achieved by hiding the messages in inconspicuous objects. The focus of this dissertation is on steganography by *cover modification*, where the *covers* are grayscale digital images in either spatial or JPEG domain. Grayscale images are used for simplification as the majority of the principles can be extended to color images as well. Other known types of steganography are steganography by *cover selection* and by *cover synthesis* (see Section 4.1 and 4.2 in Ref. [38]).

We will now formalize the prisoners' problem mentioned in Section 1.3. Before Alice and Bob went to prison, they agreed on using grayscale images as cover objects; they also designed a message-hiding and a message-extraction algorithm, and agreed on a private key. The message is usually embedded by modifying the pixels or DCT coefficients of the cover image, which creates the so-called *stego* image. Prisoners send their stego images through a communication channel completely controlled by the warden Eve. It is assumed that every message is a random uncorrelated bit stream, which is a reasonable assumption since many compression methods provide this output.

Given the sets

$$\mathcal{C} \ldots \text{set of cover objects } \mathbf{X} \in \mathcal{C}, \tag{2.1.1}$$
$$\mathcal{K}(\mathbf{X}) \ldots \text{set of all stego keys for } \mathbf{X}, \tag{2.1.2}$$
$$\mathcal{M}(\mathbf{X}) \ldots \text{set of all messages possibly communicated in } \mathbf{X}. \tag{2.1.3}$$

The steganographic scheme used by Alice and Bob (visualized in Fig. 2.1.1) consists of the following embedding and extraction functions

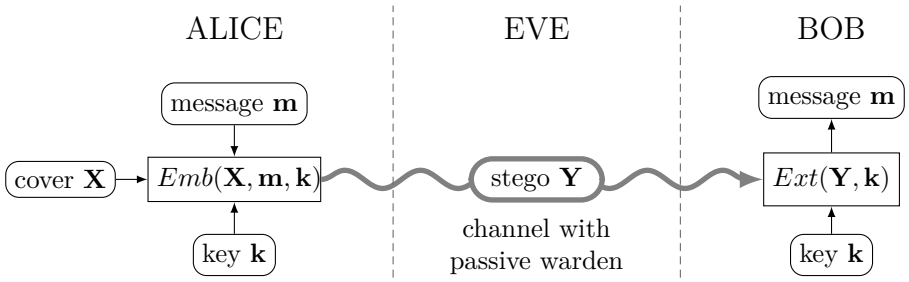

Figure 2.1.1: Components of the steganographic channel.

$$\mathrm{Emb} : \mathcal{C} \times \mathcal{M} \times \mathcal{K} \to \mathcal{C}, \tag{2.1.4}$$

$$\mathrm{Ext} : \mathcal{C} \times \mathcal{K} \to \mathcal{M}, \tag{2.1.5}$$

which meet the condition that $\forall \mathbf{X} \in \mathcal{C}$, $\forall \mathbf{k} \in \mathcal{K}(\mathbf{X})$, and $\forall \mathbf{m} \in \mathcal{M}(\mathbf{X})$:

$$\mathrm{Ext}\left(\mathrm{Emb}\left(\mathbf{X}, \mathbf{k}, \mathbf{m}\right), \mathbf{k}\right) = \mathbf{m}.$$

Eve performs steganalysis by designing a detector that attempts to distinguish between the cover image $\mathbf{X}$ and the stego image $\mathbf{Y} = \mathrm{Emb}\left(\mathbf{X}, \mathbf{k}, \mathbf{m}\right)$. The cover image $\mathbf{X}$ is a realization of a random variable that follows a distribution P_c over $\mathcal{C}$ and stego images follow a distribution P_s over $\mathcal{C}$. Naturally, if both distributions are identical, no statistical test can distinguish between innocent and steganographically modified images. Cachin [15] formalized the closeness of both distributions using the well-established Kullback–Leibler (KL) divergence [23] defined as

$$D_{\mathrm{KL}}(P_{\mathrm{c}}||P_{\mathrm{s}}) = \sum_{\mathbf{X} \in \mathcal{C}} P_{\mathrm{c}}(\mathbf{X}) \log \frac{P_{\mathrm{c}}(\mathbf{X})}{P_{\mathrm{s}}(\mathbf{X})}. \tag{2.1.6}$$

A steganographic system is called ϵ-*secure* when $D_{\mathrm{KL}}(P_{\mathrm{c}}||P_{\mathrm{s}}) \leq \epsilon$. Moreover, if $D_{\mathrm{KL}}(P_{\mathrm{c}}||P_{\mathrm{s}}) = 0$ the system is called *perfectly secure*. The KL divergence (2.1.6) provides a theoretical limit for the best possible detection performance of Eve's detector. Note that the security of the system depends not only on the embedding algorithm but also on the *cover source* $S_{\mathrm{c}} = \{\mathcal{C}, P_{\mathrm{c}}\}$ and the *message source* $S_m = \{\mathcal{M}, P_{\mathrm{m}}(\mathrm{S_c})\}$. Intuitively, the detectability depends on the message length – longer messages require more changes to the cover image causing a better detectability. The cover source also determines the steganographic security. Therefore, to maximize the security the steganographic schemes should be designed with respect to a specific cover source.

In practice, steganalysis may employ supervised machine learning (classifier) as a detector. The steganalyst designes a set of statistical features to distinguish cover and stego images – the features are thus inevitably designed and optimized for a given image database. Features extracted from the image database are usually separated into two subsets for training and testing. The classifier learns the decision boundary on a training set and then applies this boundary to the testing set while evaluating its detection performance.

2.2 Notation

2.2.1 General

This section defines the notation used consistently throughout the entire dissertation. Capital and lower-case boldface symbols stand for matrices and vectors, respectively. The calligraphic font is

reserved for sets. For a random variable X, its expected value is denoted as $E[X]$. The symbols $\mathbf{X} = (X_{ij}), \mathbf{Y} = (Y_{ij}) \in \mathcal{I}^{n_1 \times n_2}$ will always be used for a cover (and the corresponding stego) image with $n_1 \times n_2$ elements attaining values in a finite set $\mathcal{I}$. The image elements will be either 8-bit pixel values, in which case $\mathcal{I} = \{0, \dots, 255\}$, or quantized JPEG DCT coefficients, $\mathcal{I} = \{-1024, \dots, 1023\}$, arranged into an $n_1 \times n_2$ matrix by replacing each 8×8 pixel block with the corresponding block of quantized coefficients. For simplicity and without loss on generality, we will assume that n_1 and n_2 are multiples of 8. For a set of L centroids, $\mathcal{Q} = \{q_1, \dots, q_L\}$, $q_1 \leq \dots \leq q_L$, a scalar quantizer is defined as $Q_{\mathcal{Q}}(x) \triangleq \arg\min_{q \in \mathcal{Q}} |x - q|$. For matrix $\mathbf{A}$, $\mathbf{A}^{\mathrm{T}}$ is its transpose, and $|\mathbf{A}| = (|a_{ij}|)$ is the matrix of absolute values.

The performance of steganalysis is evaluated by reporting either the minimum total detection error under equal priors

$$P_{\mathrm{E}} = \min_{P_{\mathrm{FA}}} \frac{1}{2} \left(P_{\mathrm{FA}} + P_{\mathrm{D}}(P_{\mathrm{FA}}) \right) \tag{2.2.1}$$

or by reporting the detection performance of the ensemble classifier (see Section 2.4 or [77] for more details) using the out-of-bag (OOB) estimate of the testing error P_{E} defined in (2.2.1). This error, which we denote E_{OOB}, is known to be an unbiased estimate of the testing error on unseen data.

2.2.2 JPEG steganography

For side-informed JPEG steganography, a precover (raw, uncompressed) image will be denoted as $\mathbf{P} = (P_{ij}) \in \mathcal{I}^{n_1 \times n_2}$. When compressing $\mathbf{P}$, first a blockwise DCT transform is executed for each 8×8 block of pixels from a fixed grid. Then, the DCT coefficients are divided by quantization steps and rounded to integers. Let $\mathbf{P}^{(b)}$ be the bth 8×8 block when ordering the blocks, e.g., in a row-by-row fashion ($b = 1, \dots, n_1 \cdot n_2/64$). With a luminance quantization matrix $\mathbf{Q} = \{q_{kl}\}$, $1 \leq k, l \leq 8$, we denote $\mathbf{D}^{(b)} = \mathrm{DCT}(\mathbf{P}^{(b)})./\mathbf{Q}$ the raw (non-rounded) values of DCT coefficients. Here, the operation $'./'$ is an elementwise division of matrices and DCT(.) is the DCT transform used in the JPEG compressor. Furthermore, we denote $\mathbf{X}^{(b)} = [\mathbf{D}^{(b)}]$ the quantized DCT coefficients rounded to integers. We use the symbols $\mathbf{D}$ and $\mathbf{X}$ to denote the arrays of all raw and quantized DCT coefficients when arranging all blocks $\mathbf{D}^{(b)}$ and $\mathbf{X}^{(b)}$ in the same manner as the 8×8 pixel blocks in the uncompressed image. We will use the symbol $J^{-1}(\mathbf{X})$ for the JPEG image represented using quantized DCT coefficients $\mathbf{X}$ when decompressed to the spatial domain.[1]

We would like to point out that the JPEG format allows several different implementations of the DCT transform, DCT(.). The specific choice of the transform implementation may especially impact the security of side-informed steganography. In this dissertation, we work with the DCT(.) implemented as 'dct2' in Matlab when feeding in pixels represented as 'double'. In particular, a block of 8×8 DCT coefficients is computed from a precover block $\mathbf{P}^{(b)}$ as

$$\mathrm{DCT}(\mathbf{P}^{(b)})_{kl} = \sum_{i,j=0}^{7} \frac{w_k w_l}{4} \cos \frac{\pi k(2i+1)}{16} \times \cos \frac{\pi l(2j+1)}{16} P_{ij}^{(b)}, \tag{2.2.2}$$

where $k, l \in \{0, \dots, 7\}$ index the DCT mode and $w_0 = 1/\sqrt{2}$, $w_k = 1$ for $k > 0$.

To obtain an actual JPEG image from a two-dimensional array of quantized coefficients $\mathbf{X}$ (cover) or $\mathbf{Y}$ (stego), we first create an (arbitrary) JPEG image of the same dimensions $n_1 \times n_2$ using Matlab's 'imwrite' with the same quality factor, read its JPEG structure using Sallee's Matlab JPEG Toolbox[2] and then merely replace the array of quantized coefficients in this structure with $\mathbf{X}$ and $\mathbf{Y}$ to obtain the cover and stego images, respectively. This way, we guarantee that both images were created using the same JPEG compressor and that all that we will be detecting are the embedding changes rather than compressor artifacts.

[1] The process J^{-1} involves rounding to integers and clipping to the dynamic range $\mathcal{I}$.

[2] http://dde.binghamton.edu/download/jpeg_toolbox.zip

2.3 Experimental core

2.3.1 Image databases

In this dissertation, we carry out most of our experiments on two image sources. The first is the standardized database called BOSSbase 0.92 respectively 1.01 [4]. This source contains 9074 respectively 10,000 images acquired by seven digital cameras in the RAW format (CR2 or DNG) and subsequently processed by converting to 8-bit grayscale, resizing, and cropping to the size of 512×512 pixels. The script for this processing is also available from the BOSS competition web site.[3]

The second image source, Leica database, was obtained using the Leica M9 camera equipped with an 18-megapixel full-frame sensor. A total of 3,000 images were acquired in the raw DNG format, demosaicked using UFRaw[4] (with the same settings as the script used for creating BOSSbase), converted to 8-bit grayscale, and finally central-cropped to the size of 512×512. This second source is very different from BOSSbase 1.01 and was intentionally included as an example of imagery that has not been subjected to resizing, which has been shown to have a substantial effect on the detectability of embedding changes in the spatial domain [75]. By adjusting the image size of Leica images to that of the BOSSbase, we removed the effect of the square root law [67] on steganalysis, allowing interpretations of experiments on both sources.

For JPEG experiments, the databases were JPEG-compressed with standard quantization tables corresponding to quality factors 75, 85, and 95 using the algorithm described in Section 2.2.2.

2.3.2 Steganographic algorithms

In addition to algorithms devised in this dissertation, other methods are also used for comparison. In general, steganographic algorithms are divided into three types depending on the embedding domain and available information: spatial, JPEG, and side-informed JPEG. Spatial algorithms embed messages by modifying pixel values while JPEG algorithms embed into quantized DCT coefficients. Side-informed algorithms utilize the knowledge of the uncompressed image (a precover) and therefore the knowledge of non-quantized DCT coefficients and rounding errors. All steganographic algorithms used in this dissertation are described below.

Spatial domain:

- LSB matching – simple non-adaptive ± 1 embedding implemented with ternary matrix embedding. For more details, see Chapter 8 in Ref. [38].
- Edge-Adaptive (EA) [82] (Luo, 2010) – this algorithm confines the embedding changes to pixel pairs whose difference in absolute value is as large as possible (e.g., around edges).
- HUGO [90] (Pevný, 2010) – the first modern content-adaptive steganographic algorithm utilizing syndrome-trellis codes [35]. It was designed to minimize the embedding distortion in a high-dimensional feature space computed from differences of four neighboring pixels. Its embedding simulator was run with the swich --T 255 to remove the weakness discovered during the competition [4].
- HUGO BD [33] (Filler, 2010) – a modification of HUGO, in which a non-additive distortion is computed only from local neighborhoods to allow the use of the Gibbs construction [33] and ternary embedding.

[3] http://exile.felk.cvut.cz/boss/BOSSFinal/index.php?mode=VIEW&tmpl=home
[4] http://ufraw.sourceforge.net

- WOW [54] (Holub, 2012) – a highly content-adaptive scheme utilizing wavelet filter banks to evaluate the embedding distortion. Unlike HUGO, it is designed specifically to avoid making embedding changes in well modelable content.

- S-UNIWARD [59] (Holub, 2013) – as spatial domain instance of UNIWARD distortion similar to WOW, it is wavelet-based and further described in Chapter 5.

JPEG domain:

- Jsteg [105] (Upham, 1993) – embeds into DCT coefficients using the least significant bit (LSB) replacement and avoiding zeroes and ones.

- OutGuess [93] (Provos, 2001) – embeds in two phases: message embedding and histogram correction consequently preserving the histogram of DCT coefficients.

- nsF5 [47] (Fridrich, 2007) – a non-shrinkage F5 is a version of Westfeld's F5 scheme from 2001 [108] with improved coding. It changes DCT coefficients only in the direction towards zero.

- UED [51] (Guo, 2012) – its distortion function utilizes inverse values of DCT coefficients and their intra-block and inter-block neighbors. The idea is to achieve a uniform spread of embedding changes over different values of DCT coefficients.

- J-UNIWARD [59] (Holub, 2013) – a JPEG domain instance of UNIWARD distortion. It utilizes a spatial distortion for JPEG steganography and is described in Chapter 5.

Side-informed JPEG domain:[5]

- BCHopt [95] (Sachnev, 2009) – BCH codes are employed to minimize the embedding distortion in the DCT domain. BCHopt is an improved version of BCH that contains a heuristic optimization and also hides message bits into zeros.

- NPQ [60] (Huang, 2012) – the Normalized PQ was chosen over older versions of the Perturbed Quantization (PQ) algorithm [47] based on the superiority of NPQ over PQ reported in [60]. The improvement over PQ is achieved by dividing the embedding cost by the absolute value of the DCT coefficient.

- Square (served as reference in [59]) – the embedding cost of changing the ij-th DCT coefficient corresponding to the DCT mode k, l by ± 1: $\rho_{ij}^{(kl)} = (q_{kl}(1 - 2|e_{ij}|))^2$. Here, q_{kl} is the quantization step of the kl-th mode and e_{ij} is the quantization error when rounding the DCT coefficient obtained from the precover image during JPEG compression.

- EBS [107] (Wang, 2012) – Entropy-based steganography is basically a square distortion weighted by the block entropy.

- SI-UNIWARD [59] (Holub, 2013) – Side-informed JPEG domain instance of UNIWARD distortion. It is basically a J-UNIWARD distortion multiplied by the rounding error of a given DCT coefficient. For more details see Chapter 5.

[5] Note that some side-informed algorithms are modified so they avoid embedding in DCT modes $(0,0)$,$(0,4)$,$(4,0)$ and $(4,4)$ when the unquantized value is equal to $k + 0.5$, $k \in \mathbb{Z}$. The reason for this modification can also be found in Chapter 5 and Ref. [59].

2.3.3 Feature sets

Since the idea of steganalysis by supervised classification was introduced, many feature sets utilizing different statistical properties of digital images were devised. Historically, features were designed separately for detection of spatial domain and JPEG domain steganography. However, this dissertation shows that the difference between spatial and JPEG domain steganography is blurred (5). This difference also disappears in steganalysis as some JPEG domain algorithms are better detected in the spatial domain ([58] and Chapter 7). All feature sets in this dissertation are high dimensional (dimensionality from 12,000 to 35,000) as the ensemble classifier [77] can easily handle such high dimensional feature sets.

Spatial domain features:

- SRMQ1 [43] (Fridrich, 2011, dim. 12,753) – this feature set is exploits dependencies between different noise residuals created by multiple different linear and non-linear high pass filters. The noise residual is quantized, thresholded, rounded, and co-occurrences are built to capture the dependencies. The design of residuals and the features is explained in Section 4.2.
- SRM [43] (Fridrich, 2011, dim. 34,671) – an extended version of SRMQ1 features which uses three different quantization values (1, 1.5, 2) to increase feature diversity.
- PSRMQ1 [57] (Holub, 2013, dim. 12,870) – a feature set using the histograms of projections of SRMQ1's noise residuals to capture the dependencies instead of co-occurrences. This set consequently achieves a superior performance and enjoys a smaller dimensionality at the cost of increased computational complexity. This feature set is described in Section 4.3.

JPEG domain features:

- JRM [74] (Kodovský, 2012, dim. 22,510) – a cross-calibrated feature set consisting of many diverse sub-models capturing intra-block and inter-block dependencies among DCT coefficients.
- JSRM (dim. 35,253) – a merger of spatial domain SRMQ1 and JRM.
- PSRMQ3 [57] (Holub, 2013, dim. 12,870) – a version of the spatial domain PSRMQ1 with a different quantization step for better detection of JPEG domain steganography.
- JPSRM (dim. 35,380) – a merger of PSRMQ3 and JRM.
- DCTR (dim. 8,000) – A novel and faster approach to steganalysis using histograms of residual values, which are extracted from the spatial domain by convolving it with all 64 DCT basis. This feature set is introduced in Chapter 8 for the first time.

2.4 Ensemble classifier

Due to high complexity of support vector machines for supervised classification, the machine learning of choice in this dissertation is the ensemble classifier [77]. For problems in steganalysis with a large number of very weak features and cover-stego training pairs in the training set, it offers a comparable performance for a fraction of the computational costs when compared with support vector machines. The significantly lower training complexity allows the steganalyst to design high-dimensional statistical features and train on larger training sets, consequently greatly improving detection of modern steganographic schemes. This section provides explanation of ensemble's basic principles.

The ensemble classifier consists of multiple base learners independently trained on a database of cover and stego images. The base learners are simple classifiers built on a uniformly randomly

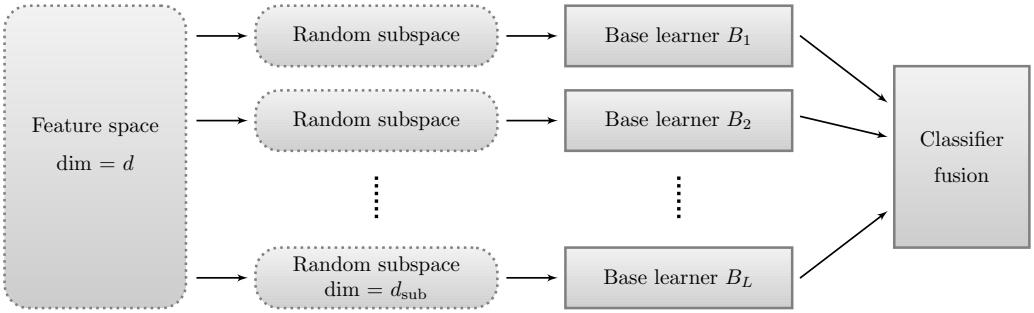

Figure 2.4.1: Diagram illustrating the ensemble classifier. The random subspaces are chosen randomly and uniformly from the entire feature space. This image is acquired from Ref. [77].

selected subspace of the feature space – the final decision is formed by aggregating the decisions of individual base learners (see Fig. 2.4.1). This aggregation strategy works only if the individual base learners are sufficiently diverse, meaning that they make different errors on unseen data. Each base learner is also trained on a bootstrap sample of the training set (a uniform sample with replacement) in order to further increase the diversity of the training set.

This approach is known in the machine learning community as bootstrap aggregating or bagging [12]. Furthermore, it also allows us to obtain an accurate estimate of the testing error, which is important for determining optimal ensemble parameters. We note that the bootstrap samples are formed "by pairs," i.e., we make sure that the pairs of cover features and the corresponding stego features are both contained in the bootstraps. This modification specific for steganalysis is important as it has been shown that breaking the cover-stego pairs into two sets, one of which is used for training and the other, testing, one for error estimation, possibly leads to a biased error estimate and suboptimal performance [96, 69].

Only about 63% of unique samples are included in each bootstrap due to the sampling with replacement, so the remaining 37% unused samples can be utilized for detection evaluation and estimation of the optimal dimensionality of random subspaces d_{sub} and the number of base learners L. The ensemble article [77] shows that the out-of-bag (OOB) detection error estimate computed from these unused samples is an unbiased estimator of the testing error and it is extensively used in this dissertation to report the detection of steganographic schemes using the ensemble classifier. The OOB estimate does not require any testing set and allows us to use larger image databases for the training phase. The search for optimal values of d_{sub} and L stops when the OOB estimate saturates while changing d_{sub} and increasing L.

All L individual base learners are mappings $\mathbb{R}^{d_{\text{sub}}} \to \{0, 1\}$, where 0 means cover and 1 stego image. Decreasing the dimensionality from full dimensionality d, to $d_{\text{sub}} \ll d$ significantly lowers the computational complexity. Even though the individual base learners can be weak, the accuracy quickly improves after fusion and eventually levels out for a sufficiently large L. The decision threshold of each base learner is adjusted to minimize P_{E} (2.2.1).

After collecting all L decisions, the final classifier output is formed by combining them using the majority voting strategy – the sum of the individual votes is compared to the decision threshold $L/2$. This threshold can be adjusted to obtain the ROC curve, however, the threshold $L/2$ represents P_{E}(2.2.1) which became a standard method of evaluating detection accuracy in steganalysis.

We recommend to implement each base learner as the Fisher Linear Discriminant (FLD) [26] because of its low training complexity. Additionally, such weak and unstable classifiers desirably increase diversity. Let us have a matrix of cover features $\mathbf{f}_C$ and stego features $\mathbf{f}_S$ of size $N \times d_{\text{sub}}$, where N is the number of training samples and $\mathbf{f}_{C_i}$ denotes i-th row of $\mathbf{f}_C$. The FLD base learner is fully

described using the generalized eigenvector

$$\mathbf{v}_l = (\mathbf{S}_{\mathrm{W}} + \lambda \mathbf{I})^{-1}(\boldsymbol{\mu}_C - \boldsymbol{\mu}_S), \tag{2.4.1}$$

where $\boldsymbol{\mu}_C, \boldsymbol{\mu}_S \in \mathbb{R}^{d_{\mathrm{sub}}}$ are the means of each class

$$\boldsymbol{\mu}_C = \frac{1}{N}\sum_{i \in N} \mathbf{f}_{C_i}, \quad \boldsymbol{\mu}_S = \frac{1}{N}\sum_{i \in N} \mathbf{f}_{C_i}, \tag{2.4.2}$$

$$\mathbf{S}_{\mathrm{W}} = \sum_{i \in N} (\mathbf{f}_{C_i} - \boldsymbol{\mu}_C)(\mathbf{f}_{C_i} - \boldsymbol{\mu}_C)^{\mathsf{T}} + (\mathbf{f}_{S_i} - \boldsymbol{\mu}_S)(\mathbf{f}_{S_i} - \boldsymbol{\mu}_S)^{\mathsf{T}} \tag{2.4.3}$$

is the within-class scatter matrix, and λ is a stabilizing parameter to make the matrix $\mathbf{S}_{\mathrm{W}} + \lambda \mathbf{I}$ positive definite and thus avoid problems with numerical instability in practice when S_{W} is singular or ill-conditioned.

Chapter 3

Content adaptive steganography

In this chapter, we introduce a relatively new concept of content adaptive steganography developed in late 2000s. Its general goal is to embed a message while minimizing a given distortion function. Content adaptive steganography is closely related to coding and its development therefore took off with practical coding schemes, such as MMx [68] and especially syndrome trellis codes (STCs) [35] – fast, user-friendly implementation wich performance arbitrarily close to the theoretical rate-distortion bound. Furthermore, development in coding theory allowed steganographers to evaluate coding efficiency and simulate optimal embedding instead of utilizing actual coding schemes.

The general concept of contend adaptive steganography and distortion function is introduced in Section 3.1; the latter is formalized in Section 3.2. Basic theoretical concepts necessary for understanding the topic, such as the rate-distortion bound and coding efficiency, are defined in Section 3.3. The simulator of optimal embedding and near-optimal STCs coding scheme are briefly described in Section 3.4. Finally, Section 3.5 contains a short reflection on the future and challenges of content adaptive steganography.

3.1 Concept

There are two ways of designing good steganographic algorithms for digital images. The fist method relies on a defined cover image model, which is preserved by steganography. Such steganography will be perfectly secure with respect to this model, however, since no complete model of digital images exists [8], this approach can be usually well detected by steganalysis working outside of the model.

The second and modern approach is steganography by minimizing some, usually heuristically-defined, embedding distortion function. It completely abandons the idea of perfectly secure steganography and embeds the message while introducing only the smallest possible distortion to the cover image. This approach is more flexible and enables development of steganographic methods driven by the detection performance of steganalysis. In fact, the majority of currently most secure distortion functions are designed with the help of these heuristic principles [90, 51, 54, 59]. For these steganographic algorithms, the distortion introduced by the individual embedding changes depends on the evaluation of local image content, consequently making more embedding changes in textured areas and less in smooth areas. Hence the name *content adaptive* steganography.

The distortion based approach can alse be used to minimize the difference between feature vectors extracted from cover and stego objects, connecting this approach to model preservation. However, the same problem as with model preservation appears since no feature space (model) is complete and all the embedding changes will be executed outside of the model, where the steganography becomes detectable. A prime example of this flawed approach is MOD [34] steganography in JPEG domain, which minimized the distortion with respect to CC-PEV [91, 71]. It was shown in Ref. [76] that

a simple change of the CC-PEV feature model by adjusting the co-occurrence threshold parameter makes MOD extremely detectable.

3.2 Distortion function

We will now closely follow the notation and definitions introduced in Ref. [35] and [33]. The message is communicated by introducing a small modification to the cover image $\mathbf{X} = (x_1, \ldots, x_n) \in \mathcal{X} = \{\mathcal{I}\}^n$, thus creating a stego image $\mathbf{Y} = (y_1, \ldots, y_n) \in \mathcal{Y} = \mathcal{I}_1 \times \ldots \times \mathcal{I}_n$, where $\mathcal{I}_i \in \mathcal{I}$ such that $x_i \in \mathcal{I}_i$. If the cardinality $\text{card}\,(\mathcal{I}_i) = 2, \forall i \in \{1, \ldots n\}$, then the embedding operation is called binary. If $\text{card}\,(\mathcal{I}_i) = 3$, the embedding operation is ternary. An example of a ternary operation is ± 1 embedding (LSB matching), which can be represented as $\mathcal{I}_i = \{x_{i-1}, x_i, x_{i+1}\}$ with obvious modifications at the boundary of the image dynamic range.

A distortion function measures the impact of embedding changes. The sender embeds his message while minimizing

$$D : \mathcal{Y}(\mathbf{X}) \to \mathbb{R}. \tag{3.2.1}$$

Except for Chapter 6, the distortion function in this dissertation is limited to the additive form

$$D(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n} \rho_i(\mathbf{X}, y_i) \tag{3.2.2}$$

where $\rho_i : \mathcal{X} \times \mathcal{I}_i \to [-K, K]$, $0 < K < \infty$, are bounded functions determining the cost of replacing the cover pixel x_i with y_i. This form accommodates for dependencies between pixels because ρ_i depends on the whole cover image $\mathbf{X}$. It is important to note that in this additive form the value of $\rho_i(\mathbf{X}, y_i)$ does not depend on changes made at other pixels – the embedding changes do not interact. In practice, we expand the domain of ρ_i to $\mathcal{X} \times \mathcal{I}$ and define $\rho_i(\mathbf{X}, y_i) = \infty$ to cases when $y_i \notin \mathcal{I}_i$.

3.3 Theoretical bounds

In this section, we state the theoretical bounds of content adaptive steganography – embedding while minimizing a distortion function.

Encryption or compression algorithms convert every message into a pseudo-random bit stream – it is assumed that messages are pseudo-random bit streams. Furthermore, we assume that every steganographic scheme in this dissertation is associated with a mapping that assigns to each cover $\mathbf{X}$ a pair $\{\mathcal{Y}, \pi\}$, where $\mathcal{Y}$ is the set of all stego images into which $\mathbf{X}$ can be modified and π is their probability distribution characterizing the actions of the sender. Therefore, the stego image is a random variable Y over $\mathcal{Y}$ with the distribution $P(Y = \mathbf{Y}) = \pi(\mathbf{Y})$. Simply put, the embedding algorithm takes a given cover image $\mathbf{X}$ and outputs stego image $\mathbf{Y} \in \mathcal{Y}$ with probability $\pi(\mathbf{Y})$. The set $\mathcal{Y}$ and all concepts derived from it in this section depend on $\mathbf{X}$ – the cover image is just a parameter we can fix in the very beginning, so we do not make the dependence on $\mathbf{X}$ explicit in the notation, which is why we will write $D(\mathbf{Y})$ instead of $D(\mathbf{X}, \mathbf{Y})$.

The maximal expected payload that the sender can communicate in this manner is the entropy

$$H(\pi) \triangleq H(Y) = -\sum_{\mathbf{Y} \in \mathcal{Y}} \pi(\mathbf{Y}) \log \pi(\mathbf{Y}) \tag{3.3.1}$$

while introducing average distortion

$$E_\pi [D] = \sum_{\mathbf{Y} \in \mathcal{Y}} \pi (\mathbf{Y}) D (\mathbf{Y}). \tag{3.3.2}$$

The whole concept of content adaptive steganography by minimizing a distortion function has a strong assumption – the distortion function must be related to statistical detectability, therefore its minimization should improve steganographic security. This assumption is not as trivial as shown in Chapter 6 as there is no practical and general measure of steganographic security.

Two related frameworks exist for message embedding. A distortion-limited sender assumes a fixed allowed distortion and maximizes the length of the message that can be communicated within that distortion. A payload-limited sender has a fixed message length (payload) m and minimizes the embedding distortion. Only the latter framework will be described since only the payload-limited sender is used in this dissertation.

The optimization problem is defined as follows:

$$\underset{\pi}{\text{minimize}} \quad E_\pi [D] = \sum_{\mathbf{Y} \in \mathcal{Y}} \pi (\mathbf{Y}) D (\mathbf{Y}) \qquad \text{subject to} \quad H (\pi) = m. \tag{3.3.3}$$

It is known that the optimal distribution of π for this problem has the Gibbs form

$$\pi_\lambda (\mathbf{Y}) = \frac{1}{Z (\lambda)} \exp (\lambda D (\mathbf{Y})) \overset{(a)}{=} \frac{1}{Z (\lambda)} \prod_{i=1}^{n} \exp (\lambda \rho_i (y_i)) \triangleq \prod_{i=1}^{n} \pi_i (y_i), \tag{3.3.4}$$

where $Z (\lambda)$ is the normalizing factor

$$Z (\lambda) = \sum_{\mathbf{Y} \in \mathcal{Y}} \exp (\lambda D (\mathbf{Y})). \tag{3.3.5}$$

The parameter $\lambda \in [0, \infty)$ can be obtained from thr constraint (3.3.3) by a binary search due to monotonicity of $H (\pi)$ with respect to λ. Step (a) is possible due to additivity of the distortion function D, i.e. the mutual independence of stego pixels.

3.4 Optimal coding – simulation and syndrome-trellis codes

Optimal embedding with best possible π can be simulated by changing each pixel i with probability π_i (3.3.4). Steganographers can therefore design and test distortion functions against steganalysis without relying on practical coding schemes. The separation of distortion function design from actual message embedding enables a faster progress in development and testing of modern steganography. Furthermore, the simulator can also be used as an upper bound to compare the efficiency of coding schemes. For these reasons, unless written otherwise, steganography in this dissertation is performed with the embedding simulator rather than actual coding.

A practical coding algorithm,[1] based on syndrome-trellis codes that embeds near the payload–distortion bound was proposed in [35]. The implementation of STCs has a single parameter h (constraint height) that drives its efficiency with respect to the optimal rate-distortion bound defined by equations (3.3.1) and (3.3.2). Even though it comes with a cost of increased complexity, the STCs are applicable in practice even when its efficiency achieves 90 % of the rate-distortion bound.

The STC Toolbox implements not only binary, but also ternary (changes by ± 1), and pentary (changes by ± 2) embedding operation. Binary and ternary embedding and its simulation is widely used in this dissertation. More detailed description of STCs is not within the scope of this dissertation, please see the original paper for more information.

[1] The STC Toolbox can be downloaded from http://dde.binghamton.edu/download/syndrome/

3.5 Challenges

Due to the separation principle and near-optimal practical coding, the last problem left to researchers in steganography is the design of the distortion function D or embedding costs ρ_i (at least in the case of additive distortion function). This problem, however, is of utmost importance. The distortion function drives the location of embedding changes and has a major influence on steganographic security – consequently, a lot of space in this dissertation is dedicated to the distortion function design. The author believes that nobody knows how to design optimal distortion function for empirical objects, such as natural images and that this is still an open problem.

New trends have been appearing in steganalysis of content adaptive schemes suggesting that overly adaptive distortion functions [59, 54] assigning small embedding costs to particular small areas and large embedding costs to the rest of the image may have a weakness. If the steganalyst knows the used steganographic algorithm, she can quite precisely estimate the probability of embedding changes for individual pixels and extract the features only from those with high probability. This approach slightly improves steganalysis of these highly adaptive schemes compared to feature extraction from the whole image, especially for small payloads.

Distortion functions can be faulty, meaning that a targeted attack can be mounted against them. An example of such an attack on the first version of S-UNIWARD is in Ref. [103] and Chapter 5. This targeted attack can bring the detection error on S-UNIWARD from 20 % (SRM) to 1 %. With this detection error the steganographic scheme is considered broken.

Chapter 4

Spatial domain steganalysis

Steganalysis is the art of revealing the presence of secret messages embedded in objects. In this chapter we focus on the case when the original (cover) object is a digital image and the steganographer hides the message by slightly modifying the pixel values of the cover.

In general, a steganalysis detector can be built either using the tools of statistical signal detection or by applying a machine-learning approach. Both approaches have their strengths as well as limitations, which is the reason why they are both useful and will likely coexist in the foreseeable future. The former approach derives the detector from a statistical model of the cover source, allowing one to obtain error bounds on the detector performance. Normalized detection statistics are also less sensitive to differences between cover sources. On the other hand, to make this approach tractable, the adopted cover model must usually be sufficiently simple, which limits the detector optimality and the validity of the error bounds to the chosen cover model. Simple models, however, cannot capture all the complex relationships among individual image elements that exist in images of natural scenes acquired using imaging sensors. Moreover, this approach has so far been applied only to rather simple embedding operations, examples of which are the LSB (least significant bit) replacement and matching [32, 113, 22, 20], and may not be easily adapted to complex, content-adaptive embedding algorithms, such as HUGO [90], WOW [54], or the schemes based on UNIWARD (Chapter 5 and [56]). This is because attacking these schemes would require working with models that allow for complex dependencies among neighboring pixels. However, given the highly non-stationary character of natural images, estimating such local model parameters will likely be infeasible.

The latter approach to steganalysis does not need the underlying cover distribution to build a detector. Instead, the task of distinguishing cover and stego objects is formulated as a classification problem. First, the image is represented using a feature vector, which can be viewed as a heuristic dimensionality reduction. Then, a database of cover and the corresponding stego images is used to build the detector using standard machine learning tools. The principal advantage of this approach is that one can easily construct detectors for arbitrary embedding algorithms. Also, for a known cover source, such detectors usually perform substantially better than detectors derived from simple cover models. The disadvantage is that the error bounds can only be established empirically, for which one needs sufficiently many examples from the cover source. While such detectors may be inaccurate when analyzing a single image of unknown origin, steganographic communication is by nature repetitive and it is not unreasonable to assume that the steganalyst has many examples from the cover source and observes the steganographic channel for a length of time.

In this chapter (also in Chapter 8), we assume that the analyst knows the steganographic algorithm and sufficiently many examples from the cover source are available. Since the embedding changes can be viewed as an additive low-amplitude noise that may be adaptive to the host image content, we follow a long-established paradigm [114, 89, 43, 50] and represent the image using a feature computed

from the image noise component – the so-called noise residual.[1] To obtain a more accurate detection of content-adaptive steganography, various authors have proposed to utilize an entire family of noise residuals, obtaining thus what is now called rich image representations [43, 50, 46].

This chapter is focused on spatial (pixel) domain steganalysis and it is organized in the following way. The concept of a residual is explained in Section 4.1 together with the computation of co-occurrences. This section proposes a method for optimizing pixel predictors in order to achieve improvement in steganalysis for specific algorithms and cover sources. Due to computational limitations, this approach can not be applied to large-scale models. A set of carefully hand-designed predictors used for SRM [43] features is defined in Section 4.2. This feature set uses co-occurrence matrices to capture dependencies among the residual values. However, we show in Section 4.3 that an alternative approach using random projections can be applied to capture these dependencies to improve detection while using the same residuals.

4.1 Optimizing pixel predictors for steganalysis

This section is a slightly modified version of author's SPIE 2012 conference paper [55].

A standard way to design steganalysis features for digital images is to choose a pixel predictor, use it to compute a noise residual, and then form joint statistics of neighboring residual samples (co-occurrence matrices). This section proposes a general data-driven approach to optimizing predictors for steganalysis. First, a local pixel predictor is parametrized and then its parameters are determined by solving an optimization problem for a given sample of cover and stego images and a given cover source. Our research shows that predictors optimized to detect a specific case of steganography may be vastly different than predictors optimized for the cover source only. The results indicate that optimized predictors may improve steganalysis by a rather non-negligible margin. Furthermore, we construct the predictors sequentially – having optimized k predictors, design the $k+1$st one with respect to the combined feature set built from all k predictors. In other words, given a feature space (image model) extend (diversify) the model in a selected direction (functional form of the predictor) in a way that maximally boosts detection accuracy.

4.1.1 Introduction

Steganalysis features for digital images represented in the spatial domain are typically computed as joint or conditional probabilities of adjacent samples from a noise residual obtained using a pixel predictor. The purpose of working with the noise residual is to increase the SNR between the stego signal and the original image by suppressing the image content and to narrow the dynamic range of the resulting signal to allow its description using higher-order co-occurrence matrices. Even the early steganalysis algorithms, then called blind detectors, utilized predictors. The very first feature-based steganalyzer proposed in 2000 by Avcibas et al. [2] employed image quality measures whose values are largely dependent on the image noise component computed by subtracting from the image its low-pass filtered version (here, interpreted as a cover prediction). Farid [31] used a shift-invariant linear predictor of wavelet coefficients and formed the features as higher-order moments of marginals of the predictor error. Here, the predictor was chosen to minimize the mean square prediction error. Higher-order moments of noise residual obtained using a denoising filter were used as features by Holotyak [53] and Goljan [49] in WAM. The SPAM feature vector [89] as well as the features proposed in Ref. [114] were formed as Markov transition probabilities of differences between neighboring pixels, which are noise residuals obtained using a very simple predictor – the value of its immediate neighbor. Recently, the authors of Refs. [46, 45, 50] pointed out the importance of forming features from a wider class of noise residuals computed using many different pixel predictors that

[1] The idea to compute features from noise residuals has already appeared in the early works on feature based steganalysis [2, 31, 83, 49].

employed mostly local polynomial models. Pixel predictor is also the cornerstone of the quantitative weighted-stego LSB detector [41, 7, 65, 21].

It is thus only natural to ask whether detection performance can be improved and by how much by optimizing the predictor within a given detection framework. To this end, we need to narrow down the set of predictors within which the optimization will operate. One possibility is to use off-the-shelf denoising filters and optimize w.r.t. their parameters, such as the variance of the Gaussian noise being removed. Such predictors, however, put great weight on the central pixel being predicted, which leads to predictions biased by the stego signal. The subsequent subtraction of the stego image when forming the residual thus undesirably suppresses the stego signal. The prediction should only utilize the immediate neighborhood excluding the pixel being predicted.

A tempting idea is to fit a Markov random field model [111] to a given cover source and use as predictors the local characteristics, which are conditional probabilities of pixel values given their neighborhood. A simpler approach would be to minimize the prediction error on covers when measured in some appropriate manner, such as in the least square sense. However, predictors built only by considering the cover source and not the embedding may not perform well for detecting steganography, which is a binary decision problem rather than cover source modeling. Indeed, one could conceivably create an embedding method tailored to be undetectable (or only weakly detectable) in a given feature space, for example using the concept of feature correction [70, 18] or the paradigm introduced in Ref. [33] by suitably defining the distortion function. Optimal predictor will thus surely be a function of both the cover source and the embedding algorithm.[2]

In this section we restrict ourselves to particularly simple predictors in the form of a shift-invariant linear filter. The predictor will be applied globally to all images and will thus be dependent on a given cover source but not on the individual images. By parametrizing the predictor kernel,[3] we determine such values of the parameters that give the most accurate detection for a given cover source, steganography method, and detection framework. We are interested in how much the detection can be improved over previously proposed constructs, such as kernels designed to minimize the square prediction error and the heuristically derived predictors introduced in Ref. [46, 45]. Improved pixel predictors will lead to more accurate steganalysis for a given feature dimensionality and will enable construction of more compact rich models obtained by merging several diverse feature sets [43]. To this end, we study "conditional design" of the predictors to increase the diversity by optimizing the predictor w.r.t. a collection of existing ones.

The author does not necessarily view the fact that the predictor will be optimized w.r.t. a given source and stego method as a deficiency. Studying steganalysis in a given source may provide useful insight and even improve methods that aspire to be universal as such systems may be built as a collection of steganalyzers designed for selected classes of cover sources supplied with a source classifier. Moreover, we argue that in the past numerous if not all steganalysis features were designed for a specific embedding paradigm and source even though the authors may not have openly stated this fact. For example, the SPAM feature vector [89] was seemingly proposed from a pure cover model but in reality it is driven by a specific case of steganography as well as cover source. The choice of the order of the Markov process as well as the threshold T and the local predictor were driven by observing the detection performance on ± 1 embedding on a fixed database of images. Had the authors used HUGO [90] instead of ± 1 embedding or larger images, different parameters of the SPAM feature would have been selected. In particular, in light of the recent work [46, 45, 50], the predictor would have probably used second- or third-order differences instead of the first-order difference and the order of the Markov chain model might have been four instead of three to leverage longer-range pixel dependencies.

This section has the following structure. In Subsection 4.1.2, we describe the used steganalysis features. This same subsection also explains the parametrization of the predictor kernels and a method for optimizing the detection performance w.r.t. to the kernel parameters. The first set of

[2] The problem of optimizing the predictor for universal steganalysis is not investigated in this dissertation due to the fact that it is unclear how to correctly formulate this difficult problem.

[3] The terms "predictor" and "kernel" are used in this section interchangeably.

results appear in Subsection 4.1.3, where we report the detection performance of kernels optimized w.r.t. four different cover sources and three steganographic algorithms. In Subsection 4.1.4, we present and interpret the results of kernel optimization w.r.t. a collection of existing predictors. The section is concluded in Section 4.1.5 with a summary of the achievements and plans for future research together with a discussion about the limitations and possible applications of the proposed predictor optimization method.

A linear predictor will be described by a kernel $\mathbf{K} = (K_{ij}) \in \mathbb{R}^{k_1 \times k_2}$, while the image of predicted pixel values is $\hat{\mathbf{X}} = \mathbf{K} * \mathbf{X}$, where the star denotes convolution. All kernels will be normalized to $\sum_{ij} K_{ij} = -1$ so that the noise residual $\mathbf{R} = \mathbf{X} - \hat{\mathbf{X}}$ is the result of a high-pass filter applied to $\mathbf{X}$.

4.1.2 Methodology

In this section, we outline the methodology for optimizing the predictor parameters and evaluating their performance that will be used in all experiments in this section.

Let us assume that we have available a total of N cover images from a given source and a corresponding set of N stego images. Prior to optimization, we randomly split all cover-stego image pairs into two disjoint sets, $\mathcal{O}$ and $\mathcal{E}$, with $|\mathcal{O}| = N^{\text{opt}}$ and $|\mathcal{E}| = N - N^{\text{opt}}$ pairs used solely for predictor optimization and evaluation, respectively. The performance of each optimized predictor will be evaluated by reporting the minimum detection error under equal priors (2.2.1) averaged over ten random splits of $\mathcal{E}$ into $|\mathcal{E}| - 1000$ images used for training and 1000 images for testing. Next, we describe in detail the predictor design, which proceeds on $\mathcal{O}$.

4.1.2.1 Kernel parametrization

Each prediction kernel is parametrized before optimization. For example,

$$\mathbf{K} = \begin{pmatrix} 0 & 0 & 0 & 0 & b & c & d \\ 0 & 0 & a & 0 & a & 0 & 0 \\ d & c & b & 0 & 0 & 0 & 0 \end{pmatrix}. \tag{4.1.1}$$

contains four parameters a, b, c, d but the optimization is carried over parameters different from a (the so-called free parameters). Since we always normalize the kernel so that the sum of all elements is -1, the parameter a can be computed from the rest. In (4.1.1), there are three free parameters, $\theta^{\text{free}} = (b, c, d)$, while a can be computed from the normalization, and 13 elements of $\mathbf{K}$ are set to zero and will not participate in the optimization. Note that this particular kernel is forced to be symmetrical about the center element.

4.1.2.2 Feature vector

Given a noise residual $\mathbf{R} = (R_{ij})$ obtained using a predictor, we will form the steganalysis features as the joint probability distributions of neighboring residual samples. Based on the arguments outlined in Ref. [43], we use four-dimensional co-occurrence matrices formed by groups of four horizontally and vertically adjacent residual samples after they were quantized and truncated to a finite range:

$$R_{ij} \leftarrow \text{round}\left(\text{trunc}_T\left(\frac{R_{ij}}{q}\right)\right), \tag{4.1.2}$$

where $\text{trunc}_T(x) = x$ for $x \in [-T, T]$ and $\text{trunc}_T(x) = T\text{sign}(x)$ otherwise, and $q > 0$ is a quantization step. The co-occurrence matrix, $\mathbf{C} = (C_{\mathbf{d}})$, $\mathbf{d} = (d_1, \ldots, d_4) \in \{-T, \ldots, T\}^4$ with $T = 2$, is the sum

$$\mathbf{C}_{\mathbf{d}} = \mathbf{C}^{(\text{h})}_{\mathbf{d}} + \mathbf{C}^{(\text{v})}_{\mathbf{d}}, \tag{4.1.3}$$

where

$$\mathbf{C}_{\mathbf{d}}^{(\mathrm{h})} = \{(i,j)|R_{ij} = d_1, R_{ij+1} = d_2, R_{ij+2} = d_3, R_{ij+3} = d_4\}, \tag{4.1.4}$$

and $\mathbf{C}_{\mathbf{d}}^{(\mathrm{v})}$ is the vertical co-occurrence matrix defined analogically. We note that the vertical co-occurrence matrix, however, is formed from a residual computed using the transposed kernel $\mathbf{K}^{\mathrm{T}}$.

As in Ref. [43], the dimensionality of the co-occurrence matrix (4.1.3) will be reduced by leveraging symmetries of natural images. This will make the features more compact and better populated and will also increase the performance-to-dimensionality ratio. The symmetrization uses the sign-symmetry[4] as well as the directional symmetry of images by applying the following two rules for all $\mathbf{d} = (d_1, \ldots, d_4)$:

$$\mathbf{C}_{\mathbf{d}} \leftarrow \mathbf{C}_{\mathbf{d}} + \mathbf{C}_{-\mathbf{d}}, \tag{4.1.5}$$

$$\mathbf{C}_{\mathbf{d}} \leftarrow \mathbf{C}_{\mathbf{d}} + \mathbf{C}_{\overleftarrow{\mathbf{d}}}, \tag{4.1.6}$$

where $\overleftarrow{\mathbf{d}} = (d_4, d_3, d_2, d_1)$. After eliminating duplicates from $\mathbf{C}$ (which had originally $(2T+1)^4 = 625$ elements), only 169 unique elements remain.

4.1.2.3 Objective function

For the optimization, we need an objective function that would measure the detection performance. Since the optimization may involve a large number of evaluations, it is important that the objective function be fast. It is equally important that it be sufficiently smooth to avoid being trapped in local minima. Our first choice was to use the `L2R_L2LOSS` criterion, which is the margin of a linear support vector machine as proposed in Ref. [34]. The authors reported that as few as 80 pairs of cover and stego images were enough to make the margin well-behaved for multi-parameter optimization. However, using the margin turned out problematic in our case because changing the predictor changes both the distribution of stego and cover features. Since the margin is a geometric quantity, one needs to normalize the distribution of cover features, which is rather difficult for a multivariate distribution.

Consequently, we decided to use as the objective function the total detection error under equal priors (2.2.1) averaged over ten splits of $\mathcal{O}$ into random and equally sized training and testing sets. To decrease the complexity of evaluating the objective function, we used the ensemble classifier [77] with automatic setting for the number of base learners and the subspace dimensionality. This way, the most time consuming part of evaluating the objective function was computing the feature vector and not the training, whose complexity was rather negligible.

4.1.2.4 Optimization method

The predictor will be optimized w.r.t. its free kernel parameters as well as the quantization step q. We will denote the set of all parameters as $\{\theta^{\mathrm{free}}, q\}$. For the optimization, we used the gradient-free Nelder–Mead (N-M) algorithm [85, Chapter 9.5] implemented by Borggaard [11]. One vertex of the initial simplex, $\mathbf{v}^{(0)} = \{\theta^{\mathrm{ini}}, 1.5\}$, was always computed as the kernel with its free parameters set to θ^{ini}, the predictor optimal in the least-square sense estimated from 50 randomly chosen cover images from the training part of $\mathcal{O}$. The remaining vertices of the initial simplex were obtained by stretching each parameter by δ, $\mathbf{v}^{(j)} = \{\ldots, \mathbf{v}_{j-1}^{(0)}, \mathbf{v}_j^{(0)}(1+\delta), \mathbf{v}_{j+1}^{(0)}, \ldots, \}$, $j = 1, \ldots, |\theta^{\mathrm{ini}}| + 1$. Thus, a larger initial simplex can be obtained by increasing δ. In our experiments, we set $\delta = 0.3$.

The iterations stop when the difference between the minimal and maximal value on the simplex is below a certain tolerance $\epsilon = 10^{-6}$ or when the total number of iterations reaches 300.

Since the complexity of evaluating the objective function is linear in N^{opt}, to speed up the optimization, N^{opt} should be as small as possible. There is, however, a trade-off between speed and

[4] Sign-symmetry means that taking a negative of an image does not change its statistical properties.

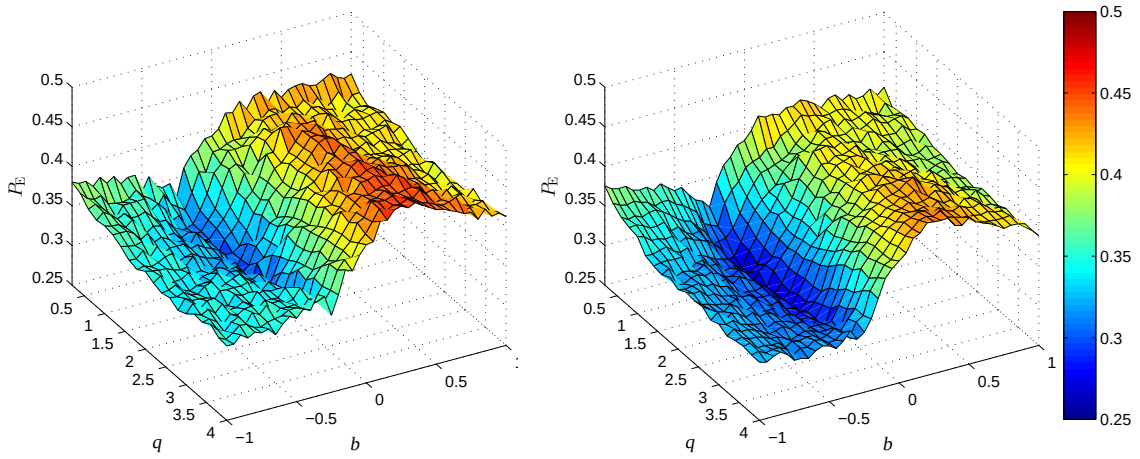

Figure 4.1.1: Detection error P_{E} as a function of one free kernel parameter (for kernel (4.1.7)) and the quantization step q for two sizes of the set $\mathcal{O}$: $N^{\mathrm{opt}} = 500$ left and $N^{\mathrm{opt}} = 2000$ right.

the smoothness of the objective function. Low values of N^{opt} would lead to a non-smooth objective function, which would increase the chances of getting trapped in local minima of the detection error P_{E}, requiring either a restart of the N-M algorithm or too many iterations to converge. According to our experience, it was in the end more efficient to use a higher value of $N^{\mathrm{opt}} = 2000$. Figure 4.1.1 shows the detection error (2.2.1) when optimizing a 3×3 rotationally symmetrical kernel with one free kernel parameter, $\theta^{\mathrm{free}} = \{b\}$:

$$\mathbf{K} = \begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix} \tag{4.1.7}$$

for $N^{\mathrm{opt}} = 500$ (left) and $N^{\mathrm{opt}} = 2000$ (right). In this particular case, the source was the BOSSbase database ver. 0.92 [36] with 9,074 cover images of size 512×512. It can be clearly seen that the surface of the right plot of the objective function is smoother.

4.1.3 Optimizing w.r.t. source and stego method

Our initial set of experiments aims at optimizing a simple predictor operating on the local 3×3 neighborhood with structure shown in (4.1.7). Our goal is to investigate how the optimal kernel parameter and the quantization step depend on the type of the cover source, the stego method, and even the steganography payload.

4.1.3.1 Cover sources

The experiments will be done on four different cover sources all containing grayscale 512×512 images. The first three are the BOSSbase ver. 0.92, NRCS512, and LEICA512, which contain raw, uncompressed images. The fourth database was obtained by JPEG compressing BOSSbase database with quality factor 80 using Matlab command 'imwrite'.[5] BOSSbase and Leica databases are described in Section 2.3.1, the NRCS512 database was derived from the NRCS database of 3,322 raw scans of negatives coming from the USDA Natural Resources Conservation Service. Two 512×512 images were obtained by cropping the central 512×1024 part of each NRCS image, splitting it in two, and converting each image to grayscale. Thus, the NRCS512 image set contained

[5] The images were always decompressed to the spatial domain before embedding.

Alg.	Pld.	Kernel	BOSSbase (a,b), q	BOSSbase P_E	NRCS (a,b), q	NRCS P_E	Leica (a,b), q	Leica P_E
HUGO	0.1	KB	(0.50, -0.25), 1.00	0.4390	(0.50, -0.25), 2.00	0.4862	(0.50, -0.25), 1.75	0.3813
		LSE	(0.45, -0.20), 2.00	0.4431	(0.51, -0.26), 1.75	0.4890	(0.48, -0.23), 1.50	0.3843
		Opt	(0.49, -0.24), 2.00	0.4378	(0.60, -0.35), 1.69	0.4886	(0.57, -0.32), 1.52	0.3654
	0.4	KB	(0.50, -0.25), 1.00	0.2637	(0.50, -0.25), 1.00	0.4395	(0.50, -0.25), 1.75	0.1358
		LSE	(0.45, -0.20), 1.50	0.2765	(0.51, -0.26), 2.00	0.4391	(0.48, -0.23), 1.50	0.1335
		Opt	(0.51, -0.26), 1.58	0.2649	(0.37, -0.12), 2.37	0.4350	(0.38, -0.13), 1.98	0.1207
EA	0.1	KB	(0.50, -0.25), 2.00	0.3785	(0.50, -0.25), 2.00	0.4766	(0.50, -0.25), 2.00	0.2477
		LSE	(0.45, -0.20), 2.00	0.3564	(0.51, -0.26), 1.75	0.4766	(0.48, -0.23), 2.00	0.2394
		Opt	(0.46, -0.21), 1.91	0.3542	(0.67, -0.42), 1.84	0.4736	(0.37, -0.12), 2.34	0.1796
	0.4	KB	(0.50, -0.25), 1.75	0.1793	(0.50, -0.25), 1.00	0.3956	(0.50, -0.25), 1.75	0.0462
		LSE	(0.45, -0.20), 1.75	0.1600	(0.51, -0.26), 1.50	0.3948	(0.48, -0.23), 2.00	0.0430
		Opt	(0.26, -0.01), 1.92	0.1374	(0.39, -0.14), 1.58	0.3706	(0.40, -0.15), 2.09	0.0352
LSBM	0.1	KB	(0.50, -0.25), 1.00	0.3105	(0.50, -0.25), 1.00	0.4782	(0.50, -0.25), 1.00	0.3689
		LSE	(0.45, -0.20), 1.00	0.3256	(0.51, -0.26), 1.50	0.4854	(0.48, -0.23), 1.50	0.3819
		Opt	(0.55, -0.30), 0.58	0.3142	(0.67, -0.42), 0.72	0.4741	(0.56, -0.31), 0.93	0.3711
	0.4	KB	(0.50, -0.25), 1.00	0.1250	(0.50, -0.25), 1.00	0.4052	(0.50, -0.25), 1.00	0.1049
		LSE	(0.45, -0.20), 1.00	0.1366	(0.51, -0.26), 1.00	0.4199	(0.48, -0.23), 1.50	0.1109
		Opt	(0.52, -0.27), 1.03	0.1248	(0.73, -0.48), 0.55	0.3970	(0.32, -0.07), 1.27	0.0828

Table 4.1: Optimized kernel parameters, a, b, and the quantization step q, together with the average testing error P_{E} for three stego methods, two payloads, and three databases of raw images.

a total of $2 \times 3322 = 6644$ images. All conversion to grayscale and resizing was carried out using the script 'convert' available from the BOSS web site [36]. The JPEG compression was done in Matlab R2011b using the command 'imwrite'.

4.1.3.2 Experiments on raw images

The results of experiments on raw images are shown in Table 4.1. The optimization algorithm was run as described in Section 4.1.3 with $N^{\mathrm{opt}} = 2000$ cover and stego images for optimizing the predictor. The remaining images from each image source were all used for testing. To investigate the effect of the message length, we repeated the experiments for two payload sizes – 0.1 and 0.4 bpp. The tables show the values of the kernel parameters a and b in (4.1.7) as well as the quantization step q. The rows with 'KB' show the average testing error (2.2.1) on $\mathcal{E}$ with the Ker–Böhme kernel,

$$\mathrm{KB} = \begin{pmatrix} -0.25 & 0.5 & -0.25 \\ 0.5 & 0 & 0.5 \\ -0.25 & 0.5 & -0.25 \end{pmatrix}, \tag{4.1.8}$$

derived in Ref. [8] when optimized over q, 'LSE' is the least-square kernel fit to covers again optimized over q, and 'Opt' denotes optimization over both the kernel and q. Shaded cells highlight interesting cases.

Note that the optimized kernel for BOSSbase is almost always rather close to the LSE kernel (the kernel that minimized the square prediction error on covers), which also coincides with the KB kernel (4.1.8). The improvement for HUGO and ± 1 embedding is thus solely due to optimizing over q rather than the kernel. The biggest improvement is observed for the EA algorithm for which the optimal kernel parameter is very different from the KB or LSE kernels – the corner parameter is almost zero, making the predictor support constrained to the four-pixel "cross" surrounding the central pixel. This can be explained by inspecting the embedding mechanism that hides message bits only in horizontal/vertical pixel pairs whose absolute difference is above a threshold determined beforehand by the payload size and the statistics of differences of each cover image. Another interesting observation is that the optimal quantization step for both adaptive methods is high, while

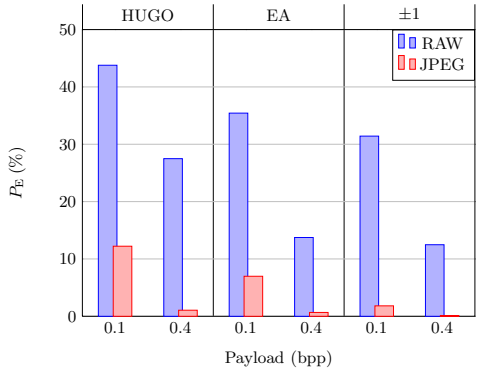

Figure 4.1.2: Average testing detection error P_E for RAW images and decompressed JPEGs (QF 80) from BOSSbase for three algorithms and two payloads.

it is small for the non-adaptive PM1. This is understandable since the adaptive methods embed in those regions of the image where the residual is large. Quantizing with a larger q moves some of the residual samples from its marginal back to the interior of the co-occurrence. In contrast, since the changes for PM1 embedding are not concentrated in edges or textures, there is no need to quantize strongly. In fact, the optimal q for small payload was $q = 0.58$. Overall, the improvement in detection error over using the KB and LSE predictors can be as large as $0.03 - 0.06$ in some cases.

For the NRCS database, the optimal predictors vary wildly across the two payloads (including the quantization step). Since the detection in this source is overall very unreliable due to the extreme noisiness of the scans, the optimized kernels most likely have no particular significance as they are affected too much by the employed machine learning and the noisiness of the objective function.

As expected, the Leica source is the easiest for steganalysis due to strong correlations among neighboring pixels. In shaded cells, the optimized kernel is very different from the KB and LSE kernels and the improvement can be very significant (e.g., for EA at 0.1 bpp where the detection error improved by almost 0.08). It is rather interesting to contrast the optimal kernel with Eq. (6.17) from Ref. [8] stating that the parameters of the LSE kernel, which is there recommended for weighted-stego steganalysis, should satisfy $-a/b = 1/(2\rho)$, where ρ is the correlation among neighboring pixels. Since this correlation is much stronger in Leica than in BOSSbase or NRCS, one would expect the ratio $-a/b$ to decrease and not increase. We interpret this as yet another example that optimizing the predictor for a binary detection problem is different than for source modeling. It is entirely possible, though, that our conclusions are due to the entire detection framework we use and if the residual was utilized differently, we may have ended up with a different optimal kernel.

Finally, as expected, HUGO is the least detectable algorithm out of the three, while EA is surprisingly less secure than the simple PM1 embedding in NRCS and Leica for both the small as well as the large payload. The detection accuracy strongly depends on the cover source, which is to be expected.

4.1.3.3 Experiments on JPEG decompressed images

The experiment from the previous section was carried out in exactly the same manner on BOSSbase ver. 0.92 JPEG compressed with quality factor 80 using the 'imwrite' command in Matlab. The results, which are displayed in Figure 4.1.2, show that feature-based classifiers can detect steganography in decompressed JPEGs significantly more reliably than in raw, uncompressed images. For example, for PM1 embedding in BOSSbase at payload 0.1, which translates to change rate $0.0112 \approx 1/90$ with an optimal ternary coder, using optimized kernel, we obtain the detection error of 0.182, down from 0.3142 for the corresponding source of raw images. We see two reasons

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} a & 0 & a \end{pmatrix}$	$0.29 \to 0.29$	$\begin{pmatrix} 0.5 & 0 & 0.5 \end{pmatrix}, 1.95$	0.2876	0.2876	169
$\begin{pmatrix} a & 0 & b \end{pmatrix}$	$0.28 \to 0.26$	$\begin{pmatrix} 0.048 & 0 & -0.952 \end{pmatrix}, 0.93$	0.3004	0.2509	338

Table 4.2: Complementing the second-order linear predictor by allowing an asymmetric kernel. Kernel orientation and co-occurrence scan are parallel.

for this shockingly better accuracy. First, decompressed JPEGs are smoother due to the low-pass effect of lossy compression. Second, realize that the original BOSSbase is a mixture of seven different sources, which increases the spread of the features and complicates the decision boundary. JPEG compression, on the other hand, homogenizes the cover source and decreases the spread of cover features, enabling thus much more reliable detection. Because the accuracy of detection of PM1 embedding is expected to be basically the same as for the LSB embedding (since our features are "parity-unaware" [44]), our feature-based detector likely outperforms in this particular source the best structural LSB detectors published in the literature that do not use JPEG compatibility, which, according to the best knowledge of the authors, never happened before. The results can be further vastly improved by using more complex feature sets instead of the simple 169-dimensional symmetrized co-occurrence from one type of residual.

Besides the exciting finding above, however, the predictor optimization for this source is not significant, which is why we do not report any detailed results here. The performance improvement of optimized kernels is mostly due to the quantization step instead of the kernel. This is possibly because JPEG compression makes the cover sources more homogeneous. Interestingly, however, for decompressed JPEG covers the optimal kernel was close to the KB kernel even for the EA algorithm. A quick inspection of the influence of individual co-occurrence bins revealed that JPEG compression almost empties particular co-occurrence bins which are later filled up again by embedding. The optimization seems to leverage this artifact of covers instead of the peculiarities of the embedding operation.

4.1.4 Conditional optimization

The accuracy of feature-based steganalysis can be significantly improved by forming the features from multiple different predictors [50, 45, 46, 73, 77, 43] as each predictor captures a different type of relationship among neighboring residual samples. This approach is recognized as steganalysis using rich models [43]. Merging features whose performance is correlated, albeit strong, is, however, not as effective as when one combines diverse features that are not necessarily as strong when used individually. Thus, having optimized a certain kernel structure, it makes sense to optimize the next predictor with respect to the existing ones. In fact, one can imagine building the entire rich model in this manner.

We next investigate the possibility of "cascading" the predictor design by optimizing the next predictor w.r.t. an existing set of predictors. We start with simple small-scale experiments that already reveal quite interesting facts and then scale up the approach. All experiments in this section are performed on the BOSSbase cover source and (unless mentioned otherwise) HUGO at payload 0.4 bpp as the stego source.

4.1.4.1 Complementing the 2nd-order predictor

It has been pointed out in Refs. [50, 46, 45] that second-order residuals obtained using the kernel $\mathbf{K} = \begin{pmatrix} 0.5 & 0 & 0.5 \end{pmatrix}$ are highly effective against HUGO.[6] This predictor essentially assumes that

[6] This is because HUGO preserves the joint pdf of pixel differences but not second-order differences among pixels.

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} a \\ 0 \\ a \end{pmatrix}$	$0.26 \to 0.26$	$\begin{pmatrix} 0.5 \\ 0 \\ 0.5 \end{pmatrix}, 1.95$	0.2545	0.2545	169
$\begin{pmatrix} a \\ 0 \\ b \end{pmatrix}$	$0.25 \to 0.24$	$\begin{pmatrix} 0.205 \\ 0 \\ 0.795 \end{pmatrix}, 1.06$	0.3176	0.2368	338

Table 4.3: Optimizing the second-order linear predictor by allowing an asymmetrical kernel. Kernel orientation is perpendicular to the co-occurrence scan.

the image is locally linear in the horizontal direction.

Running our optimization w.r.t. the quantization step only, we determined $q = 1.95$ as the one minimizing the detection error. The next question we asked was which 1×3 kernel with structure $(\ a \quad 0 \quad b\ )$ optimally supplements the second-order predictor. The optimization discovered that the best option is to use the first-order differences with quantization step $q = 0.93$, which is essentially the residual used in the SPAM feature vector!

The optimization results are displayed in Table 4.2. We use similar tables to report the results of other experiments in this section. The first column shows the structure of the kernels for optimization (recall that we do not optimize over a). The second column shows the value of the objective function (see its definition in Section 4.1.2.3) at the initial point $\mathbf{v}^{(0)}$ of the simplex, $P_{\mathrm{E}}^{\mathrm{ini}}$, and the final value, $P_{\mathrm{E}}^{\mathrm{opt}}$, after the optimization ends. The third column contains the final optimized predictor. The fourth and fifth columns hold the average detection error on $\mathcal{E}$ when only the optimized kernel is used individually, $P_{\mathrm{E}}^{\mathrm{indiv}}$, and after merging the kernels from all rows above, $P_{\mathrm{E}}^{\mathrm{merged}}$. Finally, the last column shows the dimensionality after merging the features.

Notice that the individual performance of the second predictor is not very high and there certainly exist other kernels and quantization steps that would give higher individual performance. However, when considered *jointly* with the first predictor, adding these 169 features decreases the error from almost 0.29 to about 0.25.

We repeated the same experiment with kernels oriented perpendicularly to the scan of the co-occurrence. Several interesting phenomena are apparent in Table 4.3 that shows the results. First, forming the co-occurrence in a perpendicular direction to the kernel orientation leads to better detection of HUGO. Our intuitive understanding of this, confirmed by many experiments [43], is that the larger is the support of the kernel combined with the co-occurrence matrix, the better. For example, the horizontal kernel $(\ 0.5 \quad 0 \quad 0.5\ )$ combined with 4th-order horizontal co-occurrence has a support of 6 pixels, while the vertical kernel $(\ 0.5 \quad 0 \quad 0.5\ )^T$ combined with the same co-occurrence matrix has a support of 12 pixels. Second, the best kernel is no longer the first-order difference as before. Third, there is an even bigger contrast between the rather poor individual performance of the second predictor and the improvement it provides when merged with the features from the first predictor.

4.1.4.2 Cascading the 3×3 kernel (guided)

In the next experiment, we decided to cascade predictors defined on the local 3×3 neighborhood (Table 4.4). As in the previous section, the design is "guided" by restricting the structure of the kernel at each step. The first kernel identified by the optimization is very close to the KB kernel, which also gives the smallest square prediction error on BOSSbase, and the quantization step is $q \approx 1$. In the second and third steps, we allowed an asymmetric structure for the "central cross."

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix}$	$0.28 \to 0.27$	$\begin{pmatrix} -0.259 & 0.509 & -0.259 \\ 0.509 & -1 & 0.509 \\ -0.259 & 0.509 & -0.259 \end{pmatrix}$, 1.58	0.2649	0.2649	169
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$0.26 \to 0.22$	$\begin{pmatrix} -0.034 & 0.503 & -0.034 \\ 0.064 & -1 & 0.064 \\ -0.034 & 0.503 & -0.034 \end{pmatrix}$, 2.23	0.2722	0.2177	338
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$0.22 \to 0.20$	$\begin{pmatrix} -0.044 & -0.092 & -0.044 \\ 0.682 & -1 & 0.682 \\ -0.044 & -0.09 & -0.044 \end{pmatrix}$, 2.03	0.3221	0.2025	507

Table 4.4: Cascading predictors on the local 3×3 neighborhood by guiding the process with predefined kernel symmetries for HUGO.

The optimization found essentially a one-dimensional vertical linear predictor and its corresponding horizontal counterpart. For both predictors, the optimal quantization step was $q > 2$, indicating that the predictors are forced to "see" embedding changes in textures and around edges. Note that while the third predictor has a rather weak individual performance ($P_{\mathrm{E}}^{\mathrm{indiv}} = 0.3221$), it complements the previous two predictors rather well, lowering the testing error by one and a half percent. However, it is obvious that lowering the error further by cascading kernels of the same type becomes increasingly harder. We hypothesize that cascading kernels with the same support is not the most efficient way of improving performance per dimensionality as such kernels cannot be by definition too diverse. Having said this, it is certainly interesting that this iterative design gave a 507-dimensional feature vector with detection error ~ 0.20, which is close to the performance of the winning team in the BOSS competition [46].[7]

To show the influence of the embedding algorithm on the resulting optimized kernels, we repeated this experiment using EA instead of HUGO (Table 4.5). It is mentioned in Section 4.1.3.2 that the optimized kernel adapts to a weakness of EA by nullifying the corner coefficients. By inspecting the corner coefficients of the second and the third cascaded kernels, it seems that this weakness is completely utilized by the first kernel. Note that the value of the objective function increases from $P_{\mathrm{E}}^{\mathrm{opt}} = 0.13$ in the second row to $P_{\mathrm{E}}^{\mathrm{ini}} = 0.14$ in the third row. This is caused by random selection of subspaces and bootstraps in the ensemble classifier [77] together with a relatively small size of the set $\mathcal{O}$, and the final rounding to integers.

4.1.4.3 Cascading the vertical 5×1 kernel (unguided)

In the last experiment of this section, we investigate an unguided design for a 5×1 kernel that is perpendicular to the co-occurrence scan. By unguided, we mean that the kernel structure at each step was fixed to $(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$ and thus the optimization was carried over four parameters – three free kernel parameters and the quantization step. The results of the first four steps are shown in Table 4.6, where for compactness we display the optimal kernels graphically. The kernels seem to form a "basis" of sorts as they try to complement each other. By merging the cascaded features, the detection error is gradually decreasing but eventually exhibits signs of saturation when this process continues (not shown in the table). This is most likely because cascading the same predictor structure does not allow for enough diversity to further lower the error.

Using the conditional optimization for the entire design of a rich model, however, is somewhat problematic when approached the way described in this section. We observed that when the optimization

[7] The BOSS score is not directly comparable to our experiment due to cover mismatch that plagued the detection results of participating teams.[50, 45]

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$\begin{pmatrix} b & a & b \\ a & 0 & a \\ b & a & b \end{pmatrix}$	$0.18 \to 0.15$	$\begin{pmatrix} -0.015 & 0.265 & -0.015 \\ 0.265 & -1 & 0.265 \\ -0.015 & 0.265 & -0.015 \end{pmatrix}$, 1.92	0.1406	0.1406	169
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$0.14 \to 0.13$	$\begin{pmatrix} -0.267 & 0.428 & -0.267 \\ 0.606 & -1 & 0.606 \\ -0.267 & 0.428 & -0.267 \end{pmatrix}$, 1.50	0.1782	0.1258	338
$\begin{pmatrix} c & b & c \\ a & 0 & a \\ c & b & c \end{pmatrix}$	$14 \to 13$	$\begin{pmatrix} -0.189 & 0.510 & -0.189 \\ 0.368 & -1 & 0.368 \\ -0.189 & 0.510 & -0.189 \end{pmatrix}$, 1.81	14.93	11.91	507

Table 4.5: Cascading predictors on the local 3×3 neighborhood by guiding the process with predefined kernel symmetries for EA.

Optimization on $\mathcal{O}$			Evaluation on $\mathcal{E}$		
Structure	$P_{\mathrm{E}}^{\mathrm{ini}} \to P_{\mathrm{E}}^{\mathrm{opt}}$	Optimized predictor	$P_{\mathrm{E}}^{\mathrm{indiv}}$	$P_{\mathrm{E}}^{\mathrm{merged}}$	Dim
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$0.25 \to 0.23$	 1.71	0.2436	0.2436	169
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$0.23 \to 0.22$	 1.69	0.2467	0.2312	338
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$0.23 \to 0.21$	 2.48	0.2671	0.2182	507
$(\ a \quad b \quad 0 \quad c \quad d\ )^{\mathrm{T}}$	$0.22 \to 0.21$	 2.31	0.3574	0.2013	676

Table 4.6: Cascading predictors on the local 5×1 neighborhood.

is run over five or more parameters, the optimal parameter vector becomes frequently trapped in local minima and does not find a better solution (even after restarting from a different initial condition) even when better solutions are known to exist. Moreover, for higher dimensionality of the parameter vector the objective function seems to contain numerous shallow regions where the search randomly wanders around without converging. This is undoubtedly tied to the particular form of the objective function. Our attempts to start with a larger kernel structure, such as a general unconstrained 5×5 kernel, and iterating the optimization did not provide meaningful or particularly good results.

This problem forced us to restrict the structure of the next predictor, in which case the optimization seems to produce interesting interpretable results. However, this approach towards building the rich model would mean heavy involvement of the user, which is undesirable.

4.1.5 Summary

Pixel predictors are commonly employed when constructing steganalysis features from noise residuals as co-occurrences of adjacent residual samples. The predictor plays an important role – it is known that combining features computed from residuals obtained using a diverse set of predictors markedly improves detection performance. In this section, we introduce a method for optimizing the predictor parameters to improve detection performance for a fixed source and stego method within a specific detection framework. The predictor parameters are kernel elements of a linear filter and a quantization step using which the residual is quantized.

On four different cover sources, three spatial-domain steganographic methods, and two payloads, we show the effectiveness of the proposed approach. Among other findings, we observed that the optimal predictor may strongly depend on the embedding mechanism as well as the cover source. The improvement in detection error ranges from rather small to quite substantial, depending on the source and stego method.

The proposed framework is also applicable to the case when the predictor is optimized w.r.t. a set of existing predictors, which allows "cascading" the predictors to maximize the performance–dimensionality ratio.

According to our experience, the method as proposed is limited to optimizing over a rather small parameter vector (e.g., up to dimension of five), which is most likely due to the character of the objective function. Search for better behaved objective functions that may remove this limitation is considered as part of the future effort.

The predictor optimization may also be of lesser importance when applied to a rich model consisting of feature sets from potentially hundreds of predictors as the individual feature sets may compensate as a whole for deficiencies of others. However, when the goal is to select a small subset of features with an overall good performance, the optimized predictors are expected to play an important role.

One interesting finding not related to the topic of this section, which is predictor optimization, is that feature-based steganalysis can be very effective for sources consisting of decompressed JPEG images. It is indeed possible to outperform structural detectors in such sources by a rather large margin by making the features parity aware [44].

4.2 Spatial Rich Model residuals

The previous section explained what a predictor is and its role in steganalysis. It mostly focused on developing a small number of optimized linear predictors for a given steganographic algorithm and cover source and, on some examples, achieved a rather significant improvement in detection. Unfortunately, as mentioned in the summary, this optimization approach is limited by the number of free predictor parameters, its size, and the number of predictors that can be cascaded.

This section describes a set of 45 hand-designed diverse predictors consisting of both linear and non-linear filters. The predictor set was proposed for the high dimensional Spatial Rich Model (SRM) [43] feature set as the final product of steganalyzing HUGO [90] during the BOSS competition [4]. Since it is not in the scope of this dissertation, we avoid an exhaustive description of the SRM itself – we focus merely on its predictors, because they are used in the PSRM feature set in Section 4.3. The residual quantization, truncation, and capturing dependencies using 4-dimensional co-occurrences is identical to those previously defined in Subsection 4.1.2.2, with the expection of symmetrization of residuals created by non-linear predictors. This secion follows the structure and notation laid down in Ref. [43].

4.2.1 Common approach

The overall goal is to capture multiple different types of dependencies among neighboring pixels to give the model the ability to detect embedding changes in diverse content (edges, smooth areas) made by a wide spectrum of embedding algorithms. This can not be achieved by enlarging a single model as the enlarged model would have too many underpopulated bins (e.g., think of the second order SPAM model with a large truncation threshold T employed by HUGO [90]). Instead, the model is formed by merging many small submodels, consequently avoiding the problem with underpopulated bins.

The submodels are formed from noise residuals, $\mathbf{R} = (R_{ij}) \in \mathbb{R}^{n_1 \times n_2}$, computed using high-pass filters of the following form:

$$R_{ij} = \hat{X}_{ij}\left(\mathcal{N}_{ij}\right) - cX_{ij}, \tag{4.2.1}$$

where $c \in \mathbb{N}$ is the residual order, $\mathcal{N}_{ij}$ is a local neighborhood of pixel X_{ij}, $X_{ij} \notin \mathcal{N}_{ij}$, and $\hat{X}_{ij}(.)$ is a predictor of cX_{ij} defined on $\mathcal{N}_{ij}$. The set $\{X_{ij} + \mathcal{N}_{ij}\}$ is called the support of the residual. The main advantage of residuals over pixel values is that the image content is partially suppressed in $\mathbf{R}$, which has a narrower dynamic range. This allows the steganalyst to use more compact and robust statistical descriptors. Examples of features for steganalysis formed this way are [31, 49, 57, 89].

Each residual is first quantized and truncated using Eq. (4.1.2). Truncation using $T > 0$ is applied in order to limit residual's dynamic range allowing thus their description using co-occurrence matrices with small dimensionality (co-occurrences are defined in Subsection 4.1.2.2 and Ref. [43]). The quantization makes the residual more sensitive to embedding changes at spatial discontinuities in the image (at edges and in textures) where content-adaptive steganographic algoritms make most of the embedding changes.

4.2.2 Individual submodels

All residuals used in the SRM are shown in Fig. 4.2.1. The residuals are built using locally-supported linear filters and their output is possibly combined using minimum and maximum operators in order to increase their diversity. It might be beneficial to think of each filter in terms of its predictor for better insight. For example, the first-order residual $R_{ij} = X_{i,j+1} - X_{ij}$ predicts its central pixel X_{ij} as its neighbor, $\hat{X}_{ij} = X_{i,j+1}$, while the second-order residual $R_{ij} = X_{i,j-1} + X_{i,j+1} - 2X_{ij}$ assumes that the image is locally linear in the horizontal direction, $2\hat{X}_{ij} = (X_{i,j-1} + X_{i,j+1})$. Higher-order differences and differences involving larger support make more complicated assumptions than a locally linear behavior.

Let us provide detailed explanation of Fig. 4.2.1. The central pixel X_{ij} at which the residual (4.2.1) is evaluated is always marked with a black dot and accompanied with an integer value c from (4.2.1). If the chart contains only one type of symbol besides the black dot, we say that the residual is of type 'spam' (*1a*, *2a*, *3a*, *S3a*, *E3a*, *S5a*, *E5a*) because of its similarity to the SPAM feature vector [89].

If there are two or more different symbols other than the black dot, we call its type 'minmax'. For 'spam' type, the residual is computed as a linear high-pass filter of neighboring pixels with the corresponding coefficients. For example, *2a* stands for the second-order $R_{ij} = X_{i,j-1} + X_{i,j+1} - 2X_{ij}$ and *1a* for the first-order $R_{ij} = X_{i,j+1} - X_{ij}$ residuals. The 'minmax' type residuals, on the other hand, use two or more linear filters, each filter corresponding to one symbol type. The final residual is obtained by taking the minimum (or maximum) of the filters' outputs. Therefore, there will be two minmax residuals – one for the operation of 'min' and one for 'max'. For example, *2b* is obtained as $R_{ij} = \min\{X_{i,j-1} + X_{i,j+1} - 2X_{ij}, X_{i-1,j} + X_{i+1,j} - 2X_{ij}\}$ and *1g* is $R_{ij} = \min\{X_{i-1,j-1} - X_{ij}, X_{i-1,j} - X_{ij}, X_{i-1,j+1} - X_{ij}, X_{i,j+1} - X_{ij}\}$, etc. The 'min' and 'max' operators introduce non-linearity into the residuals and desirably increase diversity of the model. These operations also make the distribution of the residual values non-symmetrical, thinning out one tail of the distribution and thickening the other.

The number of filters, f, is the first digit attached to the end of the residual name. The third-order residuals are computed just like the first-order residuals by replacing, e.g., $X_{i,j+1} - X_{ij}$ with $-X_{i,j+2} + 3X_{i,j+1} - 3X_{ij} + X_{i,j-1}$. The differences along other directions are obtained analogically.

Fig. 4.2.1 shows that the residuals are divided into six classes depending on the central pixel predictor they are built upon. The classes are denoted with the following names: 1st, 2nd, 3rd, SQUARE, EDGE3x3, and EDGE5x5. All predictors in class '1st' estimate the pixel as the value of its neighbor, while those from class '2nd' resp. '3rd' incorporate a locally linear and quadratic model respectively. The latter predictors are more accurate in image areas with a strong gradient/curvature, i.e., in regions with more complex image content.

The class 'SQUARE' uses more pixels for the prediction. The 3×3 square kernel S3a has been used in steganalysis before [65] – it also coincides with the best (in the least-square sense) shift-invariant linear pixel predictor on the 3×3 neighborhood for BOSSbase cover image database. The class

Figure 4.2.1: Definitions of all residuals. The residuals 3a – 3h are defined similar to the first-order residuals, while E5a – E5d are similar to E3a – E3d defined using the corresponding part of the 5 × 5 kernel displayed in S5a. This diagram is taken from Ref. [43].

'EDGE3x3' predictors, derived from this kernel, were included to provide better estimates at edges. The larger S5a 5×5 predictor was obtained by optimizing the coefficients of a circularly symmetrical 5×5 kernel using the approach from Section 4.1 and Ref. [55]. While this predictor was derived on a specific embedding algorithm, it performes very well against all standard steganographic algorithms. The 'EDGE5x5' residuals E5a–E5d (not shown in Fig. 4.2.1) are derived from S5a in an analogical manner as E3a–E3d are built from S3a.

4.2.3 From residuals to SRM features

The next step in forming the SRM feature vector involves computing a co-occurrence matrix of Dth order from D (horizontally and vertically) neighboring values of the quantized residual (**??**) from the entire image. As argued in the original publication, diagonally neighboring values are not included due to much weaker dependencies among residual samples in diagonal directions. To keep the co-occurrence bins well-populated and thus statistically significant, the authors of the SRM used small values for D and T: $D = 4$, $T = 2$, and $q \in \{1, 1.5, 2\}$. Finally, symmetries of natural images are leveraged to further marginalize the co-occurrence matrix to decrease the feature dimension and better populate the SRM feature vector (see Section II.C of [43]).

Note that non-linear residuals are represented using two co-occurrence matrices, one for 'min' operator and one for 'max' operator, while linear residuals require a single co-occurrence matrix. The authors of the SRM combined the co-occurrences of two linear residuals into one "submodel" to give them after symmetrization approximately the same dimensionality as the union of co-occurrences from min / max non-linear residuals. This allowed a fair comparison of detection performance of individual submodels. The authors also used a simple forward feature selection on submodels to improve the dimensionality vs. detection accuracy trade-off. There are a total of 39 submodels in the SRM.

We denote the full version of this model with all three quantization steps as SRM (its dimensionality is $34,671$). A scaled-down version of the SRM when only one quantization step $q = 1$ is used will be abbreviated as SRMQ1. Its dimensionality is $12,753$.

4.3 Random projections of residuals

This section is based on author's article [57] published in IEEE Transactions on Information Forensics and Security (TIFS) journal in 2013.

The traditional way to represent digital images for feature based steganalysis is to compute a noise residual from the image using a pixel predictor and then form the feature as a sample joint probability distribution of neighboring quantized residual samples – the so-called co-occurrence matrix. In this section, we propose an alternative statistical representation – instead of forming the co-occurrence matrix, we project neighboring residual samples onto a set of random vectors and take the first-order statistic (histogram) of the projections as the feature. When multiple residuals are used, this representation is called the projection spatial rich model (PSRM). On selected modern steganographic algorithms embedding in the spatial, JPEG, and side-informed JPEG domains, we demonstrate that the PSRM can achieve a more accurate detection as well as a substantially improved performance vs. dimensionality trade-off than state-of-the-art feature sets.

4.3.1 Introduction

Traditionally, noise residuals were represented using either sample joint or conditional probability distributions of adjacent quantized and truncated residual samples (co-occurrence matrices) [114, 89, 43, 50]. Higher-order co-occurrences detect steganographic changes better as they can capture

dependencies across multiple pixels. Since the co-occurrence dimensionality increases exponentially with its order, the co-occurrence order one can use in practice is limited by the total number of pixels, and steganalysts had to quantize and truncate the residual (sometimes quite harshly) to obtain a reasonably low-dimensional and statistically significant descriptor for subsequent machine learning [43, 50, 89].

In this section an alternative statistical descriptor for noise residuals is proposed. Instead of forming co-occurrences of neighboring quantized residual samples, we use the unquantized values and project them on random directions, which are subsequently quantized and represented using histograms as steganalytic features. This brings several advantages over the representation based on co-occurrences. First, by using large projection neighborhoods one can potentially capture dependencies among a large number of pixels. Second, by selecting random neighborhood sizes, the statistical description can be further diversified, which improves the detection accuracy. Third, since more features will be statistically significant in comparison to high-dimensional co-occurrences where numerous boundary bins may be underpopulated, projections enjoy a much more favorable feature dimensionality vs. detection accuracy trade-off. Fourth, a greater design flexibility is obtained since the size and shape of the projection neighborhoods, the number of projection vectors, as well as the histogram bins can be incrementally adjusted to achieve a desired trade-off between detection accuracy and feature dimensionality. Finally, the novel feature representation appears to be universally effective for detection of modern steganographic schemes embedding in both the spatial and JPEG domains.

The SRM [43] residuals are used to construct the PSRM (projection spatial rich model) proposed in Subsection 4.3.2. This subsection also contains several investigative experiments used to set the PSRM parameters. In Subection 4.3.3, we compare the detection performance of the proposed PSRM with the current state-of-the-art feature descriptors – the SRM and the JRM (JPEG rich model) proposed in [74]. The comparison is carried out on selected modern (and currently most secure) steganographic algorithms operating in the spatial, JPEG, and side-informed JPEG domains. This section is concluded in Subsection 4.3.4.

4.3.2 Projection spatial rich model

In this subsection, we provide the reasoning behind the proposed projection spatial rich model and describe it in detail, including the experiments used to set the PSRM parameters.

4.3.2.1 Motivation

The residual is a realization of a two-dimensional random field whose statistical properties are closely tied to the image content (e.g., larger values occur near edges and in textures while smaller values are typical for smooth regions). Steganographic embedding changes modify the statistical properties of this random field. The steganalyst's task is to compute a test statistic from this random field that would detect the embedding changes as reliably as possible.

Traditionally, and as described in the previous section, the random field is first quantized and then characterized using a joint probability mass function (co-occurrence matrix) of D neighboring residual samples. The problem with this approach is the exponential growth of the co-occurrence size with its order D. With increasing D, a rapidly increasing number of co-occurrence bins become underpopulated, which worsens the detection–dimensionality trade-off and makes subsequent machine learning more expensive and the detection less accurate. This is because adding features that are essentially random noise may decrease the ability of the machine learning tool to learn the correct decision boundary. Also, with a small value of the truncation threshold T, some potentially useful information contained in the residual tails is lost, which limits the detection accuracy of highly adaptive schemes. Finally, since the co-occurrence dimensionality is $(2T+1)^D$, changing the parameters T and D gives the steganalyst rather limited options to control the feature dimensionality.

There are several possible avenues one can adopt to resolve the above issues. It is possible, for example, to overcome the problem with underpopulated bins by replacing the uniform scalar quantizer applied to each residual with a vector quantizer designed in the D-dimensional space of residuals and optimize w.r.t. the quantizer centroids. However, as the reference [87] shows, this approach lead to a rather negligible improvement in detection. A largely unexplored direction worth investigating involves representing adjacent residual samples with a high-dimensional joint distribution and then applying various dimensionality reduction techniques.

The avenue taken in this section is to utilize dependencies among residual samples from a much larger neighborhood than what would be feasible to represent using a co-occurrence matrix. This way, we potentially use more information from the residual and thus improve the detection. Let us denote by $\mathcal{N}(\mathbf{Y}, i, j)$ an arbitrarily shaped neighborhood of pixel y_{ij} with $|\mathcal{N}|$ pixels. In the next subsection, we will consider rectangular $k \times l$ neighborhoods. Furthermore, we assume that the (unquantized) residual samples from $\mathcal{N}(\mathbf{Y}, i, j)$, $1 \leq i \leq n_1$, $1 \leq j \leq n_2$, are $|\mathcal{N}|$-dimensional vectors drawn from a probability distribution $\rho(\mathbf{x})$, $\mathbf{x} \in \mathbb{R}^{|\mathcal{N}|}$. Since for large $|\mathcal{N}|$, quantizing $\rho(\mathbf{x})$ and representing it using a co-occurrence matrix would not make a good test statistic due to heavily underpopulated bins, we instead project the residual on random vectors $\mathbf{v} \in \mathbb{R}^{|\mathcal{N}|}$, $\mathbf{v} \neq 0$, and choose the first-order statistic of the projections as steganalysis features.

While it is certainly possible to use higher-order statistics for a fixed projection vector and neighborhood, in general, however, it is better to diversify the features by adding more projection neighborhoods and vectors rather than a more detailed description for one projection and neighborhood. See [45, 46, 50] for more details.

Intuitively, when selecting sufficiently many projection vectors $\mathbf{v}$, we improve our ability to distinguish between the distributions of cover and stego images. Furthermore, the random nature of vectors $\mathbf{v}$ is an important design element as it makes the steganalyzer key-dependent, making it harder for an adversary to design a steganographic scheme that evades detection by a specific steganalysis detector. The projection vectors could be optimized for a given cover source and stego method to obtain the best trade-off between feature dimensionality and detection accuracy. However, our goal is to present a universal feature vector capable of detecting potentially all stego schemes in arbitrary cover sources.

4.3.2.2 Residual projection features

In this section, we formally describe the process used to build the projection spatial rich model. We begin by introducing several key concepts. A specific instance of a projection neighborhood is obtained by first selecting two integers, $k, l \leq s$ randomly uniformly, where s is a fixed positive integer. The projection neighborhood is a matrix $\mathbf{\Pi} \in \mathbb{R}^{k \times l}$ whose elements, π_{ij}, are $k \cdot l$ independent realizations of a standard normal random variable $N(0, 1)$ normalized to a unit Frobenius norm $\|\mathbf{\Pi}\|_2 = 1$.[8] This way, the vector $\mathbf{v}$ obtained by arranging the elements of $\mathbf{\Pi}$, e.g., by rows, is selected randomly and uniformly from the surface of a unit sphere. This choice maximizes the spread of the projection directions.

To generate another instance of a projection neighborhood, we repeat the process with a different seed for the random selection of k, l as well as the elements of $\mathbf{\Pi}$. For a given instance of the projection neighborhood $\mathbf{\Pi}$ and residual $\mathbf{Z}$, the projection values $\mathbf{P}(\mathbf{\Pi}, \mathbf{Z})$ are obtained by convolving $\mathbf{Z}$ with the projection neighborhood $\mathbf{\Pi}$:

$$\mathbf{P}(\mathbf{\Pi}, \mathbf{Z}) = \mathbf{Z} * \mathbf{\Pi}. \tag{4.3.1}$$

Similarly to the features of the SRM, we utilize symmetries of natural images to endow the statistical descriptor with more robustness. In particular, we use the fact that statistical properties of natural images do not change with direction or mirroring. For non-directional residuals, such as the one obtained using the kernel S3a in 4.2.1, we can enlarge the set $\mathbf{P}$ (4.3.1) by adding to it projections

[8] The Frobenius norm of a matrix $\mathbf{A} \in \mathbb{R}^{k \times l}$ is defined as $\|\mathbf{A}\|^2 = \sum_{i=1}^{k} \sum_{j=1}^{l} a_{ij}^2$.

with the matrix $\mathbf{\Pi}$ obtained by applying to it one or more following geometrical transformations: horizontal mirroring, vertical mirroring, rotation by 180 degrees, and transpose, respectively:

$$\overleftrightarrow{\mathbf{\Pi}} = \begin{pmatrix} \pi_{12} & \pi_{11} \\ \pi_{22} & \pi_{21} \end{pmatrix}, \tag{4.3.2}$$

$$\mathbf{\Pi}\updownarrow = \begin{pmatrix} \pi_{21} & \pi_{22} \\ \pi_{11} & \pi_{12} \end{pmatrix}, \tag{4.3.3}$$

$$\mathbf{\Pi}^{\circlearrowleft} = \begin{pmatrix} \pi_{22} & \pi_{21} \\ \pi_{12} & \pi_{11} \end{pmatrix}, \tag{4.3.4}$$

$$\mathbf{\Pi}^{T} = \begin{pmatrix} \pi_{11} & \pi_{21} \\ \pi_{12} & \pi_{22} \end{pmatrix}. \tag{4.3.5}$$

By combining these four transformations, one can obtain a total of eight different projection kernels.

The situation is a little more involved with directional residuals. The directional symmetry of natural images implies that we can merge the projections of a horizontal residual with projection kernels $\mathbf{\Pi}$, $\overleftrightarrow{\mathbf{\Pi}}$, $\mathbf{\Pi}\updownarrow$, and $\mathbf{\Pi}^{\circlearrowleft}$, and the projections obtained using their transposed versions applied to the vertical residual because its kernel is a transpose of the horizontal kernel.

Since a linear predictor (4.2.1) is a high-pass filter, the residual distribution for natural images will be zero mean and symmetrical about the y axis. Consequently, the distribution of the residual projections will also be symmetrical with a maximum at zero. Since we will be taking the first-order statistic (histogram) of the projections as the feature vector, the distribution symmetry allows us to work with absolute values of the projections and use either a finer histogram binning or a higher truncation threshold T. Denoting the bin width q, we will work with the following quantizer with $T + 1$ centroids:

$$\mathcal{Q}_{T,q} = \{q/2, 3q/2, \ldots, (2T + 1)q/2\}. \tag{4.3.6}$$

We would like to point out that by working with absolute values of the projections, our features will be unable to detect a steganographic scheme that preserves the distribution of the absolute values of projections yet one which violates the histogram symmetry. However, this is really only a minor issue as the projections are key-dependent and it would likely be infeasible to build an embedding scheme with this property for every projection vector and neighborhood. Moreover, an embedding scheme creating such an asymmetry would be fundamentally flawed as one could utilize this symmetry violation to construct a very accurate targeted quantitative attack. A good example is the Jsteg algorithm [105].

We now provide a formal description of the features. For a fixed set of quantizer centroids, $\mathcal{Q}_{T,q}$, the histogram of projections $\mathbf{P}$ is obtained using the following formula:

$$\mathbf{h}(l; \mathcal{Q}_{T,q}, \mathbf{P}) = \sum_{p \in \mathbf{P}} [Q_{\mathcal{Q}_{T,q}}(|p|) = l], \quad l \in \mathcal{Q}_{T,q}, \tag{4.3.7}$$

where [.] stands for the Iverson bracket defined as $[S] = 1$ when the statement S is true and 0 otherwise.

Considering the outputs of the residuals involved in computing a min (max) residual as independent random variables Z_1, Z_2, ..., Z_r, $E[\min\{Z_1, Z_2, ..., Z_r\}] < 0$ and $E[\max\{Z_1, Z_2, ..., Z_r\}] > 0$. Thus, the distribution of residuals obtained using the operations min (max) is not centered at zero and one can no longer work with absolute values of residuals. Instead, we use the following expanded set of centroids:

$$\mathcal{Q}^{(\mathrm{x})}_{T,q} = \mathcal{Q}_{T,q} \cup \{-\mathcal{Q}_{T,q}\}, \tag{4.3.8}$$

which has double the cardinality of $\mathcal{Q}_{T,q}$. Because for any finite set $\mathcal{R} \subset \mathbb{R}$, $\min \mathcal{R} = -\max\{-\mathcal{R}\}$, the distribution of the projections $\mathbf{P}^{(\min)}$ of residuals $\mathbf{Z}^{(\min)}$ is a mirror image about the y axis of the distribution of $\mathbf{P}^{(\max)}$ of $\mathbf{Z}^{(\max)}$. One can use this symmetry to improve the robustness of the

features and decrease their dimensionality by merging the projections $\mathbf{P}^{(\min)}$ and mirrored $\mathbf{P}^{(\max)}$ into one histogram:

$$\begin{aligned}&\mathbf{h}(l; \mathcal{Q}^{(\mathrm{x})}_{T,q}, \mathbf{P}^{(\min)}, \mathbf{P}^{(\max)}) = \textstyle\sum_{p\in\mathbf{P}^{(\min)}} [Q_{\mathcal{Q}^{(\mathrm{x})}_{T,q}}(p) = l] \\ &+ \textstyle\sum_{p\in\mathbf{P}^{(\max)}} [Q_{\mathcal{Q}^{(\mathrm{x})}_{T,q}}(-p) = -l], l \in \mathcal{Q}^{(\mathrm{x})}_{T,q}.\end{aligned} \tag{4.3.9}$$

We note that the min a max residuals from the same submodel share the same projection neighborhood $\mathbf{\Pi}$.

To reduce the feature dimensionality, we do not include in the feature vector the last (marginal) bin $\mathbf{h}(l)$ corresponding to $l = (2T+1)q/2$ because its value can be computed from the remaining bins and is thus redundant for training the machine-learning-based classifier. Thus, for each linear residual $\mathbf{Z}$, the set of projections, $\mathbf{P}(\mathbf{Z}, \mathbf{\Pi})$, is represented in the PSRM using a T-dimensional vector $\mathbf{h}(l)$, $l \in \mathcal{Q}_{T,q} - \{(2T+1)q/2\}$. Similarly, and for the same reason, for a non-linear residual, we exclude the bins corresponding to $l = \pm(2T+1)q/2$, which gives us $2T$ features. Since in the SRM the features from two linear residuals are always paired up into one submodel (see Section II.C of [43]), we do the same in the proposed PSRM, which means that the projections of residuals from a given submodel are represented using exactly $2T$ features.

In summary, for a given submodel (a pair of residuals) and a projection neighborhood $\mathbf{\Pi}$ we obtain $2T$ values towards the PSRM. Since there are a total of 39 submodels in the SRM (and in the PSRM), the final dimensionality of the PSRM is

$$d(\nu) = 39 \cdot 2 \cdot T \cdot \nu, \tag{4.3.10}$$

where ν is the number of projection neighborhoods for each residual.

4.3.2.3 Parameter setting

To construct the PSRM, we need to set the following parameters:

- ν ... the number of projection neighborhoods $\mathbf{\Pi}$ per residual;
- T ... the number of bins per projection neighborhood;
- s ... the maximum size of the projection neighborhood;
- q ... the bin width.

To capture a variety of complex dependencies among the neighboring residual samples, ν should be sufficiently large. Since larger ν increases the dimensionality of the feature space, $d(\nu)$, a reasonable balance must be stricken between feature dimensionality and detection accuracy.

Another parameter that influences the dimensionality is T – the number of bins per projection neighborhood. As mentioned in Section 4.3.2.1, the detection utilizes mainly the shape of the distribution, which is disturbed by the embedding process. Our experiments indicate that the number of bins necessary to describe the shape of the distribution of the projections can be rather small.

Figure 4.3.1 shows the detection–dimensionality tradeoff for different values of $d(\nu)$ and $T \in \{1, \ldots, 5\}$. The PSRM can clearly achieve the same detection reliability as SRM (SRMQ1) with much smaller dimensionality. One can trade a smaller value of T for larger ν to increase the performance for a fixed dimensionality. When choosing $\nu = 55$ and $T = 3$, the total dimensionality of the PSRM is $39 \cdot 2 \cdot T \cdot \nu = 12,870$, which makes its dimensionality almost the same of that of SRMQ1 $(12,753)$, allowing thus a direct comparison of both models. We opted for $T = 3$ as opposed to $T = 2$ because

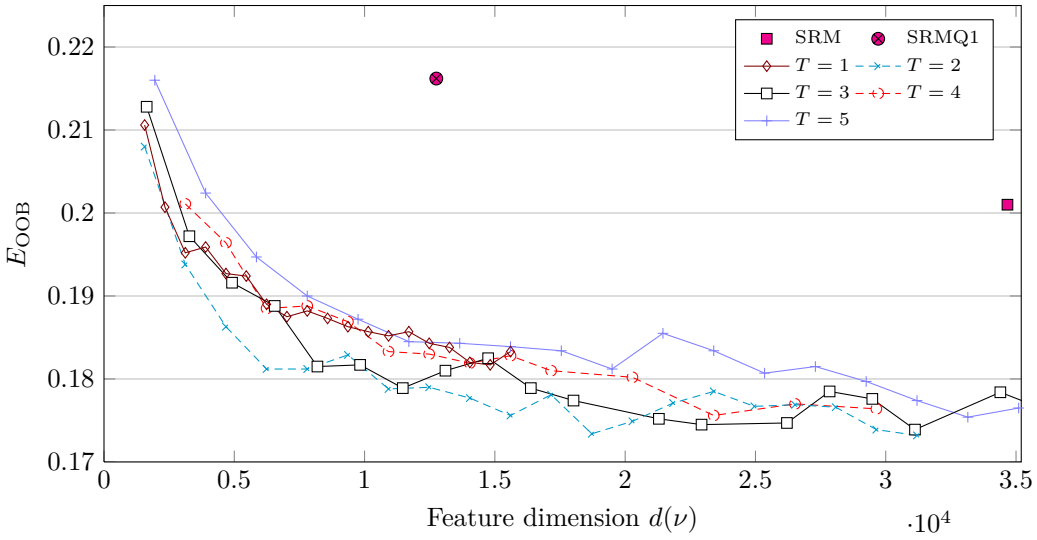

Figure 4.3.1: Detection error E_{OOB} as a function of the PSRM feature-vector dimensionality $d(\nu)$ for $T \in \{1, \ldots, 5\}$ quantization bins per projection. Tested on S-UNIWARD on BOSSbase 1.01 at payload 0.4 bpp (bits per pixel).

the performance for both choices is fairly similar and the choice $T = 3$ requires computing fewer projections for a fixed dimensionality, making the feature computation less computationally taxing.

The parameter s determines the maximal width and height of each projection neighborhood and thus limits the range of interpixel dependencies that can be utilized for detection. On the other hand, if the neighborhood is too large, the changes in the residual caused by embedding will have a small impact on the projection values, which will also become more dependent on the content. Moreover, the optimal value of s is likely to depend on the cover source. Experiments on BOSSbase 1.01 with S-UNIWARD at payload 0.4 bpp indicated a rather flat minimum around $s = 8$. We fixed s at this value and used it for all our experiments reported in this section.

To capture the shape of the distribution, it is necessary to quantize the projection values. The impact of embedding manifests in the spatial domain differently depending on whether the actual embedding changes are executed in the spatial or the JPEG domain. Given the nature of JPEG compression, a change in a DCT coefficient has a more severe impact in the spatial domain depending on the quantization step of the particular DCT mode. Consequently, the best quantization bin width q will likely be different for detection of spatial- and JPEG-domain steganography. Figure 4.3.2 shows that the optimal value of q for spatial-domain embedding is $q = 1$, while the best value of q for steganalysis of JPEG-domain steganography is $q = 3$ (Figure 4.3.3). The PSRM versions used to detect embedding in the spatial and JPEG domains will be called PSRMQ1 and PSRMQ3, respectively.

4.3.3 Experiments

To evaluate the performance of the PSRM with dimension of $12,870$, we ran experiments on multiple steganographic algorithms that embed messages in different domains. We contrast the results against several state-of-the-art domain-specific feature sets. To show the universality of the proposed detection scheme, we added experiments on a markedly different cover source – the Leica database described in Section 2.3.

In the spatial domain, we compare the PSRM with the SRM [43] (dimension $34,671$) and the SRMQ1 (dimension $12,753$).

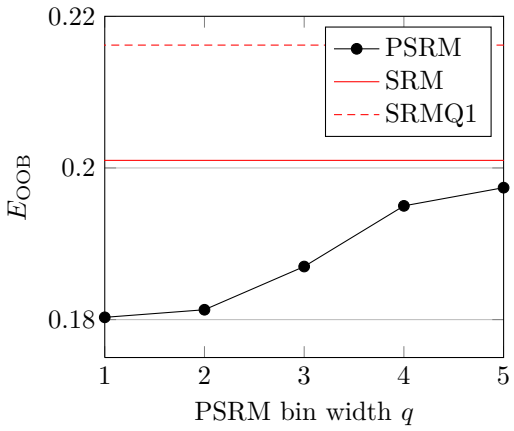

Figure 4.3.2: Detection error as a function of the quantization bin width q when steganalyzing S-UNIWARD on BOSSbase at 0.4 bpp.

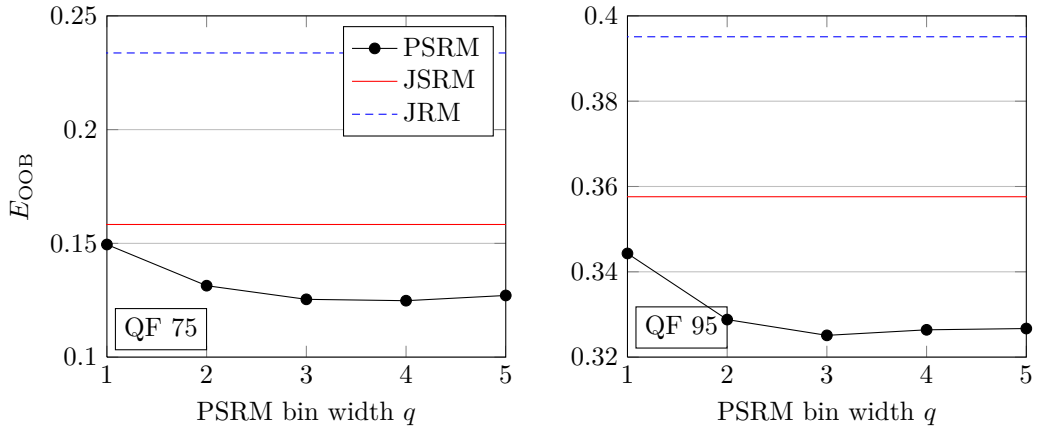

Figure 4.3.3: Detection error as a function of the quantization bin width when steganalyzing q J-UNIWARD on BOSSbase compressed using quality factors 75 and 95.

Payload	0.1 bpp			0.2 bpp			0.4 bpp		
Features	PSRMQ1	SRMQ1	SRM	PSRMQ1	SRMQ1	SRM	PSRMQ1	SRMQ1	SRM
Dimension	12,870	12,753	34,671	12,870	12,753	34,671	12,870	12,753	34,671
				BOSSbase					
HUGO	0.3564	0.3757	0.3651	0.2397	0.2701	0.2542	0.1172	0.1383	0.1278
WOW	0.3859	0.4119	0.3958	0.2950	0.3302	0.3117	0.1767	0.2170	0.1991
S-UNIWARD	0.3977	0.4182	0.4139	0.3025	0.3358	0.3159	0.1803	0.2162	0.2010
				Leica					
HUGO	0.2170	0.2273	0.2110	0.0857	0.0802	0.0723	0.0213	0.0187	0.0177
WOW	0.2438	0.2418	0.2275	0.0997	0.0993	0.0903	0.0273	0.0245	0.0197
S-UNIWARD	0.2131	0.2188	0.2023	0.0800	0.0787	0.0722	0.0198	0.0192	0.0190

Table 4.7: Detection error of PSRM vs. SRMQ1 and SRM for three content-adaptive steganographic algorithms embedding in the spatial domain.

For JPEG-domain steganography, we compare with three rich models – the SRMQ1, the JPEG Rich Model (JRM) [74] with the dimension of $22,510$, and JSRM, which is a merger of JRM and SRMQ1 with the total dimension of $35,263$. Based on a thorough comparison reported in [74], the JSRM is currently the most powerful feature set for detection of JPEG domain steganography.

The empirical steganographic security in the JPEG domain is tested on two JPEG quality factors (QF) – 75 and 95. We selected these two quality factors as typical representatives of low quality and high quality compression factors.

We evaluate the performance of all feature sets on three payloads: 0.1, 0.2, and 0.4 bits per pixel (bpp) in the spatial domain and 0.1, 0.2, and 0.4 bits per non-zero AC coefficient (bpnzAC) in the JPEG domain. The main reason for using only three payloads is the high computational complexity involved with testing high-dimensional features on many algorithms covering three embedding domains. Moreover, as will become apparent from the experimental results revealed in the next section, showing the detection accuracy on a small, medium, and a large payload seems to provide sufficient information to compare the proposed PSRM with prior art.

In order to assess the statistical significance of the results, we measured the standard deviation of the E_{OOB} for all PSRM experiments measured on ten runs of the ensemble classifier with different seeds for its random generator that drives the selection of random subspaces as well as the bootstrapping for the training sets. The standard deviation was always below 0.3 %. We do not show it in the tables below to save on space and make the table data legible. The best performing features for every cover source, steganographic algorithm, and payload are highlighted in gray.

4.3.3.1 Spatial domain

We first interpret the results on BOSSbase 1.01 shown in Table 4.7. Across all three embedding algorithms and payloads, the PSRM achieves a lower detection error than both SRMQ1 and SRM despite its almost three times larger dimensionality. Since the PSRM uses the same residuals as both SRM sets, it is safe to say that, for this image source, representing the residuals with projections is more efficient for steganalysis than forming co-occurrences. The actual improvement depends on the embedding algorithm. For HUGO, the PSRM lowers the detection error by about 2% w.r.t. the similar size SRMQ1. In light of the results of the BOSS competition reported at the 11th Information Hiding Conference [46, 45, 50, 4], this is a significant improvement. The difference between PSRMQ1 and SRMQ1 sets is even bigger ($\approx 4\%$) for the highly adaptive WOW. This confirms our intuition that the projections do capture more complex interpixel dependencies and use them more efficiently for detection.

Table 4.7 clearly shows that steganalysis is easier in Leica images than in BOSSbase. This is mainly because of stronger interpixel dependencies in Leica images. Image downsampling without

Payload	QF	0.1 bpnzAC					0.2 bpnzAC					0.4 bpnzAC				
Features		PSRMQ3	SRMQ1	JRM	JPSRM	JSRM	PSRMQ3	SRMQ1	JRM	JPSRM	JSRM	PSRMQ3	SRMQ1	JRM	JPSRM	JSRM
Dimension		12,870	12,753	22,510	35,380	35,263	12,870	12,753	22,510	35,380	35,263	12,870	12,753	22,510	35,380	35,263
BOSSbase																
nsF5	75	0.2609	0.2949	0.2115	0.1631	0.1742	0.0810	0.1162	0.0477	0.0188	0.0239	0.0057	0.0123	0.0036	0.0008	0.0013
UED ternary		0.3369	0.3621	0.3968	0.3393	0.3468	0.1856	0.2180	0.2680	0.1770	0.1934	0.0390	0.0612	0.0488	0.0202	0.0250
J-UNIWARD		0.4319	0.4578	0.4632	0.4350	0.4503	0.3244	0.3779	0.3990	0.3289	0.3564	0.1294	0.1933	0.2376	0.1228	0.1583
nsF5	95	0.3401	0.3831	0.1354	0.1220	0.1347	0.1749	0.2332	0.0114	0.0101	0.0089	0.0252	0.0540	0.0005	0.0005	0.0006
UED ternary		0.4785	0.4753	0.4750	0.4727	0.4786	0.4370	0.4331	0.4336	0.4133	0.4077	0.2759	0.2897	0.2604	0.2180	0.2205
J-UNIWARD		0.4943	0.4965	0.4923	0.4920	0.4940	0.4659	0.4752	0.4763	0.4622	0.4674	0.3256	0.3786	0.3951	0.3246	0.3576
Leica																
nsF5	75	0.2780	0.2965	0.2463	0.2040	0.2100	0.1060	0.1085	0.0783	0.0503	0.0458	0.0135	0.0114	0.0070	0.0047	0.0042
UED ternary		0.3028	0.3290	0.3643	0.2965	0.2987	0.1437	0.1570	0.2233	0.1295	0.1398	0.0270	0.0293	0.0525	0.0205	0.0200
J-UNIWARD		0.3627	0.3895	0.4233	0.3777	0.3803	0.2227	0.2538	0.3438	0.2225	0.2317	0.0610	0.0683	0.1398	0.0538	0.0593
nsF5	95	0.3833	0.4080	0.1425	0.1428	0.1370	0.2313	0.2580	0.0078	0.0090	0.0072	0.0473	0.0575	0.0002	0.0002	0.0002
UED ternary		0.4793	0.4792	0.4827	0.4767	0.4703	0.4283	0.4373	0.4410	0.4200	0.4115	0.2898	0.3020	0.2555	0.2300	0.2137
J-UNIWARD		0.4769	0.4802	0.4893	0.4797	0.4728	0.4363	0.4448	0.4517	0.4335	0.4315	0.3154	0.3380	0.3552	0.2940	0.2942

Table 4.8: Detection error of PSRM vs. JRM and JSRM for three JPEG-domain steganographic algorithms and quality factors 75 and 95.

antialiasing used to create BOSSbase images weakens the dependencies and makes the detection more difficult [75]. Moreover, the BOSSbase database was acquired by seven different cameras, which makes it likely more difficult for the machine learning to find the separating hyperplane.

While we observed a significant detection improvement over the SRM for BOSSbase for the Leica database both PSRM and SRMQ1 offer a similar detection accuracy. The reader should realize that while the SRM achieves overall the lowest detection error, comparing SRM with PSRMQ1 is not really fair as the SRM has almost three times larger dimensionality. Since the parameters of both the PSRM and the SRM sets were optimized for maximal detection on BOSSbase, we attribute this observation to the fact that the much stronger pixel dependencies in Leica images make the co-occurrence bins much better populated, which improves the steganalysis.

4.3.3.2 JPEG domain

Table 4.8 shows the results of all experiments in the JPEG domain on both BOSSbase and Leica databases for quality factors 75 and 95. In most cases, the PSRMQ3 achieved a lower detection error than SRMQ1, further fostering the claim already made in the previous section – that the projections are better suited for steganalysis than co-occurrences.

The JRM feature set, designed to utilize dependencies among DCT coefficients, shows a rather interesting behavior. Depending on the embedding algorithm and the embedding operation, the JRM's performance can be significantly better or worse than the performance of the spatial features (versions of PSRM and SRM). For example, the probability of detection error for the (by far) weakest nsF5 algorithm with payload 0.1 bpnzAC for quality factor 95 on BOSSbase using JRM is 13.54 % while it is 34.01 % for PSRMQ3 and 38.31 % for SRMQ1. This is caused by the nsF5's embedding operation designed to always decrease the absolute value of DCT coefficients. The JRM feature set is designed to exploit the effects of this "faulty" embedding operation. On the other hand, a qualitatively opposite behavior is observed for J-UNIWARD, which minimizes the relative distortion in the wavelet domain (see Chapter 5). Here, the spatial-domain features are generally much more effective than JRM since the embedding operation does not introduce artifacts in the distribution of quantized DCT coefficients detectable by the JRM.

As proposed in [71] and later confirmed in [74], the overall best detection of JPEG domain embedding algorithms is typically achieved by merging JPEG and spatial-domain features. It thus makes sense to introduce the merger of PSRMQ3 and JRM (JPSRM) whose dimensionality is similar to that of the JSRM (a merger of SRMQ1 and JRM). As expected, the JPSRM / JSRM provide the lowest detection error when compared to feature sets constrained to a specific embedding domain. On BOSSbase, the projection-based models provided the lowest detection error for almost all combinations of payload, embedding algorithm, and quality factor. On Leica, the performance of both JPSRM and JSRM was rather similar. Again, we attribute this to the fact that for the Leica source,

Payload	QF	0.1 bpnzAC					0.2 bpnzAC					0.4 bpnzAC				
Features		PSRMQ3	SRMQ1	JRM	JPSRM	JSRM	PSRMQ3	SRMQ1	JRM	JPSRM	JSRM	PSRMQ3	SRMQ1	JRM	JPSRM	JSRM
Dimension		12,870	12,753	22,510	35,380	35,263	12,870	12,753	22,510	35,380	35,263	12,870	12,753	22,510	35,380	35,263
BOSSbase																
NPQ	75	0.4613	0.4677	0.4139	0.4076	0.4078	0.3609	0.3899	0.3171	0.2779	0.2871	0.0760	0.0990	0.0654	0.0345	0.0398
SI-UNIWARD		0.4952	0.4948	0.5004	0.4970	0.4965	0.4764	0.4872	0.4908	0.4770	0.4814	0.3744	0.4083	0.4470	0.3755	0.3989
NPQ	95	0.4950	0.4960	0.4295	0.4308	0.4313	0.4708	0.4708	0.3155	0.3136	0.3095	0.3358	0.3556	0.1471	0.1342	0.1349
SI-UNIWARD		0.4955	0.4950	0.4654	0.4672	0.4696	0.4830	0.4890	0.4651	0.4599	0.4602	0.3909	0.4337	0.4418	0.3790	0.4153
NPQ	75	0.4615	0.4637	0.4257	0.4127	0.4138	0.3457	0.3545	0.3257	0.2903	0.2968	0.0802	0.0862	0.0852	0.0483	0.0508
SI-UNIWARD		0.4933	0.4960	0.4963	0.4952	0.4953	0.4727	0.4777	0.4900	0.4848	0.4748	0.3712	0.3872	0.4473	0.3752	0.3802
NPQ	95	0.4868	0.4920	0.3435	0.3505	0.3518	0.4682	0.4785	0.2920	0.3030	0.2998	0.3727	0.3773	0.1660	0.1628	0.1477
SI-UNIWARD		0.4908	0.4957	0.4460	0.4415	0.4475	0.4872	0.4973	0.4480	0.4448	0.4563	0.4312	0.4475	0.4450	0.4083	0.4220

Table 4.9: Detection error of PSRM vs. JRM and JSRM for two side-informed JPEG-domain steganographic algorithms and quality factors 75 and 95.

the co-occurrences are generally better populated than for the BOSSbase. Finally, we would like to point out that for J-UNIWARD adding the JRM to PSRMQ3 generally brings only a rather negligible improvement, indicating that the main detection power resides in the spatial features (the PSRMQ3).

4.3.3.3 Side-informed JPEG domain

The performance comparison for side-informed JPEG-domain embedding methods shown in Table 4.9 strongly resembles the conclusions from the previous section. The merged feature spaces (JPSRM and JSRM) generally provide the lowest detection error when considering the statistical spread of the data (0.3%). It is worth pointing out that the JRM features are rather effective against the NPQ algorithm (see, e.g., the quality factor 95 and payload 0.4 bpnzAC). This indicates a presence of artifacts in the distribution of DCT coefficients that are well detected with the JRM, which further implies that the NPQ algorithm determines the embedding costs in the DCT domain in a rather suboptimal way. Also note that the detection errors for BOSSbase and Leica are much more similar in the JPEG domain when compared with the spatial domain. This is likely an effect of the lossy character of JPEG compression, which "erases" the high-frequency details (differences) between both sources.

4.3.4 Conclusion

The key element in steganalysis of digital images using machine learning is their representation. Over the years, researchers converged towards a de facto standard representation that starts with computing a noise residual and then taking the sample joint distribution of residual samples as a feature for steganalysis. This co-occurrence based approach dominated the field for the past seven years. Co-occurrences, however, are rather non-homogeneous descriptors. With an increasing co-occurrence order, a large number of bins become underpopulated (statistically less significant), which leads to a feature dimensionality increase disproportional to the gain in detection performance. The co-occurrence order one can use in practice is thus limited, which prevents steganalysts from utilizing long-range dependencies among pixels that might further improve detection especially for content-adaptive steganographic schemes.

Aware of these limitations, in this article, we introduce an alternative statistical descriptor of residuals by projecting neighboring residual samples onto random directions and taking the first-order statistics of the projections as features. The resulting features are better populated and thus more statistically significant. Furthermore, the projection vectors as well as the size and shape of the projection neighborhoods further diversify the description, which boosts detection accuracy. The advantage of representing images using residual projections as opposed to co-occurrences is demonstrated on several state-of-the-art embedding algorithms in the spatial, JPEG, and side-informed JPEG domains.

The new representation is called the projection spatial rich model (PSRM). We introduce two versions – one suitable for detection of spatial-domain steganography and one for the JPEG domain. Both versions differ merely in the quantization step used to quantize the projections. The PSRM is based on the exact same set of noise residuals as its predecessor – the spatial rich model. The fact that PSRM equipped with the same set of residuals as the SRM offers a better detection performance at the same dimensionality is indicative of the fact that the projections are indeed more efficient for steganalysis than co-occurrences.

The biggest advantage of PSRM over SRM becomes apparent for highly content adaptive algorithms, such as WOW or schemes employing the UNIWARD function. Besides a more accurate detection, the PSRM also enjoys a much better performance vs. dimensionality ratio. For spatial-domain algorithms, one can achieve the same detection accuracy as that of SRM with dimensionality 7–10 times smaller. This compactification, however, comes at a price, which is the computational complexity. This seems inevitable if one desires a descriptor that is more statistically relevant and diverse – the PSRM consists of a large number of projection histograms rather than a small(er) number of high-dimensional co-occurrences. The PSRM feature computation requires computing about 65,000 convolutions and histograms. A possible speed-up of the PSRM feature computation using graphical processing units (GPUs) was proposed in [63]. The PSRM feature extractor is available from `http://dde.binghamton.edu/download/feature_extractors/`.

Finally, we make one more intriguing remark. The latest generation of currently most secure algorithms that embed messages in quantized DCT coefficients but minimize the embedding distortion computed in the spatial (wavelet) domain (J-UNIWARD and SI-UNIWARD) seem to be less detectable using features computed from quantized DCT coefficients and become, instead, more detectable using spatial-domain features (PSRM). This challenges the long heralded principle that the best detection is always achieved in the embedding domain. Unless the embedding rule is flawed (e.g, the embedding operation of LSB flipping or the F5 embedding operation), one should consider for detection representing the images in the domain in which the distortion is minimized.

Chapter 5

Steganography using universal wavelet relative distortion

This chapter contains a slighlty modified version of author's article published in EURASIP Journal on Information Security 2014.

Currently, the most successful approach to steganography in empirical objects, such as digital media, is to embed the payload while minimizing a suitably defined distortion function. The design of the distortion is essentially the only task left to the steganographer since efficient practical codes exist that embed near the payload–distortion bound. The practitioner's goal is to design the distortion to obtain a scheme with a high empirical statistical detectability. In this chapter, we propose a universal distortion design called UNIWARD (UNIversal WAvelet Relative Distortion) that can be applied for embedding in an arbitrary domain. The embedding distortion is computed as a sum of relative changes of coefficients in a directional filter bank decomposition of the cover image. The directionality forces the embedding changes to such parts of the cover object that are difficult to model in multiple directions, such as textures or noisy regions, while avoiding smooth regions or clean edges. We demonstrate experimentally using rich models as well as targeted attacks that steganographic methods built using UNIWARD match or outperform the current state of the art in the spatial domain, JPEG domain, and side-informed JPEG domain.

5.1 Introduction

Designing steganographic algorithms for empirical cover sources [9] is very challenging due to the fundamental lack of accurate models. The most successful approach today avoids estimating (and preserving) the cover source distribution because this task is infeasible for complex and highly non-stationary sources, such as digital images. Instead, message embedding is formulated as source coding with a fidelity constraint [97] – the sender hides her message while minimizing an embedding distortion. Practical embedding algorithms that operate near the theoretical payload–distortion bound are available for a rather general class of distortion functions [35, 33].

The key element of this general framework is the distortion, which needs to be designed in such a way that tests on real imagery indicate a high level of security.[1] In [34], a heuristically-defined distortion function was parametrized and then optimized to obtain the smallest detectability in terms of a margin between classes within a selected feature space (cover model). However, unless the cover model is a complete statistical descriptor of the empirical source, such optimized schemes may, paradoxically, end up being more detectable if the Warden designs the detector "outside of

[1] For a given empirical cover source, the statistical detectability is typically evaluated empirically using classifiers trained on cover and stego examples from the source.

the model" [10, 76], which brings us back to the main and rather difficult problem – modeling the source.

In the JPEG domain, by far the most successful paradigm is to minimize the rounding distortion w.r.t. the raw, uncompressed image, if available [68, 95, 107, 60, 29]. In fact, this "side-informed embedding" can be applied whenever the sender possesses a higher-quality "precover"[2] that is quantized to obtain the cover.[3] Currently, the most secure embedding method for JPEG images that does not use any side information is the Uniform Embedding Distortion (UED) [51] that substantially improved upon the nsF5 algorithm [47] – the previous state of the art. Note that most embedding algorithms for the JPEG format use only non-zero DCT coefficients, which makes them naturally content-adaptive.

In the spatial domain, embedding costs are typically required to be low in complex textures or "noisy" areas and high in smooth regions. For example, HUGO [90] defines the distortion as a weighted norm between higher-order statistics of pixel differences in cover and stego images [89], with high weights assigned to well-populated bins and low weights to sparsely populated bins that correspond to more complex content. An alternative model-free approach called WOW (Wavelet Obtained Weights) [54] uses a bank of directional high-pass filters to obtain the so-called *directional residuals*, which assess the content around each pixel along multiple different directions. By measuring the impact of embedding on every directional residual and by suitably aggregating these impacts, WOW forces the distortion to be high where the content is predictable in *at least one* direction (smooth areas and clean edges) and low where the content is unpredictable in every direction (as in textures). The resulting algorithm is highly adaptive and has been shown to better resists steganalysis using rich models [43] than HUGO [54].

The distortion function proposed in this chapter bears similarity to that of WOW but is simpler and suitable for embedding in an arbitrary domain. Since the distortion is in the form of a sum of *relative* changes between the stego and cover images represented in the wavelet domain, hence its name: UNIversal WAvelet Relative Distortion (UNIWARD).

We describe the distortion function in its most general form in Section 5.2 – one suitable for embedding in both the spatial and JPEG domains and the other for side-informed JPEG steganography. We also describe the additive approximation of UNIWARD that will be exclusively used in this chapter. A study of the best settings for UNIWARD, formed by the choice of the directional filter bank and a stabilizing constant, appear in Section 5.3. Section 5.4 contains the results of all experiments in the spatial, JPEG, and side-informed JPEG domains as well as the comparison with previous art. The security is measured empirically using classifiers trained with rich media models on a range of payloads and quality factors. The chapter is concluded in Section 5.5.

5.2 Universal distortion function UNIWARD

In this section, we provide a general description of the proposed universal distortion function UNIWARD and explain how it can be used to embed in the JPEG and the side-informed JPEG domains. The distortion depends on the choice of a directional filter bank and one scalar parameter whose purpose is stabilizing the numerical computations. The distortion design is finished in the next Section 5.3, which investigates the effect of the filter bank and the stabilizing constant on empirical security.

Since rich models [46, 43, 50, 100] currently used in steganalysis are capable of detecting changes along "clean edges" that can be well fitted using locally polynomial models, whenever possible the embedding algorithm should embed into textured/noisy areas that are not easily modellable in any

[2]The concept of precover was used for the first time by Ker [64].

[3]Historically, the first side-informed embedding method was the Embedding While Dithering algorithm [39], in which a message was embedded to minimize the color quantization error when converting a true-color image to a palette image.

direction. We quantify this using outputs of a directional filter bank and construct the distortion function in this manner.

5.2.1 Directional filter bank

By a directional filter bank, we understand a set of three linear shift-invariant filters represented with their kernels $\mathcal{B} = \{\mathbf{K}^{(1)}, \mathbf{K}^{(2)}, \mathbf{K}^{(3)}\}$. They are used to evaluate the smoothness of a given image $\mathbf{X}$ along the horizontal, vertical, and diagonal direction by computing the so-called directional residuals $\mathbf{W}^{(k)} = \mathbf{K}^{(k)} \star \mathbf{X}$, where '$\star$' is a mirror-padded convolution so that $\mathbf{W}^{(k)}$ has again $n_1 \times n_2$ elements. The mirror-padding prevents introducing embedding artifacts at the image boundary.

While it is possible to use arbitrary filter banks, we will exclusively use kernels built from one-dimensional low-pass (and high-pass) wavelet decomposition filters $\mathbf{h}$ (and $\mathbf{g}$):

$$\mathbf{K}^{(1)} = \mathbf{h} \cdot \mathbf{g}^{\mathrm{T}},\ \mathbf{K}^{(2)} = \mathbf{g} \cdot \mathbf{h}^{\mathrm{T}},\ \mathbf{K}^{(3)} = \mathbf{g} \cdot \mathbf{g}^{\mathrm{T}}. \tag{5.2.1}$$

In this case, the filters correspond, respectively, to two-dimensional LH, HL, and HH wavelet directional high-pass filters and the residuals coincide with the first-level undecimated wavelet LH, HL, and HH directional decomposition of $\mathbf{X}$. We constrained ourselves to wavelet filter banks because wavelet representations are known to provide good decorrelation and energy compactification for images of natural scenes (see, e.g., Chapter 7 in [106]).

5.2.2 Distortion function (non-side-informed embedding)

We are now ready to describe the universal distortion function. We do so first for embedding that does not use any precover. Given a pair of cover and stego images, $\mathbf{X}$, and $\mathbf{Y}$, represented in the spatial (pixel) domain, we will denote with $W_{uv}^{(k)}(\mathbf{X})$ and $W_{uv}^{(k)}(\mathbf{Y})$, $k = 1, 2, 3$, $u \in \{1, \dots, n_1\}$, $v \in \{1, \dots, n_2\}$, their corresponding uvth wavelet coefficient in the kth subband of the first decomposition level. The UNIWARD distortion function is the sum of relative changes of all wavelet coefficients w.r.t. the cover image:

$$D(\mathbf{X}, \mathbf{Y}) \triangleq \sum_{k=1}^{3} \sum_{u=1}^{n_1} \sum_{v=1}^{n_2} \frac{|W_{uv}^{(k)}(\mathbf{X}) - W_{uv}^{(k)}(\mathbf{Y})|}{\sigma + |W_{uv}^{(k)}(\mathbf{X})|}, \tag{5.2.2}$$

where $\sigma > 0$ is a constant stabilizing the numerical calculations.

The ratio in (5.2.2) is smaller when a large cover wavelet coefficient is changed (where texture and edges appear). Embedding changes are discouraged in regions where $|W_{uv}^{(k)}(\mathbf{X})|$ is small for at least one k, which corresponds to a direction along which the content is modellable.

For JPEG images, the distortion between the two arrays of quantized DCT coefficients, $\mathbf{X}$ and $\mathbf{Y}$, is computed by first decompressing the JPEG files to the spatial domain, and evaluating the distortion between the decompressed images, $J^{-1}(\mathbf{X})$ and $J^{-1}(\mathbf{Y})$, in the same manner as in (5.2.2):

$$D(\mathbf{X}, \mathbf{Y}) \triangleq D\left(J^{-1}(\mathbf{X}), J^{-1}(\mathbf{Y})\right). \tag{5.2.3}$$

Note that the distortion (5.2.2) is non-additive because changing pixel X_{ij} will affect $s \times s$ wavelet coefficients, where $s \times s$ is the size of the 2D wavelet support. Also, changing a JPEG coefficient X_{ij} will affect a block of 8×8 pixels and therefore a block of $(8 + s - 1) \times (8 + s - 1)$ wavelet coefficients. It is thus apparent that when changing neighboring pixels (or DCT coefficients), the embedding changes "interact," hence the non-additivity of D.

5.2.3 Distortion function (JPEG side-informed embedding)

By side-informed embedding in JPEG domain, we understand the following general principle. Given the raw DCT coefficient D_{ij} obtained from the precover $\mathbf{P}$, the embedder has the choice of rounding D_{ij} up or down to modulate its parity (usually the least significant bit of the rounded value). We denote with $e_{ij} = |D_{ij} - X_{ij}|$, $e_{ij} \in [0, 0.5]$, the rounding error for the ijth coefficient when compressing the precover $\mathbf{P}$ to the cover image $\mathbf{X}$. Rounding "to the other side" leads to an embedding change, $Y_{ij} = X_{ij} + \text{sign}(D_{ij} - X_{ij})$, which corresponds to a "rounding error" $1 - e_{ij}$. Thus, every embedding change increases the distortion *w.r.t. the precover* by the difference between both rounding errors: $|D_{ij} - Y_{ij}| - |D_{ij} - X_{ij}| = 1 - 2e_{ij}$. For the side-informed embedding in JPEG domain, we therefore define the distortion as the difference:

$$\begin{aligned} D^{(\text{SI})}(\mathbf{X}, \mathbf{Y}) &\triangleq D\left(\mathbf{P}, J^{-1}(\mathbf{Y})\right) - D\left(\mathbf{P}, J^{-1}(\mathbf{X})\right) \\ &= \sum_{k=1}^{3}\sum_{u=1}^{n_1}\sum_{v=1}^{n_2}\left[\frac{|W_{uv}^{(k)}(\mathbf{P}) - W_{uv}^{(k)}\left(J^{-1}(\mathbf{Y})\right)|}{\sigma + |W_{uv}^{(k)}(\mathbf{P})|} - \frac{|W_{uv}^{(k)}(\mathbf{P}) - W_{uv}^{(k)}\left(J^{-1}(\mathbf{X})\right)|}{\sigma + |W_{uv}^{(k)}(\mathbf{P})|}\right] \end{aligned} \tag{5.2.4}$$

Note that the linearity of DCT and the wavelet transforms guarantee that $D^{(\text{SI})}(\mathbf{X}, \mathbf{Y}) \geq 0$. This is because rounding a DCT coefficient (to obtain $\mathbf{X}$) corresponds to adding a certain pattern (that depends on the modified DCT mode) in the wavelet domain. Rounding "to the other side" (to obtain $\mathbf{Y}$) corresponds to subtracting the same pattern but with a *larger* amplitude. This is why $|W_{uv}^{(k)}(\mathbf{P}) - W_{uv}^{(k)}(J^{-1}(\mathbf{Y}))| - |W_{uv}^{(k)}(\mathbf{P}) - W_{uv}^{(k)}(J^{-1}(\mathbf{X}))| \geq 0$ for all k, u, v.

We note at this point that (5.2.4) bears some similarity to the distortion used in Normalized Perturbed Quantization (NPQ) [60, 29], where the authors also proposed the distortion as a *relative* change of cover DCT coefficients. The main difference is that we compute the distortion using a directional filter bank, allowing thus directional sensitivity and potentially better content adaptability. Furthermore, we do not eliminate DCT coefficients that are zeros in the cover. Finally, and most importantly, in contrast to NPQ our design naturally incorporates the effect of the quantization step because the wavelet coefficients are computed from the decompressed JPEG image.

5.2.3.1 Technical issues with zero embedding costs

When running experiments with *any* side-informed JPEG steganography in which the embedding cost is zero, when $e_{ij} = 1/2$, we discovered a technical problem that, to the best knowledge of the authors, has not been disclosed elsewhere. The problem is connected to the fact that when $e_{ij} = 1/2$ the cost of rounding D_{ij} "down" instead of "up" should not be zero because, after all, this does constitute an embedding change. This does not affect security much when the number of such DCT coefficients is small. With an increasing number of coefficients with $e_{ij} = 1/2$ (we will call them 1/2-coefficients), however, $1 - 2e_{ij}$ is no longer a good measure of statistical detectability and one starts observing a rather pathological behavior – with payload approaching zero, the detection error does not saturate at 50% (random guessing) but rather at a lower value and only reaches 50% for payloads nearly equal to zero.[4] The strength with which this phenomenon manifests depends on how many 1/2-coefficients are in the image, which in turn depends on two factors – the implementation of the DCT used to compute the costs and the JPEG quality factor. When using the slow DCT (implemented using 'dct2' in Matlab), the number 1/2-coefficients is small and does not affect security at least for low quality factors. However, in the fast-integer implementation of DCT (e.g., Matlab's 'imwrite'), all D_{ij} are multiples of 1/8. Thus, with decreasing quantization step (increasing JPEG quality factor), the number of 1/2-coefficients increases.

[4] This is because the embedding strongly prefers 1/2-coefficients.

To avoid dealing with this issue in this chapter, we used the slow DCT implemented using Matlab's 'dct2' as explained in Section 2.2.2 to obtain the costs. Even with the slow DCT, however, 1/2-coefficients do cause problems when the quality factor is high. As one can easily verify from the formula for the DCT (2.2.2), when $k, l \in \{0, 4\}$, the value of D_{kl} is always a rational number because the cosines are either 1 or $\sqrt{2}/2$, which, together with the multiplicative weights $\mathbf{w}$, gives again a rational number. In particular, the DC coefficient (mode 00) is always a multiple of 1/4, the coefficients of modes 04 and 40 are multiples of 1/8, and the coefficients corresponding to mode 44 are multiples of 1/16. For all other combinations of $k, l \in \{0, \dots, 7\}$, D_{ij} is an irrational number. In practice, *any* embedding whose costs are zero for 1/2-coefficients will thus strongly prefer these four DCT modes, causing a highly uneven distribution of embedding changes among the DCT coefficients. Because rich JPEG models [74] utilize statistics collected for each mode separately, they are capable of detecting this statistical peculiarity even at low payloads. This problem becomes more serious with increasing quality factor.

These above embedding artifacts can be largely suppressed by prohibiting embedding changes in *all* 1/2-coefficients in modes 00, 04, 40, and 44.[5] In Figure 5.4.4, where we show the comparison of various side-informed embedding methods for quality factor 95, we intentionally included the detection errors for all tested schemes where this measure was not enforced to prove the validity of the above arguments.

The solution of the problem with 1/2-coefficients, which is clearly not optimal, is related to the more fundamental problem, which is how exactly the side-information in the form of an uncompressed image should be utilized for the design of steganographic distortion functions. The authors postpone a detailed study of this quite intriguing problem for future research.

5.2.4 Additive approximation of UNIWARD

Any distortion function $D(\mathbf{X}, \mathbf{Y})$ can be used for embedding in its additive approximation [33] by using D to compute the cost ρ_{ij} of changing each pixel/DCT coefficient X_{ij}. A significant advantage of using an additive approximation is the simplicity of the overall design. The embedding can be implemented in a straightforward manner by applying nowadays a standard tool in steganography – the Syndrome-Trellis Codes (STCs) [35]. All experiments in this chapter are carried out with additive approximations of UNIWARD.

The cost of changing X_{ij} to Y_{ij}, and leaving all other cover elements unchanged, is:

$$\rho_{ij}(\mathbf{X}, Y_{ij}) \triangleq D(\mathbf{X}, \mathbf{X}_{\sim ij} Y_{ij}), \tag{5.2.5}$$

where $\mathbf{X}_{\sim ij} Y_{ij}$ is the cover image $\mathbf{X}$ with only its ijth element changed: $X_{ij} \to Y_{ij}$.[6] Note that $\rho_{ij} = 0$ when $\mathbf{X} = \mathbf{Y}$. The additive approximation to (5.2.2) and (5.2.4) will be denoted as $D_{\mathrm{A}}(\mathbf{X}, \mathbf{Y})$ and $D_{\mathrm{A}}^{(\mathrm{SI})}(\mathbf{X}, \mathbf{Y})$, respectively. For example,

$$D_{\mathrm{A}}(\mathbf{X}, \mathbf{Y}) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} \rho_{ij}(\mathbf{X}, Y_{ij})[X_{ij} \neq Y_{ij}], \tag{5.2.6}$$

where $[S]$ is the Iverson bracket equal to 1 when the statement S is true and 0 when S is false.

Note that, due to the absolute values in $D(\mathbf{X}, \mathbf{Y})$ (5.2.2), $\rho_{ij}(\mathbf{X}, X_{ij} + 1) = \rho_{ij}(\mathbf{X}, X_{ij} - 1)$, which permits us to use a *ternary* embedding operation for the spatial and JPEG domains.[7]

On the other hand, for the side-informed JPEG steganography, $D_{\mathrm{A}}^{(\mathrm{SI})}(\mathbf{X}, \mathbf{Y})$ is inherently limited to a *binary* embedding operation because D_{ij} is either rounded up or down.

[5] In practice, we assign very large costs to such coefficients.

[6] This notation was used in [33] and is also standard in the literature on Markov random fields [111].

[7] One might (seemingly rightfully) argue that the cost should depend on the polarity of the change. On the other hand, since the embedding changes with UNIWARD are restricted to textures, the equal costs are in fact plausible.

The embedding methods that use the additive approximation of UNIWARD for the spatial, JPEG, and side-informed JPEG domain will be called S-UNIWARD, J-UNIWARD, and SI-UNIWARD, respectively.

5.2.5 Relationship of UNIWARD to WOW

The distortion function of WOW bears some similarity to UNIWARD in the sense that the embedding costs are also computed from three directional residuals. The WOW embedding costs are, however, computed a different way that makes it rather difficult to use it for embedding in other domains, such as the JPEG domain.[8]

To obtain a cost of changing pixel $X_{ij} \to Y_{ij}$, WOW first computes the embedding distortion in the wavelet domain weighted by the wavelet coeffcients of the cover. This is implemented as a convolution $\xi_{ij}^{(k)} = |W_{uv}^{(k)}(\mathbf{X})| \star |W_{uv}^{(k)}(\mathbf{X}) - W_{uv}^{(k)}(\mathbf{X}_{\sim ij}Y_{ij})|$ (see Eq. (2) in [54]). These so-called "embedding suitabilities" $\xi_{ij}^{(k)}$ are then aggregated over all three subbands using the reciprocal Hölder norm, $\rho_{ij}^{(\mathrm{WOW})} = \sum_{k=1}^{3} 1/\xi_{ij}^{(k)}$ to give WOW the proper content-adaptivity in the spatial domain.

In principle, this approach could be used for embedding in the JPEG (or some other) domain in a similar way as in UNIWARD. However, notice that the suitabilities $\xi_{ij}^{(k)}$ increase with increasing JPEG quantization step (increasing spatial frequency), giving the high-frequency DCT coefficients smaller costs, $\rho_{ij}^{(\mathrm{WOW})}$, and thus a higher embedding probability than for the low-frequency coefficients. This creates both visible and statistically detectable artifacts. In contrast, the embedding costs in UNIWARD are higher for high-frequency DCT coefficients, desirably discouraging embedding changes in coefficients which are largely zeros.

5.3 Determining the parameters of UNIWARD

In this section, we study how the wavelet basis and the stabilizing constant σ in the distortion function UNIWARD affect the empirical security. We first focus on the parameter σ and then on the filter bank. All experiments in this chapter were run on the standard database BOSSbase 1.01.

The original role of σ in UNIWARD [56] was to stabilize the numerical computations when evaluating the relative change of wavelet coefficients (5.2.2). As the following experiment shows, however, σ also strongly affects the content-adaptivity of the embedding algorithm. In Figure 5.3.1, we show the embedding change probabilities for payload $\alpha = 0.4$ bpp (bits per pixel) for six values of the parameter σ. For this experiment, we selected the 8-tap Daubechies wavelet filter bank $\mathcal{B}$ whose 1D filters are shown in Table 5.1.[9] Note that a small value of σ makes the embedding change probabilities undesirably sensitive to content. They exhibit unusual interleaved streaks of high and low values. This is clearly undesirable since the content (shown in the upper left corner of Figure 5.3.1) does not change as abruptly. On the other hand, a large σ makes the embedding change probabilities "too smooth," permitting thus UNIWARD to embed in regions with less complex content. Intuitively, we need to choose some middle ground for σ to avoid introducing a weakness into the embedding algorithm.

Because the SRM consists of statistics collected from the noise residuals of all pixels in the image, it "does not see" the artifacts in the embedding probabilities – the interleaved bands of high and low values. Notice that the position of the bands is tied to the content and does not correspond to any fixed (content-independent) checkerboard pattern. Thus, we decided to introduce a new type of steganalysis features designed specifically to utilize the artifacts in the embedding probabilities to probe the security of this unusual selection channel for small values of σ.

[8] This is one of the reasons why UNIWARD was conceived.

[9] This filter bank was previously shown to provide the highest level of security for WOW [54] from among several tested filter banks. We thus selected the same bank here as a good initial candidate for the experiments.

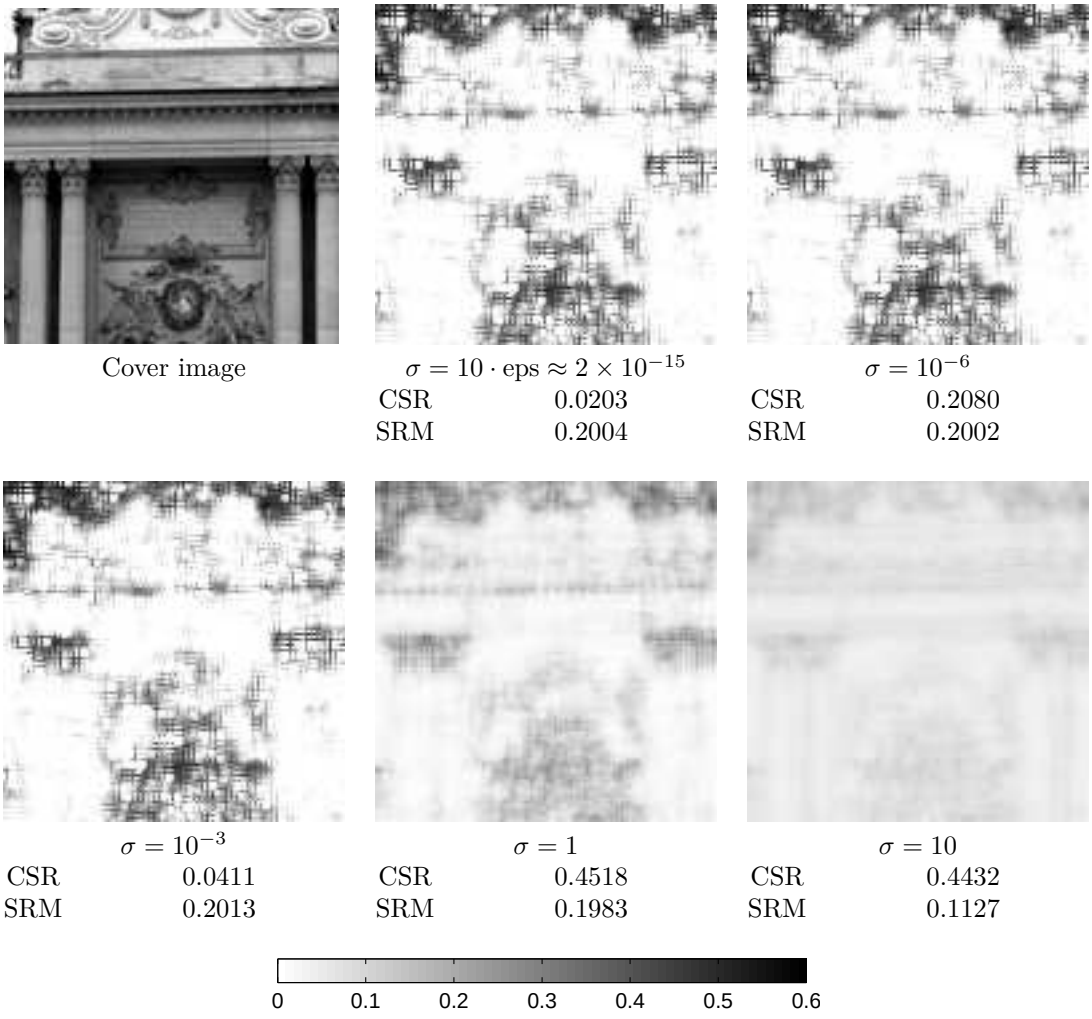

Figure 5.3.1: The effect of the stabilizing constant σ on the character of the embedding change probabilities for a 128×128 cover image shown in the upper left corner. The numerical values are the E_{OOB} obtained using the content-selective residual (CSR) and the spatial rich model (SRM) on BOSSbase 1.01 for relative payload $\alpha = 0.4$ bpp.

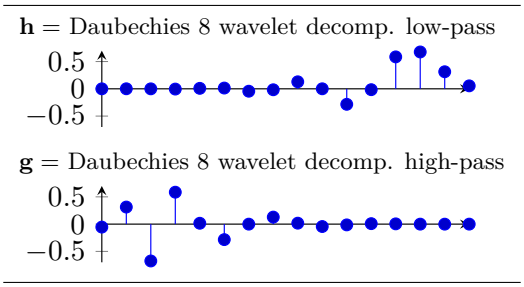

Table 5.1: UNIWARD used the Daubechies wavelet directional filter bank built from one-dimensional low-pass and high-pass filters, **h** and **g**.

5.3.1 Content-selective residuals

The idea behind the attack on the selection channel is to compute the statistics of noise residuals separately for pixels with a small embedding probability and then for pixels with a large embedding probability. The former will serve as a reference for the latter, giving strength to this attack. While it is true that the embedding probabilities estimated from the stego image will generally not exactly match those computed from the corresponding cover image,[10] they will be close and "good enough" for the attack to work.

We will use the first order noise residuals (differences among neighboring pixels):

$$R_{ij} = X_{i,j} - X_{i,j+1},\ i \in \{1, \dots, n_1\},\ j \in \{1, \dots, n_2 - 1\}. \tag{5.3.1}$$

To curb the residuals' range and allow a compact statistical representation, R_{ij} will be truncated to the range $[-T, T]$, $R_{ij} \leftarrow \mathrm{trunc}_T(R_{ij})$. Truncation is defined in Eq. (4.1.2).

Since this residual involves two adjacent pixels, we will divide all horizontally adjacent pixels in the image into four classes and compute the histogram for each class separately. Let $p_{ij}(\mathbf{X}, \overline{\alpha})$ denote the embedding change probability computed from image $\mathbf{X}$ when embedding payload of $\overline{\alpha}$ bpp. Given two thresholds $0 < t_s < t_L < 1$, we define the following four sets of residuals:

$$\mathcal{R}_{ss} = \{R_{ij} | p_{ij}(\mathbf{X}, \overline{\alpha}) < t_s \ \wedge\ p_{i,j+1}(\mathbf{X}, \overline{\alpha}) < t_s\} \tag{5.3.2}$$
$$\mathcal{R}_{sL} = \{R_{ij} | p_{ij}(\mathbf{X}, \overline{\alpha}) < t_s \ \wedge\ p_{i,j+1}(\mathbf{X}, \overline{\alpha}) > t_L\} \tag{5.3.3}$$
$$\mathcal{R}_{Ls} = \{R_{ij} | p_{ij}(\mathbf{X}, \overline{\alpha}) > t_L \ \wedge\ p_{i,j+1}(\mathbf{X}, \overline{\alpha}) < t_s\} \tag{5.3.4}$$
$$\mathcal{R}_{LL} = \{R_{ij} | p_{ij}(\mathbf{X}, \overline{\alpha}) > t_L \ \wedge\ p_{i,j+1}(\mathbf{X}, \overline{\alpha}) > t_L\}. \tag{5.3.5}$$

The so-called Content-Selective Residual (CSR) features will be formed by the histograms of residuals in each set. Because the marginal distribution of each residual is symmetrical about zero, one can merge the histograms of residuals from $\mathcal{R}_{sL}$ and $\mathcal{R}_{Ls}$. The feature vector is thus the concatenation of $3 \times (2T + 1)$ histogram bins, $l = -T, \dots, T$:

$$h_s(l) = \big|\{R_{ij} | R_{ij} = l \ \wedge\ R_{ij} \in \mathcal{R}_{ss}\}\big| \tag{5.3.6}$$
$$h_L(l) = \big|\{R_{ij} | R_{ij} = l \ \wedge\ R_{ij} \in \mathcal{R}_{LL}\}\big| \tag{5.3.7}$$
$$h_{sL}(l) = \big|\{R_{ij} | R_{ij} = l \ \wedge\ R_{ij} \in \mathcal{R}_{sL} \cup \mathcal{R}_{Ls}\}\big|. \tag{5.3.8}$$

The set $\mathcal{R}_{ss}$ holds the residual values computed from pixels with a small embedding change probability, while the other sets hold residuals that are likely affected by embedding – their tails will become thicker.

All that remains is to specify the values of the parameters t_s, t_L, and $\overline{\alpha}$. Since the steganalyst will generally not know the payload embedded in the stego image,[11] we need to choose a fixed value of $\overline{\alpha}$ that gives an overall good performance over a wide range of payloads. In our experiments, a medium value of $\overline{\alpha} = 0.4$ generally provided a good estimate of the interleaved bands in the embedding change probabilities. Finally, we conducted a grid search on images from BOSSbase to determine t_s and t_L. The found optimum was rather flat and located around $t_s = 0.05$, $t_L = 0.06$. The threshold T for $\mathrm{trunc}_T(x)$ was kept fixed at $T = 10$.

For the value of σ as originally proposed in the workshop version of this chapter [56], $\sigma = 10 \cdot \mathrm{eps} \approx 2 \times 10^{-15}$ ('eps' defined as in Matlab), the detection error of the $3 \times (2 \times 10 + 1) = 63$-dimensional CSR feature vector turned out to be a reliable detection statistic. Figure 5.3.2 shows the detection error E_{OOB} as a function of the relative payload. This confirms our intuition that too small a value of σ introduces strong banding artifacts, the stego scheme becomes overly sensitive to content, and an approximate knowledge of the faulty selection channel can be used to successfully attack S-UNIWARD.

[10] Also because the embedded payload α is unknown to the steganalyst.
[11] A study on building steganalyzers when the payload is not known appears in [86].

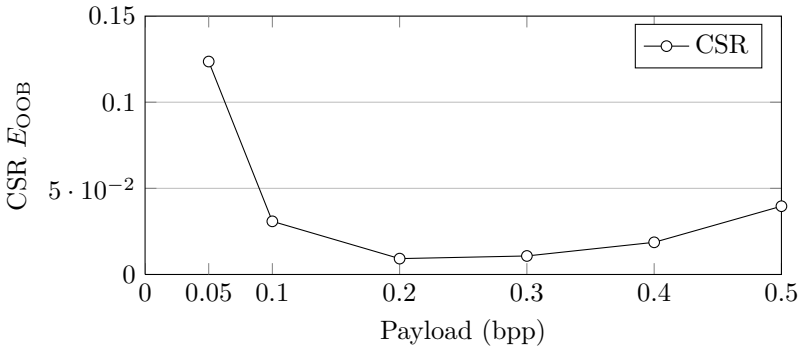

Figure 5.3.2: Detection error E_{OOB} obtained using the CSR features as a function of relative payload for $\sigma = 10 \cdot \mathrm{eps}$.

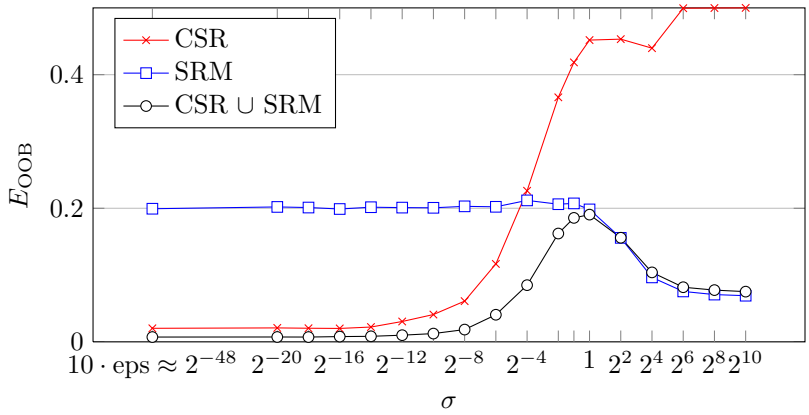

Figure 5.3.3: Detection error of S-UNIWARD with payload 0.4 bpp implemented with various values of σ for the CSR and SRM features and their union.

As can be seen from Figure 5.3.1, the artifacts in the embedding change probabilities become gradually suppressed when increasing the value of the stabilizing constant σ. To determine the proper value of σ, we steganalyzed S-UNIWARD with both the CSR and SRM feature sets (and their union) on payload $\alpha = 0.4$ bpp as a function of σ (see Figure 5.3.3).[12]The detection error using both the SRM and the CSR is basically constant until σ becomes close to 2^{-14} when a further increase of σ makes the CSR features ineffective for steganalysis. From $\sigma = 1$ the SRM starts detecting the embedding more accurately as the adaptivity of the scheme becames lower. Also, at this value of σ, adding the CSR does not lower the detection error of the SRM. Based on this analysis, we decided to set the stabilizing constant of S-UNIWARD to $\sigma = 1$ and kept it at this value for the rest of the experiments in the spatial domain reported in this chapter.

The attack based on content-selective residuals could be expanded to other residuals than pixel differences, and one could use higher-order statistics instead of histograms [103]. While the detection error for the original S-UNIWARD setting $\sigma = 10 \cdot \mathrm{eps}$ can, indeed, be made smaller this way, expanding the CSR feature set has virtually no effect on the security of S-UNIWARD for $\sigma = 1$ and the optimality of this value.

[12]When steganalyzing with the union of CSR and SRM using the ensemble classifier, we made sure that all 63 CSR features were included in each random feature subspace to avoid "diluting" their strength in this type of classifier. Also, the value of σ for extracting the embedding change probabilities $p_{ij}(\mathbf{X}; \overline{\alpha})$ was always fixed at $\sigma = 10 \cdot \mathrm{eps}$ as the location of interleaved bands of high and low probabilities are more accurately estimated this way than with the value used in S-UNIWARD for the actual message embedding.

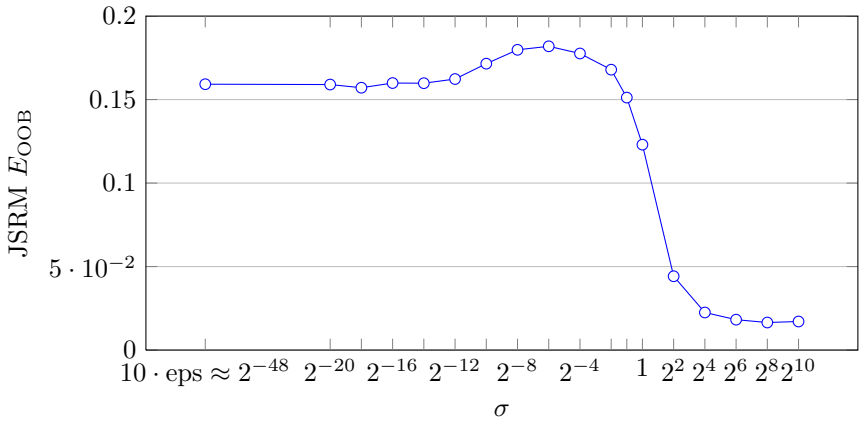

Figure 5.3.4: Detection error E_{OOB} obtained using the merger of JRM and SRMQ1 (JSRM) features as a function σ for J-UNIWARD with payload $\alpha = 0.4$ bpnzAC and JPEG quality factor 75.

	CSR		SRM	
	$\sigma = 10 \cdot eps$	$\sigma = 1$	$\sigma = 10 \cdot eps$	$\sigma = 1$
Haar	0.0649	0.3302	0.0339	0.0707
Daubechies 2	0.0278	0.4299	0.1313	0.1744
Daubechies 4	0.0106	0.4279	0.1763	0.1966
Daubechies 8	0.0203	0.4518	0.2001	0.1981
Daubechies 20	0.1934	0.4646	0.2046	0.1868
Symlet 8	0.0235	0.4410	0.1635	0.1919
Coiflet 1	0.0458	0.4426	0.0796	0.1444
Biorthogonal 44	0.0264	0.4388	0.0859	0.1683
Biorthogonal 68	0.0376	0.4459	0.1259	0.1820

Table 5.2: Detection error E_{OOB} obtained using CSR and the SRM features when using different filter banks in UNIWARD for $\sigma = 10 \cdot \text{eps}$ and $\sigma = 1$.

We note that constructing a similar targeted attack against JPEG implementations of UNIWARD is likely not feasible because the distortion caused by a change in a DCT coefficient affects a block of 8×8 pixels and, consequently, 23×23 wavelet coefficients. The distortion "averages out" and no banding artefacts show up in the embedding probability map. Steganalysis of J-UNIWARD with JSRM shown in Figure 5.3.4 indicates that the optimal σ for J-UNIWARD is 2^{-6}, which we selected for all experiments with J-UNIWARD and SI-UNIWARD in this chapter.

5.3.2 Effect of the filter bank

As a final experiment of this section aimed at finding the best settings of UNIWARD, we studied the influence of the directional filter bank. We did so for a fixed payload $\alpha = 0.4$ bpp and two values of σ when steganalyzing using the CSR and SRM features. Table 5.1 shows the results for five different wavelet bases[13] with varying parameters (support size s). The best results have been achieved with the 8-tap Daubechies wavelet, whose 1D low and high-pass filters are displayed in Table 5.1.

[13] http://wavelets.pybytes.com/wavelet/db8/

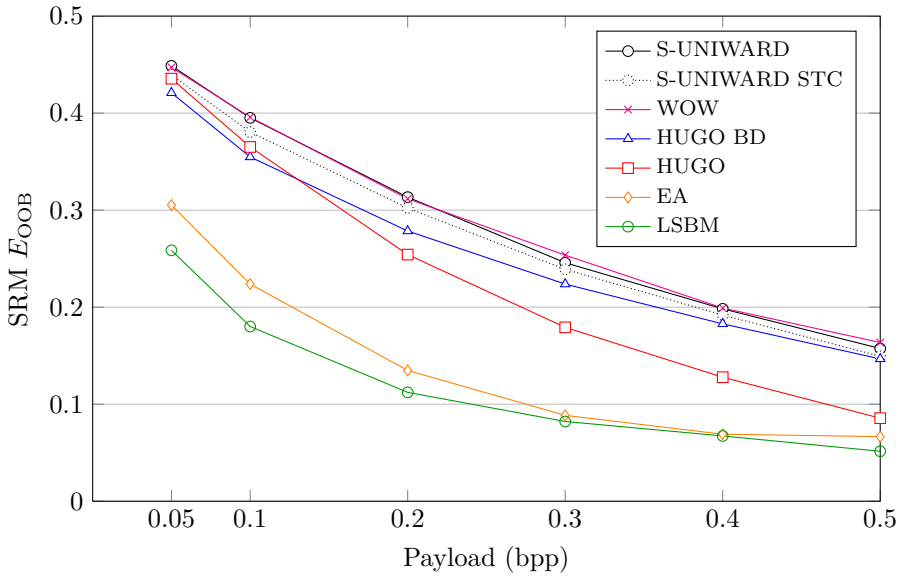

Figure 5.4.1: Detection error E_{OOB} using SRM as a function of relative payload for S-UNIWARD and five other spatial-domain steganographic schemes.

5.4 Experiments

In this section, we test the steganography using UNIWARD implemented with the 8-tap Daubechies directional filter bank and $\sigma = 1$ for S-UNIWARD and $\sigma = 2^{-6}$ for J- and SI-UNIWARD. We report the results on a range of relative payloads 0.05, 0.1, 0.2, ..., 0.5 bits per pixel (bpp), while JPEG-domain (and side-informed JPEG) methods will be tested on the same payloads expressed in bits per non-zero cover AC DCT coefficient (bpnzAC).

5.4.1 Spatial domain

In the spatial domain, we compare the proposed method with HUGO [90], HUGO implemented using the Gibbs construction with bounding distortion (HUGO BD) [33], WOW [54], LSB Matching (LSBM), and the Edge Adaptive (EA) algorithm [82]. With the exception of the EA algorithm, in which the costs and the embedding algorithm are inseparable, the results of all other algorithms are reported for embedding simulators that operate at the theoretical payload–distortion bound. The only algorithm that we implemented using STCs (with constraint height $h = 12$) to assess the coding loss is the proposed S-UNIWARD method.

For HUGO, we used the embedding simulator [36] with default settings $\gamma = 1$, $\sigma = 1$, and the switch --T with $T = 255$ to remove the weakness reported in [76]. HUGO BD starts with a distortion measure implemented as a weighted norm in the SPAM feature space, which is non-additive and not locally supported either. The bounding distortion is a method (see Section VII in [33]) to give the distortion the form needed for the Gibbs construction to work – the local supportedness. HUGO BD was implemented using the Gibbs construction with two sweeps as described in the original publication with the same parameter settings as for HUGO. The non-adaptive LSBM was simulated at the ternary bound corresponding to uniform costs, $\rho_{ij} = 1$ for all i, j.

Figure 5.4.1 shows the E_{OOB} error for all stego methods as a function of the relative payload expressed in bpp. While the security of the S-UNIWARD and WOW is practically the same due to the similarity of their distortion functions, the improvement over both versions of HUGO is

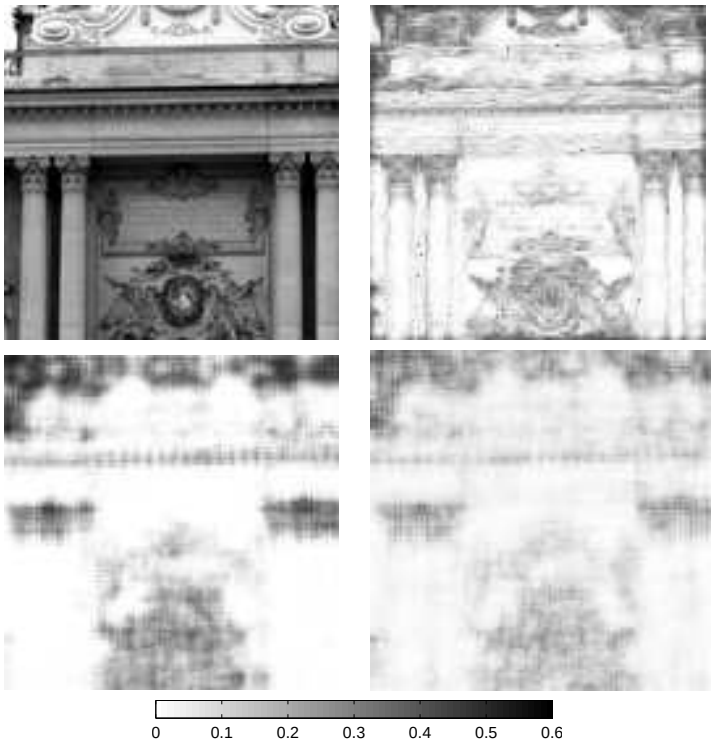

Figure 5.4.2: Embedding probability for payload 0.4 bpp using HUGO (top right), WOW (bottom left), and S-UNIWARD (bottom right) for a 128×128 grayscale cover image (top left).

quite apparent. HUGO BD performs better than HUGO especially for large payloads, where its detectability becomes comparable to that of S-UNIWARD. As expected, the non-adaptive LSBM performs poorly across all payloads, while EA appears only marginally better than LSBM.

In Figure 5.4.2, we contrast the probability of embedding changes for HUGO, WOW, and S-UNIWARD. The selected cover image has numerous horizontal and vertical edges and also some textured areas. Note that while HUGO embeds with high probability into the pillar edges as well as the horizontal lines above the pillars, S-UNIWARD directional costs force the changes solely into the textured areas. The placement of embedding changes for WOW and S-UNIWARD is quite similar, which is correspondingly reflected in their similar empirical security.

5.4.2 JPEG domain (non-side informed)

For the JPEG domain without side-information, we compare J-UNIWARD with nsF5 [47] and the recently proposed UED algorithm [51]. Since the costs used in UED are independent of the embedding change direction, we decided to include for comparison the UED implemented using *ternary* codes rather than binary, which indeed produced a more secure embedding algorithm.[14] All methods were again simulated at their corresponding payload–distortion bounds. The costs for nsF5 were uniform over all non-zero DCTs with zeros as the wet elements [42]. Figure 5.4.3 shows the results for JPEG quality factors 75, 85, and 95. As in the spatial domain, J-UNIWARD clearly outperformed both nsF5 and both versions of UED by a sizeable margin across all three quality factors. Furthermore, when using STCs with constraint height $h = 12$, the coding loss appears rather small.

[14] The authors of UED were apparently unaware of this possibility to further boost the security of their algorithm.

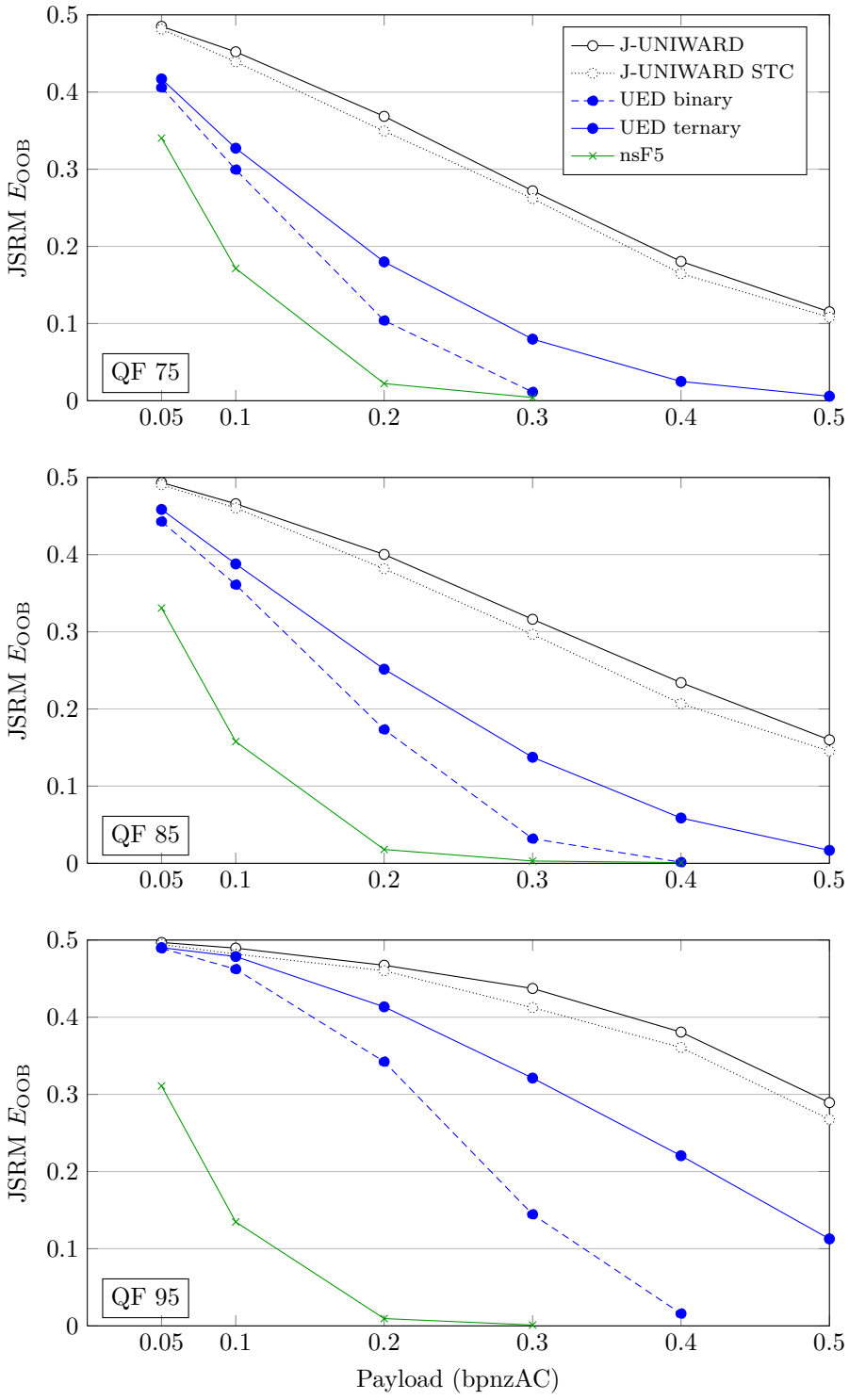

Figure 5.4.3: Testing error E_{OOB} for J-UNIWARD, nsF5, and binary (ternary) UED on BOSSbase 1.01 with the union of SRMQ1 and JRM and ensemble classifier for quality factors 75, 85, and 95.

5.4.3 JPEG domain (side-informed)

Working with the same three quality factors, we compare SI-UNIWARD with four other methods – the block entropy-weighted method of [107] (EBS), the NPQ [60], BCHopt [95], and the fourth method, which can be viewed as a modification (or simplification) of [95] or as [107] in which the normalization by block entropy has been removed. Following is a list of cost assignments for these four embedding methods; $\rho_{ij}^{(kl)}$ is the cost of changing DCT coefficient ij corresponding to DCT mode kl.

1. $\rho_{ij}^{(kl)} = \left(\frac{q_{kl}(0.5-|e_{ij}|)}{H(\mathbf{X}^{(b)})}\right)^2$
2. $\rho_{ij}^{(kl)} = \frac{q_{kl}^{\lambda_1}(1-2|e_{ij}|)}{(\mu+|X_{ij}|)^{\lambda_2}}$
3. $\rho_{ij}^{(kl)}$ as defined in [95]
4. $\rho_{ij}^{(kl)} = (q_{kl}(1-2|e_{ij}|))^2$

In Method 1 (EBS), $H(\mathbf{X}^{(b)})$ is the block entropy defined as $H(\mathbf{X}^{(b)}) = -\sum_i h_i^{(b)} \log h_i^{(b)}$, where $h_i^{(b)}$ is the normalized histogram of all non-zero DCT coefficients in block $\mathbf{X}^{(b)}$. Per the experiments in [60], we set $\mu = 0$ as NPQ embeds only in non-zero AC DCT coefficients, and $\lambda_1 = \lambda_2 = 1/2$ as this setting seemed to produce the most secure NPQ scheme for most payloads when tested with various feature sets. The cost ρ_{ij} for Methods 1–4 is equal to zero when $e_{ij} = 1/2$. Methods 1 and 4 embed into all DCT coefficients, including the DC term and coefficients that would otherwise round to zero ($X_{ij} = 0$). We remind from Subsection 5.2.3.1 that methods 1, 2, and 4 avoid embedding into 1/2-coefficients from DCT modes 00, 04, 40, and 44. Since the cost assignment in Method 3 (BCHopt) is inherently connected to its coding scheme, we kept this algorithm it unchanged in our tests.

Figure 5.4.4 shows that SI-UNIWARD achieves the best security among the tested methods for all payloads and all JPEG quality factors. The coding loss is also quite negligible. Curiously, the weighting by block entropy in the EBS method paid off only for quality factor 95. For factors 85 and 75, the weighting actually increases the statistical detectability using our feature vector (c.f., the "Square" and "EBS" curves). The dashed curves for quality factor 95 in Figure 5.4.4 are included to show the negative effect when 1/2-coefficients from DCT modes 00, 04, 40, and 44 are used for embedding (see the discussion in Section 5.2.3.1). In this case, the detection error levels off at approximately $25-30\%$ for small–medium payloads because most embedding changes are executed at the above four DCT modes. Note that NPQ and BCHopt do not exhibit the pathological error saturation as strongly because they do not embed into the DC term (mode 00).

5.5 Conclusion

Perfect security seems unachievable for empirical cover sources, examples of which are digital images. Currently, the best the steganographer can do for such sources is to minimize the detectability when embedding a required payload. A standard way to approach this problem is to embed while minimizing a carefully crafted distortion function, which is tied to empirical statistical detectability. This converts the problem of secure steganography to one that has been largely resolved in terms of known bounds and general near-optimal practical coding constructions.

The contribution of this section is a clean and universal design of the distortion function called UNIWARD, which is independent of the embedding domain. The distortion is always computed in the wavelet domain as a sum of relative changes of wavelet coefficients in the highest frequency

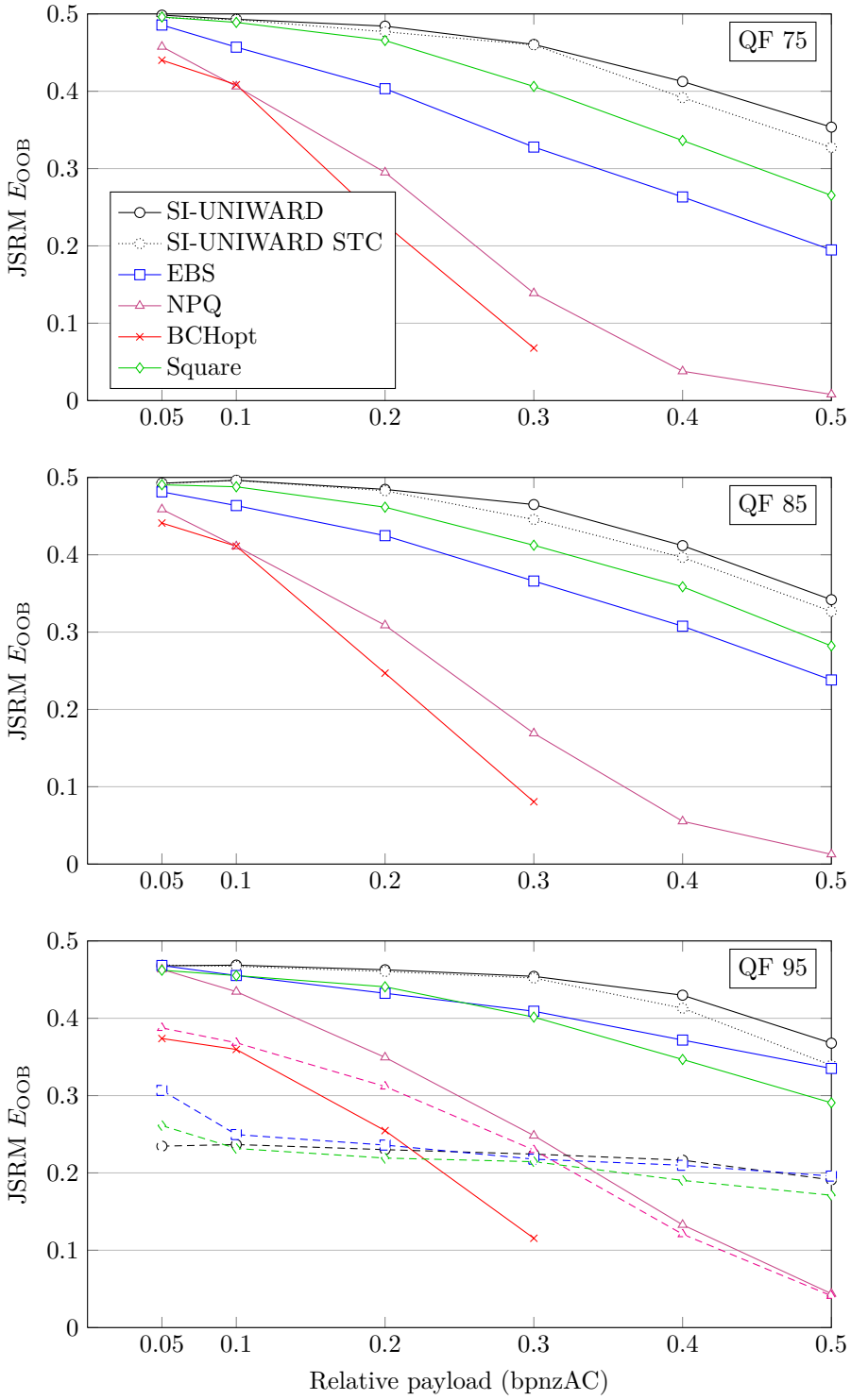

Figure 5.4.4: Detection error E_{OOB} for SI-UNIWARD and four other methods with the union of SRMQ1 and JRM and the ensemble classifier for JPEG quality factors 75, 85, and 95. The dashed lines in the graph for QF 95 correspond to the case when all the embedding methods use all coefficients, including the DCT modes 00 04 40 44 independently of the value of the rounding error e_{ij}.

undecimated subbands. The directionality of wavelet basis functions permits the sender to assess the neighborhood of each pixel for the presence of discontinuities in multiple directions (textures and "noisy" regions) and thus avoid making embedding changes in those parts of the image that can be modeled along at least one direction (clean edges and smooth regions). This model-free heuristic approach has been implemented in the spatial, JPEG, and side-informed JPEG domains. In all three domains, the proposed steganographic schemes matched or outperformed current state-of-the-art steganographic methods. A quite significant improvement was especially obtained for the JPEG and side-informed JPEG domains. As demonstrated by experiments, the innovative concept to assess the costs of changing a JPEG coefficient in an alternative domain seems to be quite promising.

Although all proposed methods were implemented and tested with an additive approximation of UNIWARD, this distortion function is naturally defined in its non-additive version, meaning that changes made to neighboring pixels (DCT coefficients) interact in the sense that the total imposed distortion is not a sum of distortions of individual changes. This potentially allows UNIWARD to embed while taking into account the interaction among the changed image elements. We explore this direction in the next chapter.

Last but not least, we have discovered a new phenomenon that hampers the performance of side-informed JPEG steganography that computes embedding costs based solely on the quantization error of DCT coefficients. When unquantized DCT coefficients that lie exactly in the middle of the quantization intervals are assigned zero costs, any embedding that minimizes distortion starts introducing embedding artifacts that are quite detectable using the JPEG rich model. While the makeshift solution proposed in this article is by no means optimal, it raises an important open question, which is how to best utilize the side information in the form of an uncompressed image when embedding data into the JPEG compressed form.

Chapter 6

Embedding using non-additive distortion

The distortion functions in their most general form (5.2.2) and (5.2.4) proposed in Chapter 5 are naturally non-additive – they allow embedding changes to interact. The Gibbs construction [33] is a general framework capable of embedding with non-additive distortion. It requires that the distortion be a sum of locally-supported potential functions, which is satisfied in our case (see the discussion in Section 5.2.2). Since Gibbs construction is a rather complex topic, we avoid repeating all the details in this dissertation and we only point out the most relevant facts and refer to the original publication. This chapter is organized as follows. A brief summary of the Gibbs construction can be found in Section 6.1. During our research an issue appeared with the Gibbs construction approach. The issue is explaned in Section 6.2 and it was not forseen by the authors of the original paper because of the distortion function they used for the experiments, which is described in Section 6.3. Finally, an interesting property of embedding changes by the non-additive S-UNIWARD embedding changes is shown in Section 6.4.

6.1 Gibbs construction summary

As mentioned in Section 3.3 in detail, an embedding scheme operating on the payload–distortion bound changes $\mathbf{X}$ to $\mathbf{Y} \in \mathcal{Y}$, where $\mathcal{Y}$ is the set of all possible stego images obtainable from $\mathbf{X}$, with probability $\pi_{\mathbf{X}}(\mathbf{Y}) = Z^{-1}\exp(-\lambda D(\mathbf{X},\mathbf{Y}))$, where $Z(\mathbf{X}) = \sum_{\mathbf{Y}\in\mathcal{Y}}\exp(-\lambda D(\mathbf{X},\mathbf{Y}))$ is the partition function (normalizing factor) of the Gibbs distribution for cover $\mathbf{X}$. The parameter $\lambda > 0$ is determined from the payload constraint, $m = H(\pi_{\mathbf{X}})$, if one wishes to embed m bits (the so-called payload-limited sender), where $H(\pi_{\mathbf{X}})$ is the entropy of $\pi_{\mathbf{X}}$.

The key observation in the Gibbs construction is the fact that $D(\mathbf{X},\mathbf{Y})$ is additive on a sublattice of pixels/DCT coefficients that are separated by more than the support width of the potential functions. For the spatial domain, since changing pixel ij affects the wavelet coefficients $u, v \in \{i-7,\ldots,i+8\}\times\{j-7,\ldots,j+8\}$, one can decompose the pixels into L^2 ($L = 16$) regular square sublattices, whose pixels do not interact:

$$\mathcal{L}_{ab} \triangleq \{(i,j) \in \{1,\ldots,n_1\}\times\{1,\ldots,n_2\} | i = a + (L+1)i_a, j = b + (L+1)j_b\} \tag{6.1.1}$$

, $a, b \in \{1,\ldots,L\}$, i_a, j_b non-negative integers. The embedding changes on each $\mathcal{L}_{ab}$ can thus be executed *independently* with embedding change probabilities given by the local characteristics (conditional probabilities) $\Pr(X_{ij} = Y_{ij}|\mathbf{Y}_{\sim ij})$, which allows utilizing the STCs as shown in Ref. [33].

Thus, given a payload of m bits, one can embed m/L^2 bits in each sublattice using STCs. After embedding into all sublattices, we have embedded the entire payload and also technically completed

one *sweep* of a Gibbs sampler [111]. Starting with the cover image and iterating this process N times, obtaining the stego image $\mathbf{Y}^{(N)}$ after the Nth sweep, asymptotically (for large N) $\mathbf{Y}^{(N)}$ will be selected with the correct probability $\pi_{\mathbf{X}}(\mathbf{Y})$. The number of iterations N needed depends on the distortion function. In practice, we terminate the sweeps once the embedding distortion $D(\mathbf{X}, \mathbf{Y}^{(N)})$ as a function of N saturates.

6.2 Issue with non-additive S-UNIWARD distortion

When the Gibbs construction was applied to S-UNIWARD with the distortion (5.2.2), with each sweep the distortion $D(\mathbf{X}, \mathbf{Y}^{(N)})$ increased and eventually saturated (see Figure 6.2.1). Since the correct message has already been embedded after the first sweep, paradoxically, adding more sweeps only increased the distortion, and we experimentally confirmed that the resulting stego images were correspondingly more detectable.

The failure of the Gibbs construction to embed a given payload with minimal distortion was traced to the following problem. The embedding by sublattices embeds in each sweep the so-called *erasure* entropy (Section VI.C in Ref. [33]):

$$H^{-} = \sum_{a,b=1}^{L} H(\mathbf{Y}_{[ab]}|\mathbf{Y}_{\sim[ab]}),$$

where we denoted with $\mathbf{Y}_{[ab]}$ the stego image $\mathbf{Y}$ restricted to $\mathcal{L}_{ab}$ and $\mathbf{Y}^{(N)}_{\sim[ab]}$ the union of all remaining sublattices of $\mathbf{Y}$. In general, $H^{-} \leq H(\pi_{\mathbf{X}})$ with the equality holding when all $\mathbf{Y}_{[ab]}$ are independent. The stronger the interactions among the embedding changes are, the larger the difference between both entropies becomes. Rephrased, when adding more sweeps of the Gibbs sampler, we end up with a distortion that corresponds to a higher entropy – the entropy of the Markov random field $H(\pi_{\mathbf{X}})$, but we only embedded the payload of H^{-} bits. It seems that the only way to overcome this problem is to replace the embedding by STCs on sublattices with a different algorithm, which inevitably calls for a novel coding scheme.

To verify the above considerations, we selected the standard grayscale Lenna image[1] and computed its payload–distortion bound using the method of stochastic integration (Section V.B in Ref. [33]). This bound renders the entropy per pixel (relative payload) $H(\pi_{\mathbf{X}})/(n_1 n_2)$ as a function of the minimal expected distortion per pixel needed to embed this payload, $E_{\pi_{\mathbf{X}}}[D(\mathbf{X}, \mathbf{Y})]/(n_1 n_2) = 1/(n_1 n_2) \sum_{\mathbf{Y} \in \mathcal{Y}} \pi_{\mathbf{X}}(\mathbf{Y}) D(\mathbf{X}, \mathbf{Y})$. The bound is shown in Figure 6.2.1 on the left as a solid black line. The parameter σ in UNIWARD distortion function is set to $\sigma \approx 2 \cdot 10^{-15}$ in order to increase the strength of interactions among pixels to make the problem more visible. The acronym 'AA' Stands for additive approximation in all the plots in this chapter. The red solid line is the payload–distortion relationship achieved using the additive approximation $D_{\mathrm{A}}(\mathbf{X}, \mathbf{Y})$, while the remaining blue lines render the results after $N = 1, 2, \ldots, 20$ sweeps of the Gibbs sampler. The fact that the curve corresponding to the additive approximation lies below the payload–distortion bound for the Markov field $\pi_{\mathbf{X}}$ testifies that, indeed, for a fixed distortion $D(\mathbf{X}, \mathbf{Y})$, the additive approximation embeds a lower payload than the entropy $H(\pi_{\mathbf{X}})$. This difference also shows how much one could increase the secure payload if one was able to embed the true entropy of the Markov field rather than the erasure entropy.

Figure 6.2.1 demonstrates one more curious fact. The first sweep of the Gibbs sampler provides a better payload–distortion relationship than the additive approximation. Yet, stego images embedded using one sweep of the Gibbs sampler exhibited a higher statistical detectability than images embedded using the additive approximation. In other words, the distortion and statistical detectability (as evaluated *empirically* on a given cover source, using a specific feature space, and a classifier) do not correspond. The reason for this could be a) the interactions among adjacent pixels are not

[1] Lenna image can be downloaded from http://en.wikipedia.org/wiki/Lenna.

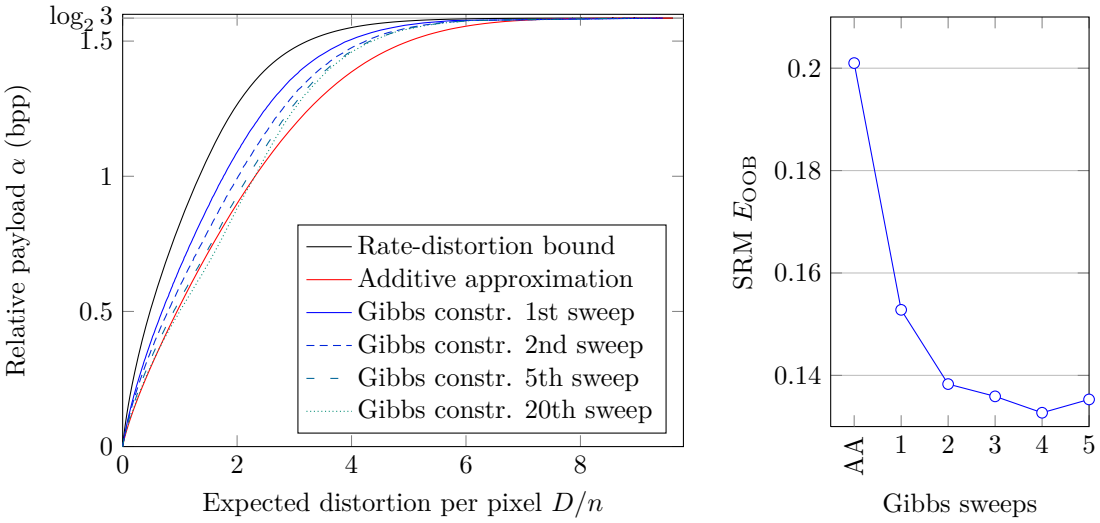

Figure 6.2.1: Payload–distortion bound for S-UNIWARD and the payload–distortion relations for its additive approximation and Gibbs sweeps for the standard 512×512 grayscale Lenna image (left). Classification error using SRM features for different number of Gibbs seeps (right) for payload 0.4 bpp. Both plots are computed using S-UNIWARD with parameter $\sigma \approx 2 \cdot 10^{-15}$ to stress the desired property.

well captured by the non-additive distortion $D(\mathbf{X}, \mathbf{Y})$, b) the distortion function does correspond to statistical detectability but our empirical classifiers provide a skewed perspective and for some reason better detect steganography by iterative embedding on sublattices than embedding using the additive approximation, which can be viewed as embedding on a single sublattice. In other words, assuming the case b), the results might come out differently with a better steganalyzer.

6.3 Non-additive HUGO BD

In the original paper [33], the authors did not run into the issue mentioned in the previous section most likely because the interactions among changed pixels (the non-additivity of the distortion) were much weaker, which kept the difference between H^- and $H(\pi_{\mathbf{X}})$ small. This conjecture seems to be supported by the fact that only two sweeps of the Gibbs sampler were needed for convergence.

Figure 6.3.1 shows the payload–distortion relationship for the bounding distortion of HUGO BD. For each expected distortion per pixel, D/n, the black line bound is the payload that one could theoretically embed using the weighted norm in the SPAM [88, 90] feature space – the Markov random field entropy $H(\pi_{\mathbf{X}})$. The red line corresponds to the additive approximation of this function, while the blue lines correspond to the Gibbs sweeps for the bounding distortion initiated at the stego image obtained by the additive approximation. The sweeps decrease the embedding distortion for each payload – or one can say that the payload one can embed for a fixed distortion increases and gets relatively close to the payload-distortion bound. This behavior is in contrast with what was observed in the previous section and it is caused by the much weaker interactions among changed pixels.

This fact leads to a rather interesting conclusion. The gibbs construction was proposed for embedding using distortion functions with interaction among embedding changes. When there are no interactions, this distortion function is additive and STCs can be used directly. However, if those interactions are too strong, the Gibbs construction increases the total embedding distortion instead of decreasing it. Both cases render the Gibbs construction for non-additive distortion functions unusable.

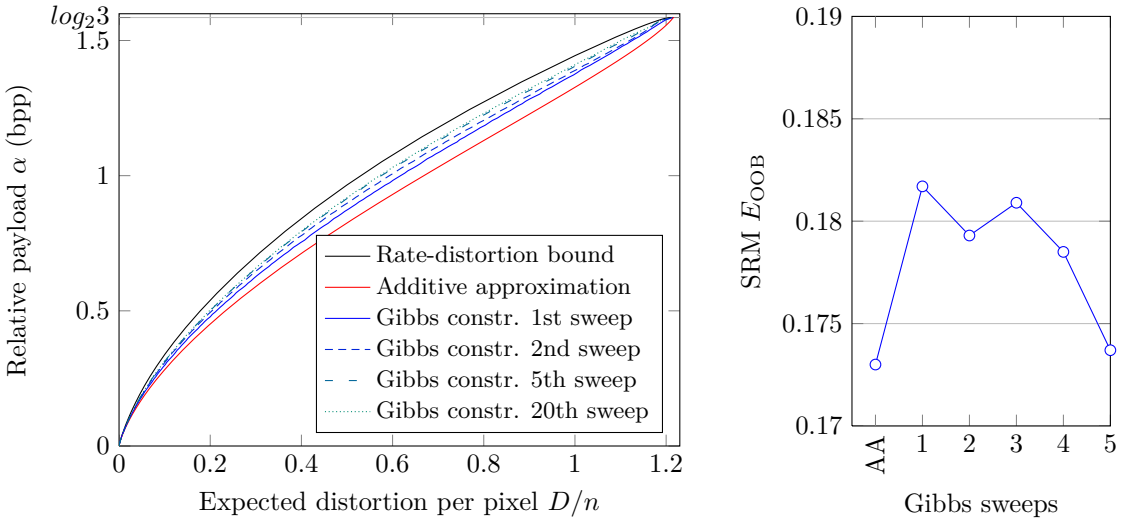

Figure 6.3.1: Rate–distortion bound for HUGO BD, the payload–distortion after Gibbs sweeps, and the additive approximation for the standard 512×512 grayscale Lenna image (left). Classification error using SRM features for different number of Gibbs seeps (right) for payload 0.4 bpp.

6.4 Change rate and non-additive S-UNIWARD distortion

Since the steganographic approach using embedding while minimizing a distortion function was introduced, measuring the number of embedding changes became meaningless. In this short section, we will bring the number embedding changes and their location back into the picture.

The BOSSbase database is again embedded with S-UNIWARD using the Gibbs construction and a payload 0.4 bits per pixel – this time the parameter σ was set to the default value $\sigma = 1$. Intuitively, much higher number of embedding changes should cause a decrease in the steganographic security. However, this is not the case here. Figure 6.4.1 shows the relationship between detectability, change rate, and the number of Gibbs sweeps. The average change rate for this embedding using additive approximation is 0.074, meaning that 7.4 % of all pixels were changed either by $+1$ or -1. This translates to the detection error of 0.2 using the SRM [43] feature set. Surprisingly, after 15 sweeps of the Gibbs construction are applied while communicating an identical message, the change rate increased three times to 0.21 but the detection error remained almost identical at 0.198.

Let us take a closer look at the embedding changes shown in Figure 6.4.2. The difference in the number of embedding changes for the additive approximation and 15 Gibbs sweeps can be seen immediately. However, what makes these approaches equally detectable is the the way Gibbs construction groups these changes. Instead of making ± 1 modifications with the same probability, the sign of the modification strongly correlates among the neighboring pixels, consequently creating larger patches of embedding changes with the same sign.

Making embedding changes in larger patches makes a lot of sense with respect to the properties of modern steganalytic features. Every modern feature set first extracts image residuals (e. g. (4.2.1)), thus considering only differences among pixels while completely ignoring their magnitude. When a large patch is modified by adding $+1$, only the boundary of the patch will have an effect on the extracted image statistics, making it less detectable than what one would expect by counting embedding changes.

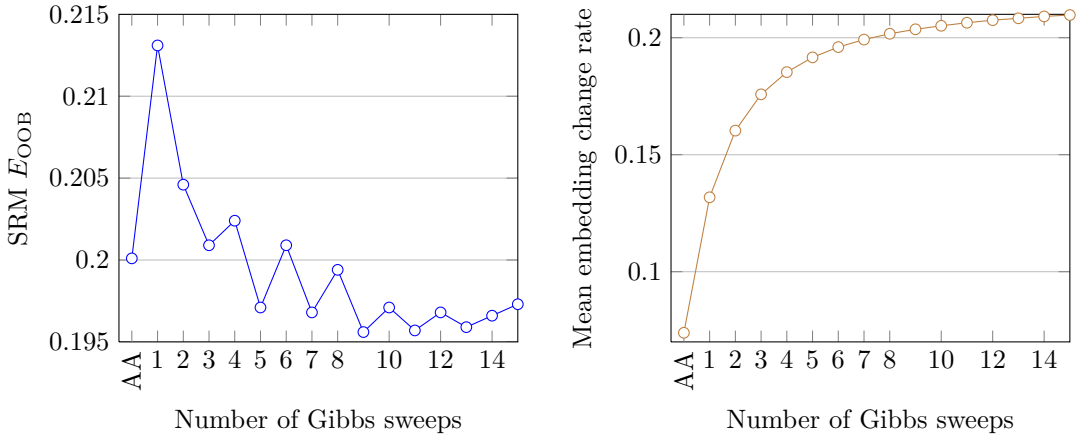

Figure 6.4.1: Classification error using SRM features (left) and average change rate for payload 0.4 bpp over the whole BOSSbase database (right) for different number of Gibbs sweeps.

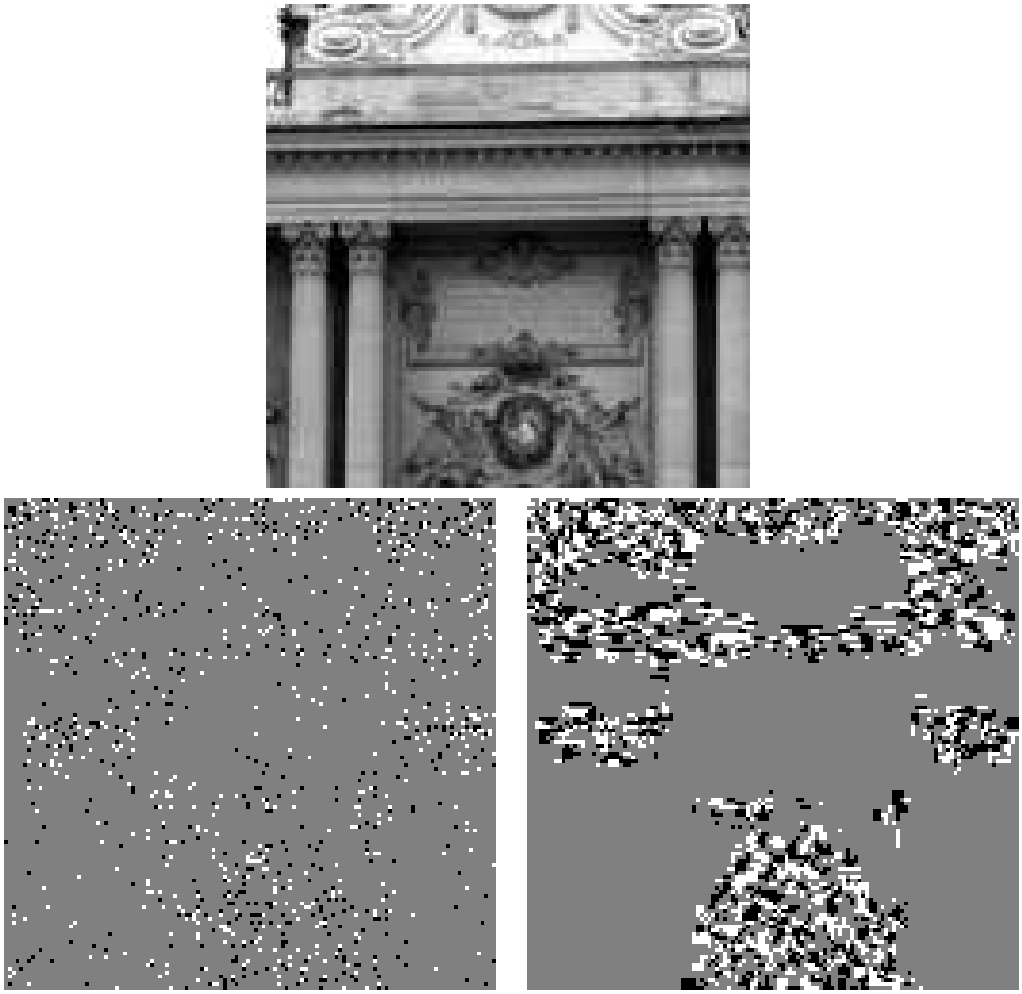

Figure 6.4.2: 128×128 pixel cover image (top), location of embedding changes for additive approximation (bottom left) and 15 sweeps (bottom right). Modifications by $+1$ are marked white, modifications by -1 are marked black.

Chapter 7

Challenging the doctrines of JPEG steganography

This chapter is an taken from author's SPIE 2014 conference paper [58]. This paper is included in this chapter almost unchanged for chronological reasons even though some of the raised questions are answered in Chapter 8.

The design of both steganography and steganalysis methods for digital images heavily relies on empirically justified principles. In steganography, the domain in which the embedding changes are executed is usually the preferred domain in which to measure the statistical impact of embedding (to construct the distortion function). Another principle almost exclusively used in steganalysis states that the most accurate detection is obtained when extracting the steganalysis features from the embedding domain.

While a substantial body of prior art seems to support these two doctrines, this chapter challenges both principles when applied to the JPEG format. Through a series of targeted experiments on numerous older as well as current steganographic algorithms, we lay out arguments for why measuring the embedding distortion in the spatial domain can be highly beneficial for JPEG steganography. Moreover, as modern embedding algorithms avoid introducing easily detectable artifacts in the statistics of quantized DCT coefficients, we demonstrate that more accurate detection is obtained when constructing the steganalysis features in the spatial domain where the distortion function is minimized, challenging thus both established doctrines.

7.1 Introduction

It is an obvious fact that if the sender executes the embedding changes uniformly pseudo-randomly across the cover image, a scheme that on average introduces the fewest number of embedding changes ought to be more secure than its competitors. This reasoning provided a bridge between the theory of covering codes and steganography [24, 6, 48] responsible for an avalanche of papers on matrix embedding and a suite of more secure steganographic algorithms, such as the F5 algorithm [108] and its improved version called nsF5 [47].

Measuring the embedding distortion by counting the embedding changes, however, fails to take into account the fact that modifications of quantized DCT coefficients from the same 8×8 block strongly interact and that the embedding changes may have different "costs" depending on the associated quantization step and the local image content. Moreover, DCT coefficients that are adjacent either in the frequency or spatial domain exhibit complex dependencies that are not well understood. While discernible objects and their orientation are easily identifiable in the spatial domain, it is harder to

determine them by inspecting DCT coefficients. From this perspective, it appears that it might be advantageous to abandon the doctrine that requires measuring distortion in the embedding domain as it is more manageable to design distortion functions that correlate with statistical detectability in the spatial domain. This thesis seems to be in agreement with recent developments in steganography that we discuss below.

The authors of BCHopt [95] were the first to recognize that a good distortion measure needs to consider the effect of the quantization step associated with the modified coefficients. Barring some unimportant details, the distortion function was basically designed to minimize the embedding distortion w.r.t. the uncompressed cover image (the precover). The minimized quantity was the square of the product of the quantization step and the change in the DCT coefficient w.r.t. the precover. Such an embedding distortion, however, could equivalently be defined as an L_2 norm in the spatial domain due to Parseval equality because the DCT is orthonormal. The more recent Entropy Block Steganography (EBS) [107] improved significantly upon BCHopt using a similar distortion function by replacing the BCH codes with the much more powerful Syndrome–Trellis Codes (STCs) [35].

Viewing both algorithms from the perspective of the current state of the art, both BCHopt and EBS hinted at a trend to embed in JPEG images by minimizing an embedding distortion defined in the spatial domain. This development culminated in the design of the recently proposed UNIWARD distortion function (see Chapter 5 or Ref. [59]), which provides a universal method for measuring the embedding distortion independently of where the embedding changes are executed. Schemes based on UNIWARD were shown to significantly outperform prior art for steganography in JPEG images (both with and without side information at the sender). In UNIWARD, the distortion is computed as a sum of relative changes of directional residuals obtained using a Daubechies 8-tap filter bank. As shown later in this chapter using experiments with the JPEG rich model [74], minimizing a spatial-domain-based UNIWARD seems to minimize the impact on the statistics of DCT coefficients as well. UNIWARD also naturally incorporates the effect of the quantization step that other schemes need to build in, usually in some ad hoc manner (see, e.g., NPQ [60] and its improved version [29]).

We now take a closer look at the opposite problem, which is the detection of steganography (steganalysis). A doctrine has been formulated in 2004 (Ref. [37]) claiming that the most accurate steganalysis will naturally be achieved in the embedding domain because this is where the embedding changes are lumped and isolated. This doctrine seemed to hold true for embedding algorithms available at that time. This was mostly due to the fact that the early JPEG-domain stego algorithms, e.g., Jsteg [105], F5, and OutGuess [93], introduced quite detectable artifacts into the distribution of DCT coefficients (both their first-order and higher-order statistics). Furthermore, this doctrine was engraved even deeper in the minds of researchers after the BOSS competition [4] when all successful participants used steganalysis features constructed in the spatial (embedding) domain.

The fact that features computed in other domains can be useful for steganalysis is not new and it appeared already in the first papers on feature-based blind steganalysis [30] as well as in Ref. [37] (the "blockiness" feature is defined in the spatial domain). For a long time it remained true, though, that features constructed in the embedding domain provided the most accurate steganalysis results. The authors of Ref. [71] proposed the so-called Cross-Domain Features (CDFs) to improve the attack on YASS [102]. This was not surprising as YASS embeds in a key-dependent domain and thus one cannot construct features in the embedding domain. With the development of rich image models for both the spatial (SRM) [43] and DCT (JRM) [74] domains it was shown in Ref. [74] that virtually all JPEG-domain algorithms can be detected more reliably with the union of the SRM and JRM called JSRM. The size of the improvement was dependent on the algorithm and was generally larger for those embedding algorithms that were harder to detect, which were exactly those that somehow utilized the spatial domain representation in computing their distortion function. Using selected experiments, we demonstrate in this chapter that the current most advanced JPEG-domain stego algorithms are better detected in the domain in which the distortion is minimized rather than the domain where the embedding changes are executed.

In the next section, we introduce the common core of all experiments and briefly describe the

steganalysis features and steganographic algorithms utilized in experiments. In Section 7.2, we introduce the results of all experiments and their interpretation that challenges both doctrines discussed above. Section 7.3 contains a brief summary.

Even though parts of this work have appeared in a scattered form in other papers, the authors believe that clearly spelling out the main message (the challenge of both doctrines) in a stand-alone chapter supported with dedicated experiments is valuable for the steganographic community.

With the exception of BCHopt, all side-informed embedding algorithms avoid making embedding changes to DCT coefficients with rounding error $e_{ij} = 1/2$ in DCT modes $(k, l) \in \{(0, 0), (0, 4), (4, 0), (4, 4)\}$ to avoid a singular behavior for small payloads that is especially apparent for large quality factors (see Subsection 5.2.3.1 for details).

7.2 Experiments

In this section, we interpret the results of experiments shown in Table 7.1. By doing so, we challenge the doctrines mentioned in the introduction. The table shows the E_{OOB} detection error obtained using the JRM, the spatial domain PSRMQ3, and the combined JPSRM on the JPEG steganographic algorithms listed in Section 2.3.2. All experiments in this chapter were run on the standard database BOSSbase 1.01. The results are presented for two quality factors and one small and one large relative payload expressed in bits per non-zero AC DCT coefficient (bpnzAC). Since the coding in BCHopt does not allow embedding 0.4 bpnzAC in all images, we tested it for 0.3 bpnzAC.

Figure 7.2.1 displays the same results in a graphical form for the quality factor 75. In the figure, the algorithms are ordered by their statistical detectability obtained using the JPSRM. To give the reader a sense of the statistical significance of small changes in the E_{OOB}, we measured this error over ten runs of the ensemble classifier with different seeds for its random number generator that drives the selection of random subspaces as well as the bootstrapping for the training sets. The standard deviation of E_{OOB} was rather stable across the payloads, quality factors, as well as embedding algorithms, and it was always below 0.003. For better readability, we refrain from including this spread in the table.

When the JRM can detect a stego algorithm efficiently, one can say that the embedding disturbs important statistics of DCT coefficients. We view such algorithms as "faulty." Depending on the stego algorithm, the problem is either in the embedding operation or in the design of the distortion function that is supposed to measure the statistical detectability of embedding changes. Both the LSB replacement embedding operation of Jsteg and the operation of nsF5, which always decreases the absolute value of the DCT coefficient, predictably modify the first-order (and higher-order) statistics of coefficients. Such artifacts are understandably better detected by the JRM than the PSRM. The same is true for OutGuess, which turned out as the most detectable out of all tested algorithms. Even though it preserves the global histogram, it does so at the expense of introducing additional changes, and, in the end, disturbs the statistics of DCT coefficients even more. (Recall, that the JRM uses statistics of individual pairs of DCT modes, which are not necessarily preserved by OutGuess.)

While the ternary coded UED algorithm is markedly better than the older non side-informed algorithms, it is clearly outperformed by J-UNIWARD, which minimizes a distortion function defined in the spatial domain. This experimental fact challenges the first doctrine from Section 7.1 that claims that one should always minimize distortion defined in the embedding domain. The distortion function of J-UNIWARD seems to capture the impact on the statistics of DCT coefficients rather well. This finding should be taken "with a grain of salt" as it is entirely possible that better, more sophisticated distortion functions can be built in the DCT domain. The authors, however, believe that designing such functions will be rather challenging for the reasons mentioned in the introduction.

Two of the distortion-based side-informed steganographic schemes, BCHopt and NPQ, are also better detectable by the JRM than the PSRM. Their embedding operation is LSB matching, which

Payload	SI	0.1 bpnzAC			0.4 bpnzAC		
Features		JRM	PSRMQ3	JPSRM	JRM	PSRMQ3	JPSRM
Dimension		22,510	12,870	35,380	22,510	12,870	35,380
Quality factor 75							
OutGuess		0.0010	0.0011	0.0005	0.0001	0.0003	0.0001
Jsteg		0.0578	0.1159	0.0372	0.0004	0.0007	0.0003
nsF5		0.2115	0.2609	0.1631	0.0036	0.0057	0.0008
UED ternary		0.3968	0.3369	0.3393	0.0488	0.0390	0.0202
J-UNIWARD		0.4632	0.4319	0.4350	0.2376	0.1294	0.1228
BCHopt	•	0.4122	0.4228	0.3941	0.0830*	0.1039*	0.0546*
NPQ	•	0.4139	0.4613	0.4076	0.0654	0.0760	0.0345
Square loss	•	0.4908	0.4880	0.4914	0.3656	0.3246	0.3246
SI-UNIWARD	•	0.5004	0.4952	0.4970	0.4470	0.3744	0.3755
Quality factor 95							
OutGuess		0.0006	0.0015	0.0005	0.0001	0.0012	0.0002
Jsteg		0.0429	0.2033	0.0352	0.0001	0.0054	0.0003
nsF5		0.1354	0.3401	0.1220	0.0005	0.0252	0.0005
UED ternary		0.4750	0.4785	0.4727	0.2604	0.2759	0.2180
J-UNIWARD		0.4923	0.4943	0.4920	0.3951	0.3256	0.3246
BCHopt	•	0.3600	0.4715	0.3582	0.1172*	0.3491*	0.1144*
NPQ	•	0.4295	0.4950	0.4308	0.1471	0.3358	0.1342
Square loss	•	0.4556	0.4865	0.4554	0.3664	0.3952	0.3442
SI-UNIWARD	•	0.4654	0.4955	0.4672	0.4418	0.3909	0.3790

Table 7.1: Detection error E_{OOB} achieved using three different rich models for two JPEG quality factors and two payloads. The dot in the column labeled "SI" highlights those JPEG algorithms that use side information in the form of the uncompressed image. The asterisk highlights the fact that BCHopt was tested for payload 0.3 bpnzAC instead of 0.4 because its coding does not allow embedding payloads of this size in all images.

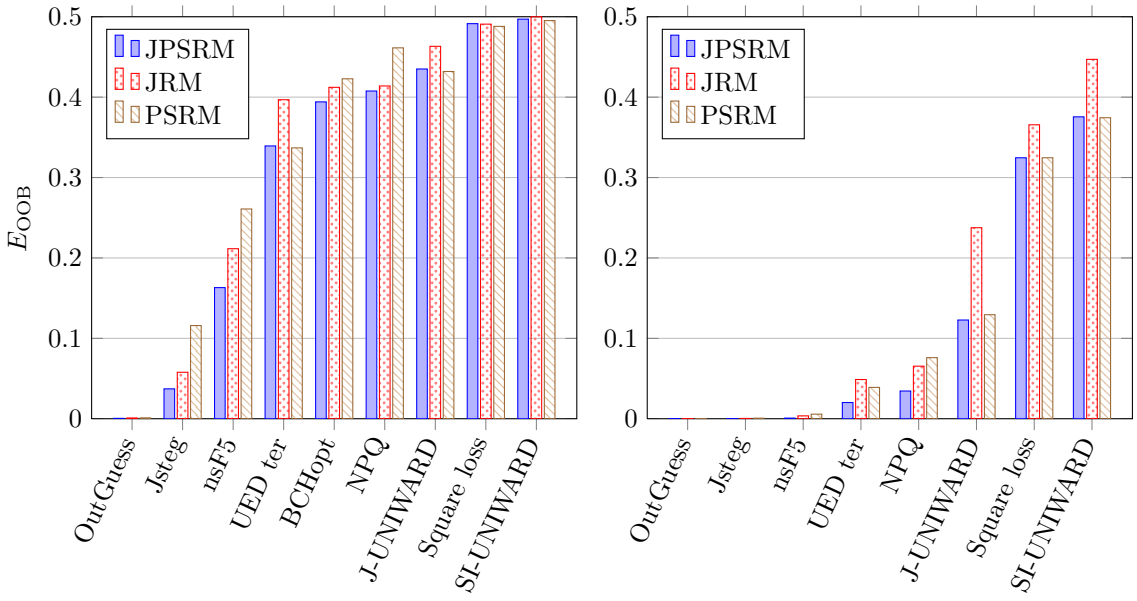

Figure 7.2.1: Detection error E_{OOB} using JPSRM, JRM, and PSRM on all tested steganographic algorithms for quality factor 75 with payloads 0.1 (left) and 0.4 (right) bits per non-zero AC coefficients. Note especially the cases when the spatial-domain features detect better than JPEG-domain features (when the brown bar is smaller than the red bar). Note that the merged JPSRM always provides the smallest detection error. This figure also nicely shows the progress made in JPEG steganography over the years.

introduces less strong artifacts in the statistics of coefficients than LSB replacement or the operation of nsF5. However, since all algorithms with the exception of nsF5, Jsteg, and OutGuess use LSB matching, the increased detectability of BCHopt and NPQ by JRM is most likely due to weaknesses in their distortion function, which does not capture the statistical dependencies among DCT coefficients well.

On the other hand, the most secure JPEG-domain algorithms, J-UNIWARD, and the side-informed Square Loss and SI-UNIWARD, are better detectable by the spatial-domain PSRM than by the JRM.[1] In fact, for the UNIWARD family the entire detection power seems to be coming from the PSRMQ3 as adding the JRM does not lead to any statistically significant improvement. This seems to point to two interesting facts. Reiterating and strengthening what has already been said about J-UNIWARD, since the distortion functions of the UNIWARD family are designed in the spatial domain, they naturally incorporate the effect of the quantization step and can better evaluate the impact of embedding on blockiness. What is more remarkable is that the schemes minimizing the impact in the spatial domain also seem to avoid introducing artifacts in the JPEG domain.

Moreover, with more sophisticated JPEG-domain algorithms that avoid disturbing the statistics of DCT coefficients it becomes more advantageous to steganalyze by representing the images in the domain in which the distortion is designed rather than in the embedding domain.

7.3 Conclusion

Throughout the history, researchers have converged to a few empirical principles widely used when designing both steganography and steganalysis algorithms. The two most prominent doctrines concern the role of the embedding domain as the preferred domain in which to measure the impact of

[1] For the small payload of 0.1 bpnzAC, they are essentially undetectable using any of the rich models.

embedding as well as extract steganalysis features. In this chapter, we provide experimental evidence that these doctrines may not be valid for embedding in JPEG images. This is mainly because the quantized DCT coefficients form 64 parallel channels that exhibit complex dependencies that are not easily quantified. On the contrary, in the spatial domain, elements that form typical objects, such as edges, segments, and textures, are easily identifiable, which allows for a simpler and more transparent design of distortion functions as well as extraction of good steganalysis features.

Experiments on older as well as modern steganographic algorithms for JPEG images point to several interesting findings:

1. Embedding algorithms that introduce easily identifiable artifacts in the statistics of DCT coefficients are better detected using features constructed in the embedding domain. This applies to older algorithms, such as OutGuess, nsF5, Jsteg, and Model-based steganography.
2. JPEG algorithms whose distortion function takes into account the impact of embedding in the spatial domain tend to exhibit higher security and avoid introducing artifacts that can be captured using the JPEG rich model.
3. Modern embedding algorithms that minimize the embedding impact computed in the spatial domain are generally better detected using the spatial rich model rather than the JPEG rich model.

These findings pose some intriguing open questions pertaining to both steganography design and detection. In particular, with modern and more secure steganographic algorithms, the domain of choice for steganalysis might shift from the embedding domain to the domain in which the distortion is minimized.

Chapter 8

Low Complexity Features for JPEG Steganalysis Using Undecimated DCT

This chapter introduces a novel feature set for steganalysis of JPEG images. The features are engineered as first-order statistics of quantized noise residuals obtained from the decompressed JPEG image using 64 kernels of the discrete cosine transform (the so-called undecimated DCT). This approach can be interpreted as a projection model in the JPEG domain, forming thus a counterpart to the projection spatial rich model. The most appealing aspect of this proposed steganalysis feature set is its low computational complexity, lower dimensionality in comparison to other rich models, and a competitive performance w.r.t. previously proposed JPEG domain steganalysis features.

8.1 Introduction

Steganalysis of JPEG images is an active and highly relevant research topic due to the ubiquitous presence of JPEG images on social networks, image sharing portals, and in Internet traffic in general. There exist numerous steganographic algorithms specifically designed for the JPEG domain. Such tools range from easy-to-use applications incorporating quite simplistic data hiding methods to advanced tools designed to avoid detection by a sophisticated adversary. According to the information provided by Wetstone Technologies, Inc, a company that keeps an up-to-date comprehensive list of all software applications capable of hiding data in electronic files, as of March 2014 a total of 349 applications that hide data in JPEG images were available for download.[1]

Historically, two different approaches to steganalysis have been developed. One can start by adopting a model for the statistical distribution of DCT coefficients in a JPEG file and design the detector using tools of statistical hypothesis testing [104, 113, 19]. In the second, much more common approach, a representation of the image (a feature) is identified that reacts sensitively to embedding but does not vary much due to image content. For some simple steganographic methods that introduce easily identifiable artifacts, such as Jsteg, it is often possible to identify a scalar feature – an estimate of the payload length [110, 112, 109, 7, 72]. More sophisticated embedding algorithms usually require higher-dimensional feature representation to obtain more accurate detection. In this case, the detector is typically built using machine learning through supervised training during which the classifier is presented with features of cover as well as stego images. Alternatively, the classifier can be trained that recognizes only cover images and marks all outliers as suspected stego

[1] Personal communication by Chet Hosmer, CEO of Wetstone Tech.

images [83, 92]. Recently, Ker and Pevný proposed to shift the focus from identifying stego images to identifying "guilty actors," e.g., Facebook users, using unsupervised clustering over actors in the feature space [66]. Irrespectively of the chosen detection philosophy, the most important component of the detectors is the feature space – their detection accuracy is directly tied to the ability of the features to capture the steganographic embedding changes.

Selected examples of popular feature sets proposed for detection of steganography in JPEG images are the historically first image quality metric features [2], first-order statistics of wavelet coefficients [31], Markov features formed by sample intra-block conditional probabilities [99], inter- and intra-block co-occurrences of DCT coefficients [17], the PEV feature vector [91], inter and intra-block co-occurrences calibrated by difference and ratio [80], and the JPEG Rich Model (JRM) [74]. Among the more general techniques that were identified as improving the detection performance is the calibration by difference and Cartesian calibration [80, 71]. By inspecting the literature on features for steganalysis, one can observe a general trend – the features' dimensionality is increasing, a phenomenon elicited by developments in steganography. More sophisticated steganographic schemes avoid introducing easily detectable artifacts and more information is needed to obtain better detection. To address the increased complexity of detector training, simpler machine learning tools were proposed that better scale w.r.t. feature dimensionality, such as the FLD-ensemble [77] or the perceptron [81]. Even with more efficient classifiers, however, the obstacle that may prevent practical deployment of high-dimensional features is the time needed to extract the feature [5, 57, 79, 63].

In this article, we propose a novel feature set for JPEG steganalysis, which enjoys low complexity, relatively small dimension, yet provides competitive detection performance across all tested JPEG stegoalgorithms. The features are built as histograms of residuals obtained using the basis patterns used in the DCT. The feature extraction thus requires computing mere 64 convolutions of the decompressed JPEG image with 64 8×8 kernels and forming histograms. The features can also be interpreted in the DCT domain, where their construction resembles the PSRM with non-random orthonormal projection vectors. Symmetries of these patterns are used to further compactify the features and make them better populated. The proposed features are called DCTR features (Discrete Cosine Transform Residual).

In the next section, we introduce the undecimated DCT, which is the first step in computing the DCTR features. Here, we explain the essential properties of the undecimated DCT and point out its relationship to calibration and other previous art. The complete description of the proposed DCTR feature set as well as experiments aimed at determining the free parameters appear in Section 8.4. In Section 8.5, we report the detection accuracy of the DCTR feature set on selected JPEG domain steganographic algorithms. The results are contrasted with the performance obtained using current state-of-the-art rich feature sets, including the JPEG Rich Model and the Projection Spatial Rich Model. The chapter is concluded in Section 8.6, where we discuss future directions.

8.2 Undecimated DCT

In this section, we describe the undecimated DCT and study its properties relevant for building the DCTR feature set in the next section. Since the vast majority of steganographic schemes embed data only in the luminance component, we limit the scope of this chapter to grayscale JPEG images. For easier exposition, we will also assume that the size of all images is a multiple of 8.

8.2.1 Description

Given an $M \times N$ grayscale image $\mathbf{X} \in \mathbb{R}^{M \times N}$, the undecimated DCT is defined as a set of 64 convolutions with 64 DCT basis patterns $\mathbf{B}^{(k,l)}$:

$$\begin{aligned} \mathcal{U}(\mathbf{X}) &= \{\mathbf{U}^{(k,l)} | 0 \leq k, l \leq 7\} \\ \mathbf{U}^{(k,l)} &= \mathbf{X} \star \mathbf{B}^{(k,l)}, \end{aligned} \tag{8.2.1}$$

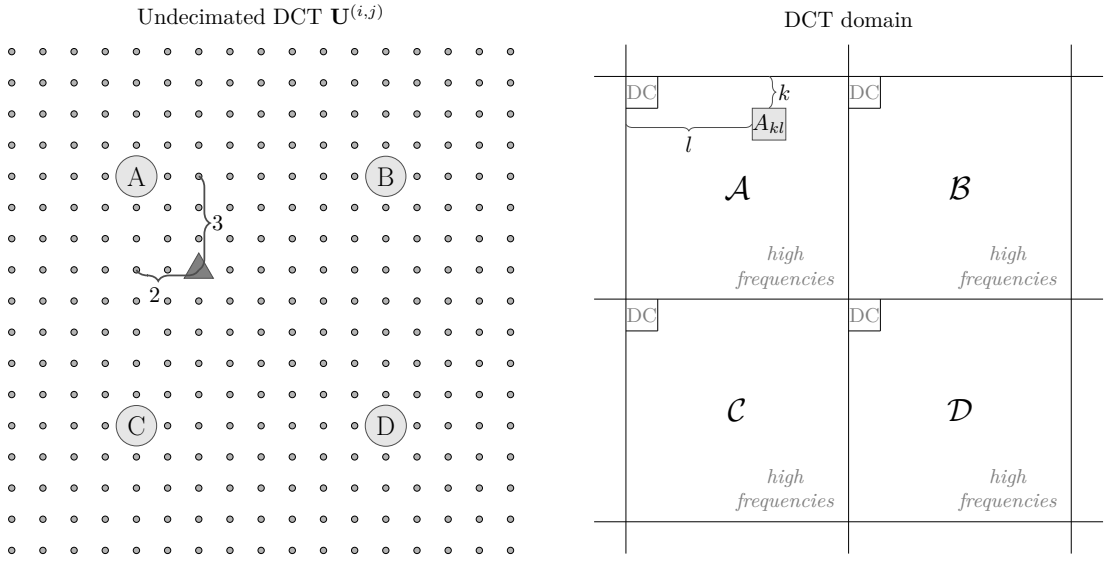

Figure 8.2.1: Left: Dots correspond to elements of $\mathbf{U}^{(i,j)} = \mathbf{X} \star \mathbf{B}^{(i,j)}$, circles correspond to grid points from $\mathcal{G}_{8\times 8}$ (DCT coefficients in the JPEG representation of $\mathbf{X}$). The triangle is an element $u \in \mathbf{U}^{(i,j)}$ with relative coordinates $(a, b) = (3, 2)$ w.r.t. its upper left neighbor (A) from $\mathcal{G}_{8\times 8}$. Right: JPEG representation of $\mathbf{X}$ when replacing each 8×8 pixel block with a block of quantized DCT coefficients.

where $\mathbf{U}^{(k,l)} \in \mathbb{R}^{(M-7)\times(N-7)}$ and $'\star'$ denotes a convolution without padding. The DCT patterns are 8×8 matrices, $\mathbf{B}^{(k,l)} = (B^{(k,l)}_{mn})$, $0 \le m, n \le 7$:

$$B^{(k,l)}_{mn} = \frac{\mathbf{w}_k \mathbf{w}_l}{4} \cos\frac{\pi k(2m+1)}{16} \cos\frac{\pi l(2n+1)}{16}, \tag{8.2.2}$$

and $\mathbf{w}_0 = 1/\sqrt{2}$, $\mathbf{w}_k = 1$ for $k > 0$.

When the image is stored in the JPEG format, before computing its undecimated DCT it is first decompressed to the spatial domain without quantizing the pixel values to $\{0, \ldots, 255\}$ to avoid loss of information.

For better readability, from now on we will reserve the indices k, l and i, j to index DCT modes (spatial frequencies); they will always be in the range $0 \le k, l, i, j \le 7$.

8.2.1.1 Relationship to prior art

The undecimated DCT has already found applications in steganalysis. The concept of calibration, for the first time introduced in [40], formally consists of computing the undecimated DTC, subsampling it on an 8×8 grid shifted by four pixels in each direction, and computing a reference feature vector from the subsampled and quantized signal. Liu [80] made use of the entire transform by computing 63 features and averaging them to form a more powerful reference that was used for calibration by difference and by ratio. In this chapter, we show that the undecimated DCT contains a lot of information that can be successfully used for steganalysis.

8.2.2 Properties

First, notice that when subsampling the convolution $\mathbf{U}^{(i,j)} = \mathbf{X} \star \mathbf{B}^{(i,j)}$ on the grid $\mathcal{G}_{8\times 8} = \{0, 7, 15, \ldots, M-9\} \times \{0, 7, 15, \ldots, N-9\}$ (circles in Figure 8.2.1on the left), one obtains all

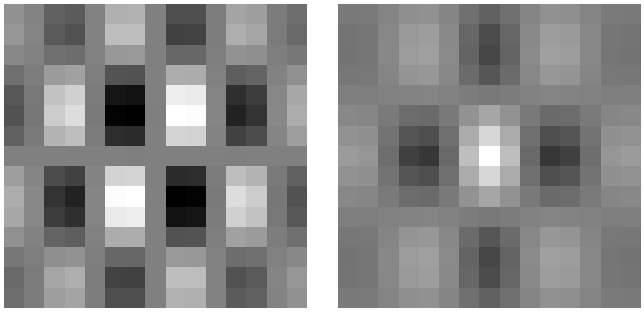

Figure 8.2.2: Examples of two unit responses scaled so that medium gray corresponds to zero.

unquantized values of DCT coefficients for DCT mode (i, j) that form the input into the JPEG representation of $\mathbf{X}$.

We will now take a look at how the values of the undecimated DCT $\mathcal{U}(\mathbf{X})$ are affected by changing one DCT coefficient of the JPEG representation of $\mathbf{X}$. Suppose one modifies a DCT coefficient in mode (k, l) in the JPEG file corresponding to $(m, n) \in \mathcal{G}_{8\times 8}$. This change will affect all 8×8 pixels in the corresponding block and an entire 15×15 neighborhood of values in $\mathbf{U}^{(i,j)}$ centered at $(m, n) \in \mathcal{G}_{8\times 8}$. In particular, the values will be modified by what we call the "unit response"

$$\mathbf{R}^{(i,j)(k,l)} = \mathbf{B}^{(i,j)} \otimes \mathbf{B}^{(k,l)}, \tag{8.2.3}$$

where $\otimes$ denotes the full cross-correlation. While this unit response is not symmetrical, its absolute values are symmetrical by both axes: $|\mathbf{R}^{(i,j)(k,l)}_{a,b}| = |\mathbf{R}^{(i,j)(k,l)}_{-a,b}|$, $|\mathbf{R}^{(i,j)(k,l)}_{a,b}| = |\mathbf{R}^{(i,j)(k,l)}_{a,-b}|$ for all $0 \le a, b \le 7$ when indexing $\mathbf{R} \in \mathbb{R}^{15\times 15}$ with indices in $\{-7, \dots 1, 0, 1, \dots, 7\}$.

Figure 8.2.2 shows two examples of unit responses. Note that the value at the center $(0, 0)$ is zero for the response on the left and 1 for the response on the right. This central value equals to 1 only when $i = k$ and $j = l$.

We now take a closer look at how a particular value $u \in \mathbf{U}^{(i,j)}$ is computed. First, we identify the four neighbors from the grid $\mathcal{G}_{8\times 8}$ that are closest to u (follow Figure 8.2.1 where the location of u is marked by a triangle). We will capture the position of u w.r.t. to its four closest neighbors from $\mathcal{G}_{8\times 8}$ using relative coordinates. With respect to the upper left neighbor (A), u is at position (a, b), $0 \le a, b, \le 7$ ($(a, b) = (3, 2)$ in Figure 8.2.1). The relative positions w.r.t. the other three neighbors (B–D) are, correspondingly, $(a, b - 8)$, $(a - 8, b)$, and $(a - 8, b - 8)$. Also recall that the elements of $\mathbf{U}^{(i,j)}$ collected across all (i, j), $0 \le i, j \le 7$, at A, form all non-quantized DCT coefficients corresponding to the 8×8 block $\mathcal{A}$ (see, again Figure 8.2.1).

Arranging the DCT coefficients from the neighboring blocks $\mathcal{A}$–$\mathcal{D}$ into 8×8 matrices A_{kl}, B_{kl}, C_{kl}, and D_{kl}, $u \in \mathbf{U}^{(i,j)}$ can be expressed as

$$\begin{aligned} u = \sum_{k=0}^{7}\sum_{l=0}^{7} Q_{kl} \Big[& A_{kl} R^{(i,j)(k,l)}_{a,b} + B_{kl} R^{(i,j)(k,l)}_{a,b-8} \\ & + C_{kl} R^{(i,j)(k,l)}_{a-8,b} + D_{kl} R^{(i,j)(k,l)}_{a-8,b-8} \Big], \end{aligned} \tag{8.2.4}$$

where the subscripts in $R^{(i,j)(k,l)}_{a,b}$ capture the position of u w.r.t. its upper left neighbor and Q_{kl} is the quantization step of the (k, l)-th DCT mode. This can be written as a projection of 256

dequantized DCT coefficients from four adjacent blocks from the JPEG file with a projection vector $\mathbf{p}_{a,b}^{(i,j)}$

$$u = \begin{pmatrix} Q_{00}A_{00} \\ \vdots \\ Q_{77}A_{77} \\ Q_{00}B_{00} \\ \vdots \\ Q_{77}B_{77} \\ \vdots \\ Q_{00}D_{00} \\ \vdots \\ Q_{77}D_{77} \end{pmatrix}^T \cdot \underbrace{\begin{pmatrix} R_{a,b}^{(i,j)(1,1)} \\ \vdots \\ R_{a,b}^{(i,j)(8,8)} \\ R_{a-8,b}^{(i,j)(1,1)} \\ \vdots \\ R_{a-8,b}^{(i,j)(8,8)} \\ \vdots \\ R_{a-8,b-8}^{(i,j)(1,1)} \\ \vdots \\ R_{a-8,b-8}^{(i,j)(8,8)} \end{pmatrix}}_{\mathbf{p}_{a,b}^{(i,j)}}. \tag{8.2.5}$$

It is proved in Section that the projection vectors form an orthonormal system satisfying for all (a,b), (i,j), and (k,l)

$$\mathbf{p}_{a,b}^{(i,j)T} \cdot \mathbf{p}_{a,b}^{(k,l)} = \delta_{(i,j),(k,l)}, \tag{8.2.6}$$

where δ is the Kronecker delta. The projection vectors also satisfy the following symmetry

$$\left|\mathbf{p}_{a,b}^{(i,j)}\right| = \left|\mathbf{p}_{a,b-8}^{(i,j)}\right| = \left|\mathbf{p}_{a-8,b}^{(i,j)}\right| = \left|\mathbf{p}_{a-8,b-8}^{(i,j)}\right| \tag{8.2.7}$$

for all i,j and a,b when interpreting the arithmetic operations on indices as mod8.

8.3 Orthonormality of projection vectors in undecimated DCT

Here, we provide the proof of orthonormality (8.2.6) of vectors $\mathbf{p}_{a,b}^{(k,l)}$ defined in (8.2.5). It will be useful to follow Figure 8.3.1 for easier understanding. For each a,b, $0 \le a,b \le 7$, the (i,j)th DCT basis pattern $\mathbf{B}^{(i,j)}$ positioned so that its upper left corner has relative index (a,b) is split into four 8×8 subpatterns: κ stands for cirκle, μ stands for diaμond, τ for τriangle, and σ for σtar:

$$\kappa_{mn}^{(i,j)} = \begin{cases} B_{m-a,n-b}^{(i,j)} & a \le m \le 7 \\ & b \le n \le 7 \\ 0 & \text{otherwise} \end{cases}$$

$$\mu_{mn}^{(i,j)} = \begin{cases} B_{m-a,8+n-b}^{(i,j)} & a \le m \le 7 \\ & 0 \le n < b \\ 0 & \text{otherwise} \end{cases}$$

$$\tau_{mn}^{(i,j)} = \begin{cases} B_{8+m-a,n-b}^{(i,j)} & 0 \le m < a \\ & b \le n \le 7 \\ 0 & \text{otherwise.} \end{cases}$$

$$\sigma_{mn}^{(i,j)} = \begin{cases} B_{8+m-a,8+n-b}^{(i,j)} & 0 \le m < a \\ & 0 \le n < b \\ 0 & \text{otherwise} \end{cases}$$

In Figure 8.3.1 top, the four patterns are shown using four different markers. The light-color markers correspond to zeros. The first 64 elements of $\mathbf{p}_{a,b}^{(i,j)}$ are simply projections of $\kappa_{mn}^{(i,j)}$ onto the 64 patterns

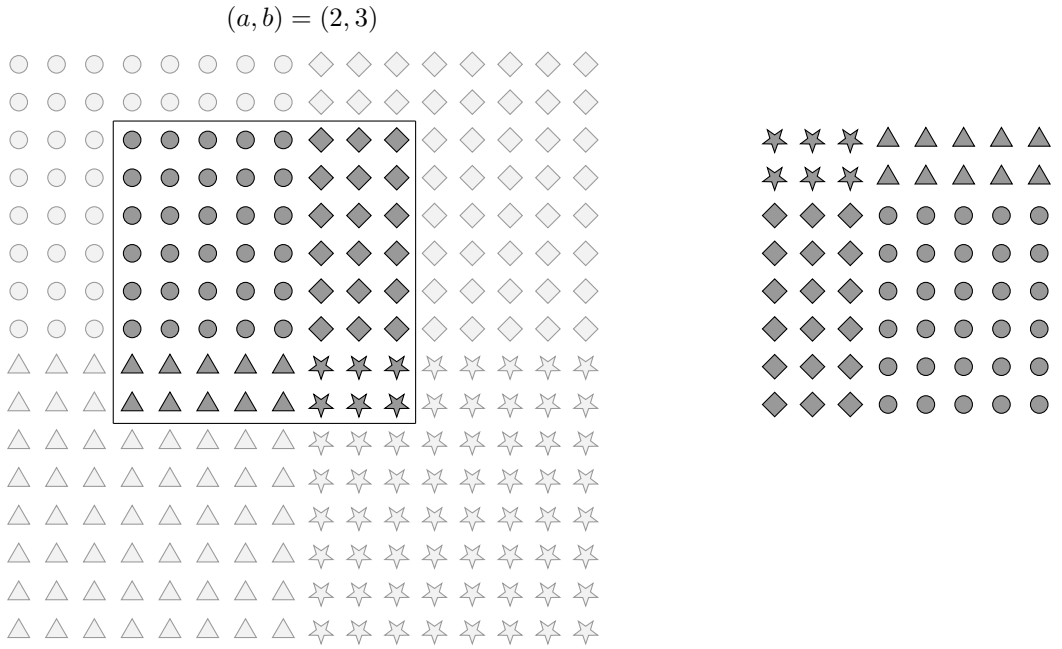

Figure 8.3.1: Diagram showing the auxiliary patterns κ (cirκle), μ (diaμond), τ (τriangle), and σ (σtar). The black square outlines the position of the DCT basis pattern $\mathbf{B}^{(i,j)}$.

forming the DCT basis. The next 64 elements are projections of $\mu_{mn}^{(i,j)}$ onto the DCT basis, the next 64 are projections of $\tau_{mn}^{(i,j)}$, and the last 64 are projections of $\sigma_{mn}^{(i,j)}$. We will denote these projections with the same Greek letters but with a single index instead: $(\kappa_1^{(i,j)}, \ldots, \kappa_{64}^{(i,j)})$, $(\mu_1^{(i,j)}, \ldots, \mu_{64}^{(i,j)})$, $(\tau_1^{(i,j)}, \ldots, \tau_{64}^{(i,j)})$, and $(\sigma_1^{(i,j)}, \ldots, \sigma_{64}^{(i,j)})$. In terms of the introduced notation,

$$\mathbf{p}_{a,b}^{(i,j)T} \cdot \mathbf{p}_{a,b}^{(k,l)} = \sum_{r=1}^{64} \kappa_r^{(i,j)} \kappa_r^{(k,l)} + \sum_{r=1}^{64} \mu_r^{(i,j)} \mu_r^{(k,l)} + \sum_{r=1}^{64} \tau_r^{(i,j)} \tau_r^{(k,l)} + \sum_{r=1}^{64} \sigma_r^{(i,j)} \sigma_r^{(k,l)}. \tag{8.3.1}$$

Note that the sum $\kappa^{(i,j)} + \mu^{(i,j)} + \tau^{(i,j)} + \sigma^{(i,j)}$ is the entire DCT mode (i, j) split into four pieces and rearranged back together to form an 8×8 block (Figure 8.3.1 right). For fixed a, b, due to the orthonormality of DCT modes (i, j) and (k, l), $\kappa^{(i,j)} + \mu^{(i,j)} + \tau^{(i,j)} + \sigma^{(i,j)}$ and $\kappa^{(k,l)} + \mu^{(k,l)} + \tau^{(k,l)} + \sigma^{(k,l)}$ are thus also orthonormal and so are their projections onto the DCT basis (because the DCT transform is orthonormal):

$$\sum_{r=1}^{64} (\kappa_r^{(i,j)} + \mu_r^{(i,j)} + \tau_r^{(i,j)} + \sigma_r^{(i,j)}) \times (\kappa_r^{(k,l)} + \mu_r^{(k,l)} + \tau_r^{(k,l)} + \sigma_r^{(k,l)}) = \delta_{(i,j),(k,l)}. \tag{8.3.2}$$

The orthonormality now follows from the fact that the LHS of (8.3.2) and the RHS of (8.3.1) have the exact same value because the sum of every mixed term in (8.3.2) is zero (e.g., $\sum_{r=1}^{64} \kappa_r^{(i,j)} \tau_r^{(k,l)} = 0$, etc.). This is because the subpatterns $\kappa^{(i,j)}$ and $\tau^{(k,l)}$ have *disjoint supports* (their dot product in the spatial domain is 0 and thus the product in the DCT domain is also 0 because DCT is orthonormal).

$a\backslash b$	0	1	2	3	4	5	6	7
0	a	b	c	d	e	d	c	b
1	e	f	g	h	i	h	g	f
2	j	k	l	m	n	m	l	k
3	o	p	q	r	s	r	q	p
4	t	u	v	w	x	w	v	u
5	o	p	q	r	s	r	q	p
6	j	k	l	m	n	m	l	k
7	e	f	g	h	i	h	g	f

Table 8.1: Histograms $\mathbf{h}_{a,b}$ to be merged are labeled with the same letter. All 64 histograms can thus be merged into 25. Light shading denotes merging of four histograms, medium shading two histograms, and dark shading denotes no merging.

8.4 DCTR features

The DCTR features are built by quantizing the absolute values of all elements in the undecimated DCT and collecting the first-order statistic separately for each mode (k,l) and each relative position (a,b), $0 \le a,b \le 7$. Formally, for each (k,l) we define the matrix[2] $\mathbf{U}^{(k,l)}_{a,b} \in \mathbb{R}^{(M-8)/8\times(N-8)/8}$ as a submatrix of $\mathbf{U}^{(k,l)}$ with elements whose relative coordinates w.r.t. the upper left neighbor in the grid $\mathcal{G}_{8\times 8}$ are (a,b). Thus, each $\mathbf{U}^{(k,l)} = \cup_{a,b=0}^{7}\mathbf{U}^{(k,l)}_{a,b}$ and $\mathbf{U}^{(k,l)}_{a,b}\cap\mathbf{U}^{(k,l)}_{a',b'} = \emptyset$ whenever $(a,b)\neq(a',b')$. The feature vector is formed by normalized histograms for $0 \le k,l \le 7$, $0 \le a,b \le 7$:

$$\mathbf{h}^{(k,l)}_{a,b}(r) = \frac{1}{\left|\mathbf{U}^{(k,l)}_{a,b}\right|}\sum_{u\in\mathbf{U}^{(k,l)}_{a,b}}[Q_T(|u|/q)=r], \tag{8.4.1}$$

where Q_T is a quantizer with integer centroids $\{0,1,\ldots,T\}$, q is the quantization step, and $[P]$ is the Iverson bracket equal to 0 when the statement P is false and 1 when P is true. We note that q could potentially depend on a,b as well as the DCT mode indices k,l, and the JPEG quality factor (see Section 8.4.3 for more discussions).

We work with absolute values because each $\mathbf{U}^{(i,j)}$ is an output of a high-pass filter and thus the distribution of $u \in \mathbf{U}^{(i,j)}_{a,b}$ is symmetrical centered at 0 for each i,j and a,b. This gives us features that have a lower dimension and are better populated.

Due to the symmetries of projection vectors (8.2.7), it is possible to further decrease the feature dimensionality by adding together the histograms corresponding to indices (a,b), $(a,8-b)$, $(8-a,b)$, and $(8-a,8-b)$ under the condition that these indices stay within $\{0,\ldots,7\}\times\{0,\ldots,7\}$ (see Figure 8.1). Note that for $(a,b)\in\{1,2,3,5,6,7\}^2$, we merge four histograms. When exactly one element of (a,b) is in $\{0,4\}$, only two histograms are merged, and when both a and b are in $\{0,4\}$ there is only one histogram. Thus, the total dimensionality of the symmetrized feature vector is $64\times(36/4+24/2+4)\times(T+1) = 1600\times(T+1)$.

In the rest of this section, we provide experimental evidence that working with absolute values and symmetrizing the features indeed improves the detection accuracy. We also experimentally determine the proper values of the threshold T and the quantization step q, and evaluate the performance of different parts of the DCTR feature vector w.r.t. the DCT mode indices k,l. For experiments in Sections 8.4.1–8.4.4, the steganographic method was J-UNIWARD at 0.4 bit per non-zero AC DCT coefficient (bpnzAC) with JPEG quality factor 75. We selected this steganographic method as an example of a state-of-the-art data hiding method for the JPEG domain.

[2]Since $\mathbf{U}^{(k,l)} \in \mathbb{R}^{(M-7)\times(N-7)}$, the height (width) of $\mathbf{U}^{(k,l)}_{a,b}$ is larger by one when $a=0$ ($b=0$).

$a\backslash b$	0	1	2	3	4	5	6	7
0	0.427	0.343	0.298	0.336	0.304	0.335	0.298	0.345
1	0.366	0.409	0.349	0.367	0.340	0.370	0.352	0.408
2	0.335	0.372	0.338	0.345	0.327	0.344	0.343	0.371
3	0.358	0.378	0.339	0.347	0.326	0.356	0.336	0.377
4	0.334	0.348	0.319	0.328	0.310	0.325	0.323	0.351
5	0.358	0.379	0.335	0.350	0.326	0.352	0.340	0.379
6	0.335	0.374	0.340	0.347	0.324	0.346	0.340	0.372
7	0.369	0.404	0.348	0.365	0.334	0.361	0.348	0.404

Table 8.2: $E^{\text{Single}}_{a,b}$ is the detection OOB error when steganalyzing with $\mathbf{h}_{a,b}$.

8.4.1 Symmetrization validation

In this section, we experimentally validate the feature symmetrization. We denote by $E_{\text{OOB}}(X)$ the OOB error obtained when using features X. The histograms concatenated over the DCT mode indices will be denoted as

$$\mathbf{h}_{a,b} = \bigvee_{k,l=0}^{7} \mathbf{h}^{(k,l)}_{a,b}. \tag{8.4.2}$$

For every combination of indices $a, b, c, d \in \{0, \dots, 7\}^2$, we computed three types of error (the symbol $'\&'$ means feature concatenation):

1. $E^{\text{Single}}_{a,b} \triangleq E_{\text{OOB}}(\mathbf{h}_{a,b})$
2. $E^{\text{Concat}}_{(a,b),(c,d)} \triangleq E_{\text{OOB}}(\mathbf{h}_{a,b} \vee \mathbf{h}_{c,d})$
3. $E^{\text{Merged}}_{(a,b),(c,d)} \triangleq E_{\text{OOB}}(\mathbf{h}_{a,b} + \mathbf{h}_{c,d})$

to see the individual performance of the features across the relative indices (a, b) as well as the impact of concatenating and merging the features on detectability. In the following experiments, we fixed $q = 4$ and $T = 4$. This gave each feature $\mathbf{h}_{a,b}$ the dimensionality of $64 \times (T + 1) = 320$.

Table 8.2 informs us about the individual performance of features $\mathbf{h}_{a,b}$. Despite the rather low dimensionality of 320, every $\mathbf{h}_{a,b}$ achieves a decent detection rate by itself (c.f., Figure 8.5.1 in Section 8.5).

The next experiment was aimed at assessing the loss of detection accuracy when merging histograms corresponding to different relative coordinates as opposed to concatenating them. When this drop of accuracy is approximately zero, both feature sets can be merged. Table 8.3 shows the detection drop $E^{\text{Merged}}_{(a,b),(c,d)} - E^{\text{Concat}}_{(a,b),(c,d)}$ when merging $\mathbf{h}_{1,2}$ with $\mathbf{h}_{c,d}$ as a function of c, d. The results clearly show which features should be merged; they are also consistent with the symmetries analyzed in Section 8.2.2.

8.4.2 Mode performance analysis

In this section, we analyze the performance of the DCTR features by DCT modes when steganalyzing with the merger $\mathbf{h}^{(k,l)} \triangleq \sum_{a,b=0}^{7} \mathbf{h}^{(k,l)}_{a,b}$ of dimension $25 \times (T + 1) = 125$. Interestingly, as Table 8.4 shows, for J-UNIWARD the histograms corresponding to high frequency modes provide the same or better distinguishing power than those of low frequencies.

$(a,b) = (1,2)$

$c\backslash d$	0	1	2	3	4	5	6	7
0	0.039	0.054	0.031	0.067	0.046	0.063	0.030	0.048
1	0.059	0.050	0	0.058	0.035	0.059	0.001	0.046
2	0.074	0.067	0.033	0.071	0.057	0.071	0.032	0.065
3	0.055	0.053	0.030	0.061	0.044	0.059	0.019	0.050
4	0.055	0.045	0.024	0.060	0.044	0.058	0.024	0.050
5	0.059	0.058	0.023	0.060	0.044	0.064	0.022	0.055
6	0.070	0.064	0.021	0.068	0.048	0.067	0.025	0.057
7	0.052	0.049	0.002	0.056	0.037	0.056	0.000	0.043

Table 8.3: $E^{\text{Merged}}_{(a,b),(c,d)} - E^{\text{Concat}}_{(a,b),(c,d)}$ for (a,b) as a function of (c,d).

	0	1	2	3	4	5	6	7
0	0.483	0.473	0.449	0.411	0.370	0.387	0.395	0.414
1	0.479	0.455	0.427	0.394	0.365	0.385	0.395	0.421
2	0.459	0.440	0.4220	0.398	0.392	0.397	0.405	0.424
3	0.446	0.420	0.414	0.421	0.426	0.428	0.427	0.431
4	0.419	0.403	0.406	0.423	0.432	0.443	0.438	0.438
5	0.407	0.399	0.407	0.428	0.445	0.453	0.451	0.440
6	0.406	0.402	0.410	0.428	0.448	0.460	0.446	0.427
7	0.402	0.422	0.423	0.434	0.435	0.439	0.434	0.433

Table 8.4: $E_{\text{OOB}}(\mathbf{h}^{(k,l)})$ as a function of k,l.

8.4.3 Feature quantization and normalization

In this section, we investigate the effect of quantization and feature normalization on the detection performance.

We carried out experiments for two quality factors, 75 and 95, and studied the effect of the quantization step q on detection accuracy (the two top charts in Figure 8.4.1). Additionally, we also investigated whether it is advantageous, prior to quantization, to normalize the features by the DCT mode quantization step, Q_{kl}, and by scaling $\mathbf{U}^{(k,l)}$ to a zero mean and unit variance (the two bottom charts in Figure 8.4.1).

Figure 8.4.1 shows that the effect of feature normalization is quite weak and it appears to be slightly more advantageous to not normalize the features and keep the feature design simple. The effect of the quantization step q is, however, much stronger. For quality factor 75 (95), the optimal quantization steps were 4 (0.8). Thus, we opted for the following linear fit[3] to obtain the proper value of q for an arbitrary quality factor in the range $50 \leq K \leq 99$:

$$q_K = 8 \times \left(2 - \frac{K}{50}\right). \tag{8.4.3}$$

8.4.4 Threshold

As Table 8.5 shows, the detection performance is quite insensitive to the threshold T. Although the best performance is achieved with $T = 6$, the gain is negligible compared to the dimensionality increase. Thus, in this chapter we opted for $T = 4$ as a good compromise between performance and detectability.

[3]Coincidentally, the term in the bracket corresponds to the multiplier used for computing standard quantization matrices.

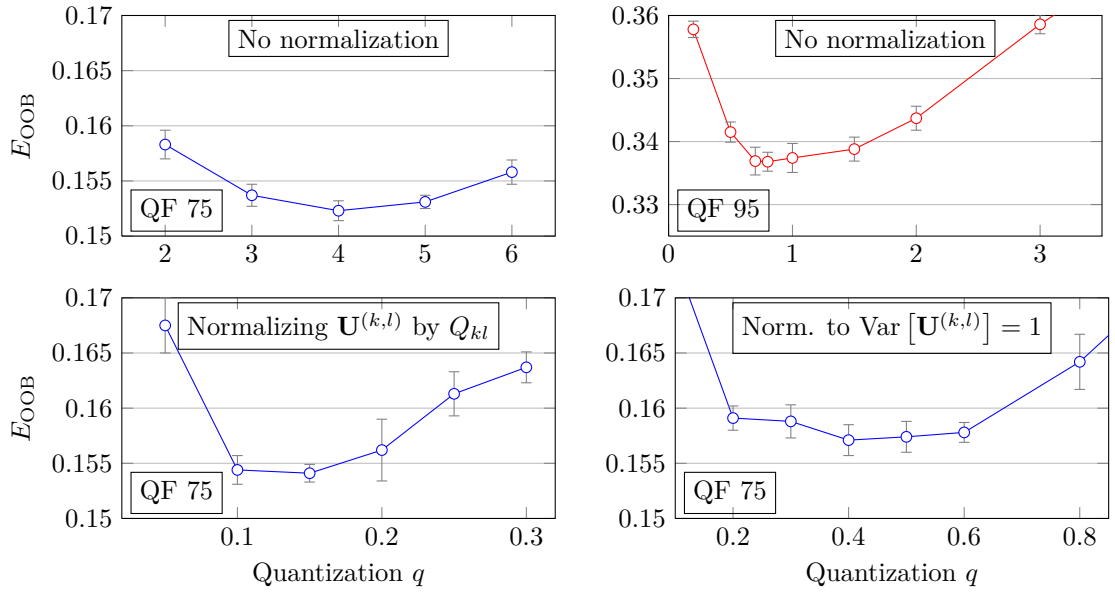

Figure 8.4.1: The effect of feature quantization without normalization (top charts) and with normalization (bottom charts) on detection accuracy.

T	3	4	5	6
E_{OOB}	0.1545	0.1523	0.1524	0.1519

Table 8.5: E_{OOB} of the entire DCTR feature set with dimensionality $1600 \times (T+1)$ as a function of the threshold T for J-UNIWARD at 0.4 bpnzAC.

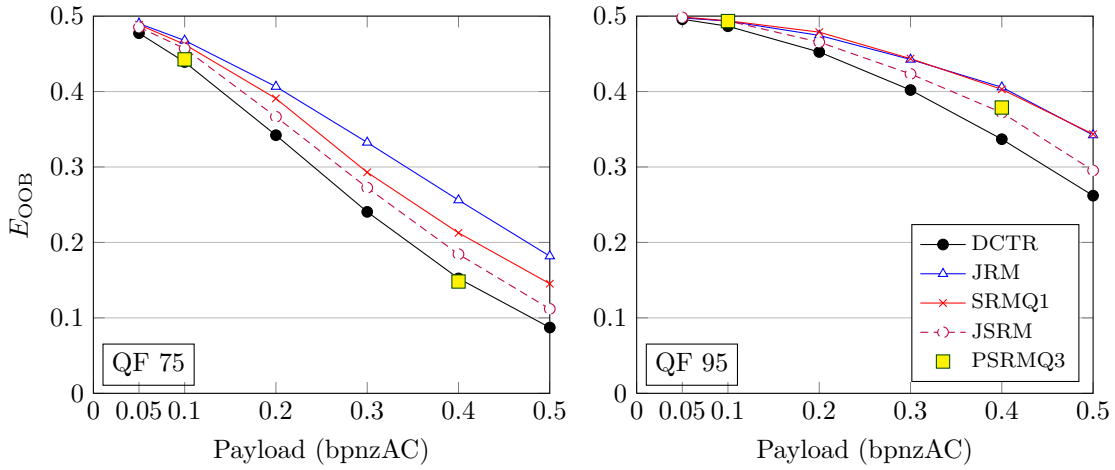

Figure 8.5.1: Detection error E_{OOB} for J-UNIWARD for quality factors 75 and 95 when steganalyzed with the proposed DCTR and other rich feature sets.

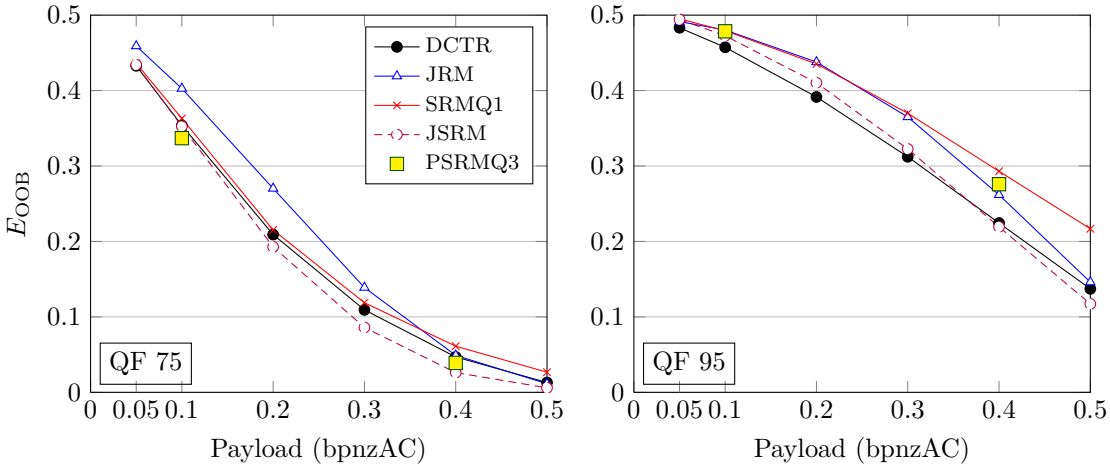

Figure 8.5.2: Detection error E_{OOB} for UED with ternary embedding for quality factors 75 and 95 when steganalyzed with the proposed DCTR and other rich feature sets.

To summarize, the final form of DCTR features includes the symmetrization as explained in Section 8.4, no normalization, quantization according to (8.4.3), and $T = 4$. This gives the DCTR set the dimensionality of 8,000.

8.5 Experiments

In this section, we subject the newly proposed DCTR feature set to tests on selected state-of-the-art JPEG steganographic schemes as well as examples of older embedding schemes. Additionally, we contrast the detection performance to previously proposed feature sets. Each time a separate classifier is trained for each image source, embedding method, and payload to see the performance differences.

Figures 8.5.1, 8.5.2 and 8.5.3 show the detection error E_{OOB} for J-UNIWARD [59], ternary-coded UED (Uniform Embedding Distortion) [51], and nsF5 [47] achieved using the proposed DCTR, the JPEG Rich Model (JRM) [74] of dimension 22,510, the 12,753-dimensional version of the Spatial

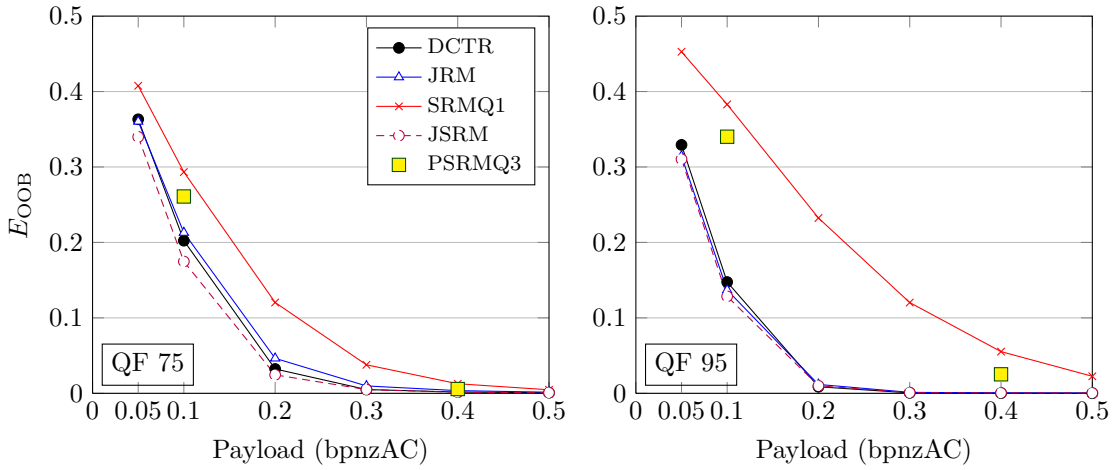

Figure 8.5.3: Detection error E_{OOB} for nsF5 for quality factors 75 and 95 when steganalyzed with the proposed DCTR and other rich feature sets.

Rich Model called SRMQ1 [43], the merger of JRM and SRMQ1 abbreviated as JSRM (dimension 35,263), and the 12,870 dimensional Projection Spatial Rich Model with quantization step 3 specially designed for the JPEG domain (PSRMQ3) [57]. When interpreting the results, one needs to take into account the fact that the DCTR has by far the lowest dimensionality and computational complexity of all tested feature sets.

The most significant improvement is seen for J-UNIWARD. Despite its compactness and significantly lower computational complexity, the DCTR set is the best performer for the higher quality factor and provides about the same level of detection as PSRMQ3 for quality factor 75. For the ternary UED, the DCTR is the best performer for the higher JPEG quality factor for all but the largest tested payload. For quality factor 75, the much larger 35,263-dimensional JSRM gives a slightly better detection. The DCTR also provides quite competitive detection for nsF5. The detection accuracy is roughly at the same level as for the 22,510-dimensional JRM.

The DCTR feature set is also performing quite well against the state-of-the-art side-informed JPEG algorithm SI-UNIWARD [59] (Figure 8.5.4). On the other hand, JSRM and JRM are better suited to detect NPQ [60] (Figure 8.5.5). This is likely because NPQ introduces (weak) embedding artifacts into the statistics of JPEG coefficients that are easier to detect by the JRM, whose features are entirely built as co-occurrences of JPEG coefficients. We also point out the saturation of the detection error below 0.5 for quality factor 95 and small payloads for both schemes. This phenomenon, which was explained in [59], is caused by the tendency of both algorithms to place embedding changes into four specific DCT coefficients.

In Table 8.6, we take a look at how complementary the DCTR features are in comparison to the other rich models. This experiment was run only for J-UNIWARD at 0.4 bpnzAC. The DCTR seems to well complement PSRMQ3 as this 20,870-dimensional merger achieves so far the best detection of J-UNIWARD, decreasing E_{OOB} by more than 3% w.r.t. the PSRMQ3 alone. Next, we report on the computational complexity when extracting the feature vector using a Matlab code. The extraction of the DCTR feature vector for one BOSSbase image is twice as fast as JRM, ten times faster than SRMQ1, and almost 200 times faster than the PSRMQ3. Furthermore, a C++ (Matlab MEX) implementation takes only between 0.5–1 sec.

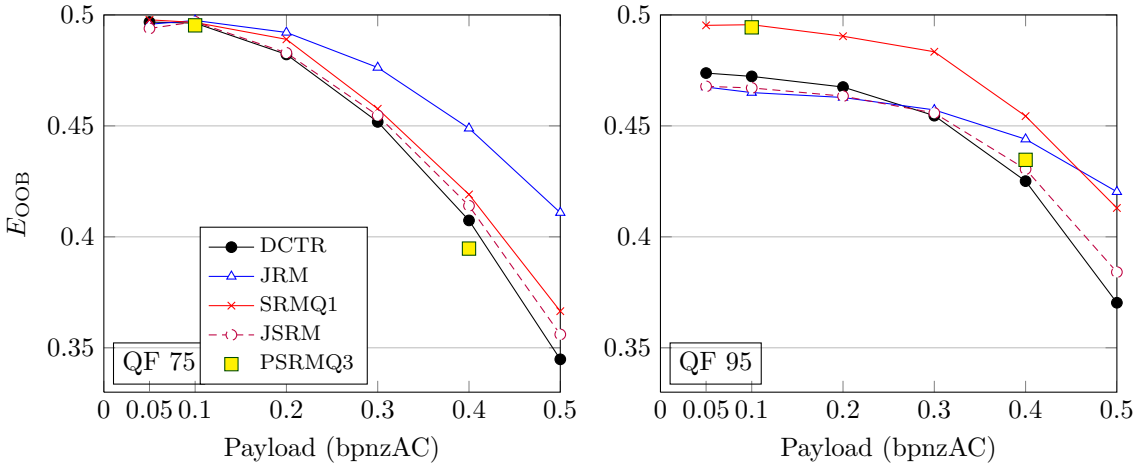

Figure 8.5.4: Detection error E_{OOB} for the side-informed SI-UNIWARD for quality factors 75 and 95 when steganalyzed with the proposed DCTR and other rich feature sets. Note the different scale of the y axis.

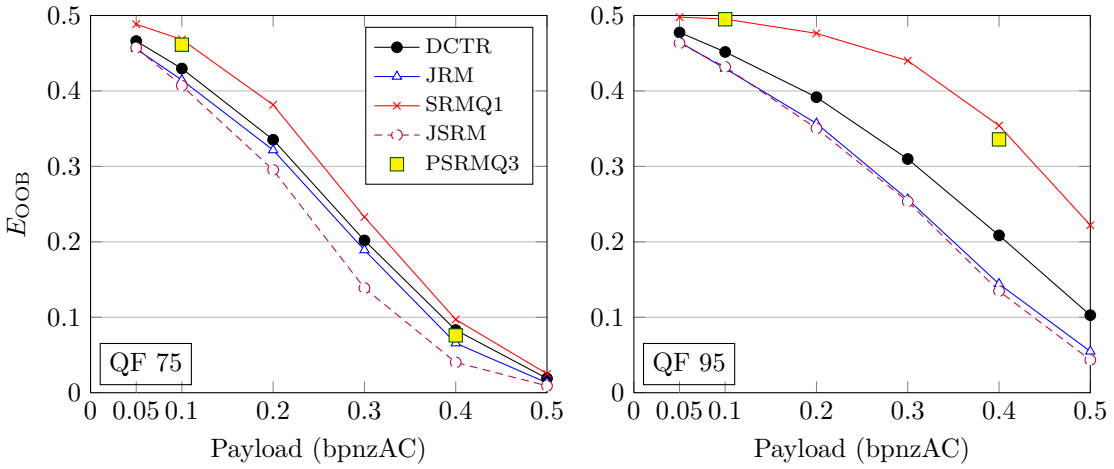

Figure 8.5.5: Detection error E_{OOB} for the side-informed NPQ for quality factors 75 and 95 when steganalyzed with the proposed DCTR and other rich feature sets.

DCTR (8000)	JRM (22510)	SRMQ1 (12753)	PSRMQ3 (12870)	E_{OOB}	Dim.	Time(s) (Matlab)
•				**0.1523**	8,000	3
	•			0.2561	22,510	6
		•		0.2127	12,753	30
			•	0.1482	12,870	520
•	•			0.1431	30,510	9
•		•		0.1407	20,753	33
•			•	**0.1146**	20,870	523
•	•	•		0.1316	43,263	39
•	•		•	0.1252	43,380	529
	•	•		0.1844	35,263	36
	•		•	0.1429	35,380	526

Table 8.6: Detection of J-UNIWARD at payload 0.4 bpnzAC when merging various feature sets. The table also shows the feature dimensionality and time required to extract a single feature for one BOSSbase image on an Intel i5 2.4 GHz computer platform.

8.6 Conclusion

This chapter introduces a novel feature set for steganalysis of JPEG images. Its name is DCTR because the features are computed from noise residuals obtained using the 64 DCT bases. Its main advantage over previous art is its relatively low dimensionality (8,000) and a significantly lower computational complexity while achieving a competitive detection across many JPEG algorithms. These qualities make DCTR a good candidate for building practical steganography detectors and in steganalysis applications where the detection accuracy and the feature extraction time are critical.

The DCTR feature set utilizes the so-called undecimated DCT. This transform has already found applications in steganalysis in the past. In particular, the reference features used in calibration are essentially computed from the undecimated DCT subsampled on an 8×8 grid shifted w.r.t. the JPEG grid. The main point of this chapter is the discovery that the undecimated DCT contains much more information that is quite useful for steganalysis.

In the spatial domain, the proposed feature set can be interpreted as a family of one-dimensional co-occurrences (histograms) of noise residuals obtained using kernels formed by DCT bases. Furthermore, the feature set can also be viewed in the JPEG domain as a projection-type model with orthonormal projection vectors. Curiously, we were unable to improve the detection performance by forming two-dimensional co-occurrences instead of first-order statistics. This is likely because the neighboring elements in the undecimated DCT are qualitatively different projections of DCT coefficients, making the neighboring elements essentially independent.

We contrast the detection accuracy and computational complexity of DCTR with four other rich models when used for detection of five JPEG steganographic methods, including two side-informed schemes.

Chapter 9

Conclusion

Recently, steganography and steganalysis have experienced an explosive growth caused by advancements in coding and scalable machine learning. This dissertation contributes to this rapidly developing field.

Modern steganalysis in the spatial domain is based on noise residuals extracted using pixel predictors. These predictors have a major impact on detection. This dissertation describes several general techniques for a) optimizing these residuals to achieve superior performance for a given cover source and steganographic method, and b) finding more efficient statistical descriptors of these residuals to further improve the detection accuracy. One of the most singificant contributions of this work is the Projection Spatial Rich Model. It improves upon the previously proposed Spatial Rich Model (SRM), which utilizes 45 hand-designed diverse linear and non-linear pixel predictors. While these predictors give the SRM a superior detection power, its ability to capture dependencies among the noise residuals is limited by the chosen statistical descriptors – sample joint distributions of neighboring residuals captured using co-occurrences. A new alternative proposed here is to replace the sparsely populated co-occurrences with a more robust statistical descriptor formed by first-order statistics of numerous random projections of noise residuals on larger pixel neighborhoods. This lead to a markedly improved detection rate for a fixed dimensionality as well an overall improved detection accuracy. The gain is especially markable for modern highly content adaptive steganographic schemes.

This dissertation also advances the detection of JPEG domain steganography. A new feature set was designed by utilizing the so-called undecimated JPEG tranform formed by residuals obtained using the 64 dicrete cosine transform bases. These residuals, which can be interpreted as projections of DCT coefficients onto a set of orthonormal projection vectors, have a strong distinguishing power to separate cover and stego images embedded with JPEG domain steganography. The new feature set called DCTR (Discrete Cosine Transform Residual) enjoys a markedly lower comptational complexity and competitive detection power across both older and modern steganographic algorithms hiding in the JPEG domain. This makes it an ideal candidate for practical applications where computational complexity and performance are crucial.

The biggest challenge in steganography and steganalysis in any domain is the absence of a model which would capture the complex dependencies among individual image elements (pixels, in the spatial domain and DCT coefficients in the JPEG domain). Due to this model absence, development of steganography is closely related to the development of features used in steganalysis and vice versa. In this dissertation, a steganographic method is introduced that avoids modeling the dependencies among image elements and instead quantifies the detectability as a distortion in the spatial domain as a sum of relative changes to wavelet coefficients (UNIWARD) independently of the domain in which the embedding is executed. The logic behind this choice is that the spatial domain is much better understood, therefore, preserving spatial domain statistics leads to preservation of the much more complex dependencies in the JPEG domain. This gave the universal UNIWARD methods a far

superior security over previously proposed schemes. The proposed UNIWARD distortion function can also be used to design steganography in JPEG domain when the steganographer has a side-information in the form of the uncompressed image (unquantized DCT coefficients).

Last but not least, the contributions in this dissertation indicate a possible paradigm shift. It appears that both the distortion functions for steganography as well as the features for detection should be built not in the domain where the embedding is executed but in the domain where the distortion is minimized. This appears to hold true for modern steganographic schemes free of easily identifiable embedding artifacts.

The main contributions of this dissertation can be summarized as follows:

- Steganography:
  - UNIWARD distortion family (S-UNIWARD, J-UNIWARD, and SI-UNIWARD) for design of steganographic scemes in an arbitrary domain. These are currently the most secure steganographic schemes in the spatial domain, JPEG domain, and JPEG domain with side-information.
- Steganalysis:
  - New statistical descriptor of noise residuals that repalces co-occurrence matrices with random projections of noise residuals. This increases the overall detection accuracy and significantly improves the detection accuracy vs. feature dimensionality trade off.
  - Low complexity JPEG domain features with competitive performance suitable for practical applications requiring computational efficiency and high detection accuracy.

Bibliography

[1] Fingerprints for car parts. *The Economist*, December 8, 2005.

[2] I. Avcibas, N. D. Memon, and B. Sankur. Steganalysis using image quality metrics. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents III*, volume 4314, pages 523–531, San Jose, CA, January 22–25, 2001.

[3] F. Bacon. *Of the Advancement and Proficiencie of Learning or the Partitions of Sciences*, volume VI. Leon Lichfield, Oxford, for R. Young and E. Forest, 1640.

[4] P. Bas, T. Filler, and T. Pevný. Break our steganographic system – the ins and outs of organizing BOSS. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, volume 6958 of Lecture Notes in Computer Science, pages 59–70, Prague, Czech Republic, May 18–20, 2011.

[5] S. Bayram, A. E. Dirik, H. T. Sencar, and N. Memon. An ensemble of classifiers approach to steganalysis. In *20th International Conference on Pattern Recognition (ICPR)*, pages 4376–4379, Istanbul, Turkey, August 23 2010.

[6] J. Bierbrauer. On Crandall's problem. Personal communication available from http://www.ws.binghamton.edu/fridrich/covcodes.pdf, 1998.

[7] R. Böhme. Weighted stego-image steganalysis for JPEG covers. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Science, pages 178–194, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[8] R. Böhme. *Improved Statistical Steganalysis Using Models of Heterogeneous Cover Signals*. PhD thesis, Faculty of Computer Science, Technische Universität Dresden, Germany, 2008.

[9] R. Böhme. *Advanced Statistical Steganalysis*. Springer-Verlag, Berlin Heidelberg, 2010.

[10] R. Böhme and A. Westfeld. Breaking Cauchy model-based JPEG steganography with first order statistics. In P. Samarati, P. Y. A. Ryan, D. Gollmann, and R. Molva, editors, *Computer Security - ESORICS 2004. Proceedings 9th European Symposium on Research in Computer Security*, volume 3193 of Lecture Notes in Computer Science, pages 125–140, Sophia Antipolis, France, September 13–15, 2004. Springer, Berlin.

[11] J. Borggaard, 2009, Software available at http://people.sc.fsu.edu/~jburkardt/m_src/nelder_mead/nelder_mead.html.

[12] L. Breiman. Bagging predictors. *Machine Learning*, 24:123–140, August 1996.

[13] D. Brewster. *Microscope*, volume XIV. Encyclopaedia Britannica or the Dictionary of Arts, Sciences, and General Literature, Edinburgh, IX – Application of photography to the microscope, 8th edition, 1857.

[14] I. Burrell. Fifty held in worldwide paedophile crackdown. *The Independent*, July 3, 2002.

[15] C. Cachin. An information-theoretic model for steganography. In D. Aucsmith, editor, *Information Hiding, 2nd International Workshop*, volume 1525 of Lecture Notes in Computer Science, pages 306–318, Portland, OR, April 14–17, 1998. Springer-Verlag, New York.

[16] J. Carr. Anti-forensic methods used by Jihadist web sites. *eSecurity Planet*, August 16, 2007.

[17] C. Chen and Y. Q. Shi. JPEG image steganalysis utilizing both intrablock and interblock correlations. In *Circuits and Systems, ISCAS 2008. IEEE International Symposium on*, pages 3029–3032, Seattle, WA, May, 18–21, 2008.

[18] V. Chonev and A. D. Ker. Feature restoration and distortion metrics. In A. Alattar, N. D. Memon, E. J. Delp, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics III*, volume 7880, pages 0G01–0G14, San Francisco, CA, January 23–26, 2011.

[19] R. Cogranne and F. Retraint. Application of hypothesis testing theory for optimal detection of LSB Matching data hiding. *Signal Processing*, 93(7):1724–1737, July, 2013.

[20] R. Cogranne and F. Retraint. An asymptotically uniformly most powerful test for LSB Matching detection. *IEEE Transactions on Information Forensics and Security*, 8(3):464–476, 2013.

[21] R. Cogranne, C. Zitzmann, L. Fillatre, F. Retraint, I. Nikiforov, and P. Cornu. A cover image model for reliable steganalysis. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 178–192, Prague, Czech Republic, May 18–20, 2011.

[22] R. Cogranne, C. Zitzmann, F. Retraint, I. Nikiforov, L. Fillatre, and P. Cornu. Statistical detection of LSB Matching using hypothesis testing theory. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 46–62, Berkeley, California, May 15–18, 2012.

[23] T. M. Cover and J. A. Thomas. *Elements of Information Theory*. New York: John Wiley & Sons, Inc., 1991.

[24] R. Crandall. Some notes on steganography. *Steganography Mailing List*, available from http://dde.binghamton.edu/download/Crandall_matrix.pdf, 1998.

[25] A. Dasgupta. Mumbai police fail to crack July 11 suspects' mail. *Daily News & Analysis*, October 16, 2006.

[26] R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification*. New York: John Wiley & Sons, Inc., 2nd edition, 2001.

[27] R. Engel. Al-Qaida spreads across the web. *NBC Nightly News*, August 21, 2007.

[28] T. Ernst. Schwarzweisse Magie. Der Schlüssel zum dritten Buch der *Steganographia* des Trithemius. *Daphnis*, 25(1), 1996.

[29] J. Huang F. Huang, W. Luo and Y.-Q. Shi. Distortion function designing for JPEG steganography with uncompressed side-image. In W. Puech, M. Chaumont, J. Dittmann, and P. Campisi, editors, *1st ACM IH&MMSec. Workshop*, Montpellier, France, June 17–19, 2013.

[30] H. Farid. Detecting steganographic messages in digital images. Technical Report TR2001-412, Dartmouth College, New Hampshire, 2001.

[31] H. Farid and L. Siwei. Detecting hidden messages using higher-order statistics and support vector machines. In F. A. P. Petitcolas, editor, *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 340–354, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[32] L. Fillatre. Adaptive steganalysis of least significant bit replacement in grayscale images. *IEEE Transactions on Signal Processing*, 60(2):556–569, 2011.

[33] T. Filler and J. Fridrich. Gibbs construction in steganography. *IEEE Transactions on Information Forensics and Security*, 5(4):705–720, 2010.

[34] T. Filler and J. Fridrich. Design of adaptive steganographic schemes for digital images. In A. Alattar, N. D. Memon, E. J. Delp, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics III*, volume 7880, pages OF 1–14, San Francisco, CA, January 23–26, 2011.

[35] T. Filler, J. Judas, and J. Fridrich. Minimizing additive distortion in steganography using syndrome-trellis codes. *IEEE Transactions on Information Forensics and Security*, 6(3):920–935, September 2011.

[36] T. Filler, T. Pevný, and P. Bas. BOSS (Break Our Steganography System). http://www.agents.cz/boss, July 2010.

[37] J. Fridrich. Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes. In J. Fridrich, editor, *Information Hiding, 6th International Workshop*, volume 3200 of Lecture Notes in Computer Science, pages 67–81, Toronto, Canada, May 23–25, 2004. Springer-Verlag, New York.

[38] J. Fridrich. *Steganography in Digital Media: Principles, Algorithms, and Applications*. Cambridge University Press, 2009.

[39] J. Fridrich and R. Du. Secure steganographic methods for palette images. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of Lecture Notes in Computer Science, pages 47–60, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[40] J. Fridrich, M. Goljan, and D. Hogea. Steganalysis of JPEG images: Breaking the F5 algorithm. In *Information Hiding, 5th International Workshop*, volume 2578 of Lecture Notes in Computer Science, pages 310–323, Noordwijkerhout, The Netherlands, October 7–9, 2002. Springer-Verlag, New York.

[41] J. Fridrich, M. Goljan, and D. Hogea. New methodology for breaking steganographic techniques for JPEGs. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security and Watermarking of Multimedia Contents V*, volume 5020, pages 143–155, Santa Clara, CA, January 21–24, 2003.

[42] J. Fridrich, M. Goljan, D. Soukal, and P. Lisoněk. Writing on wet paper. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 328–340, San Jose, CA, January 16–20, 2005.

[43] J. Fridrich and J. Kodovský. Rich models for steganalysis of digital images. *IEEE Transactions on Information Forensics and Security*, 7(3):868–882, June 2011.

[44] J. Fridrich and J. Kodovský. Steganalysis of lsb replacement using parity-aware features. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 31–45, Berkeley, California, May 15–18, 2012.

[45] J. Fridrich, J. Kodovský, M. Goljan, and V. Holub. Breaking HUGO – the process discovery. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, Prague, Czech Republic, May 18–20, 2011.

[46] J. Fridrich, J. Kodovský, M. Goljan, and V. Holub. Steganalysis of content-adaptive steganography in spatial domain. In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 102–117, Prague, Czech Republic, May 18–20, 2011.

[47] J. Fridrich, T. Pevný, and J. Kodovský. Statistically undetectable JPEG steganography: Dead ends, challenges, and opportunities. In J. Dittmann and J. Fridrich, editors, *Proceedings of the 9th ACM Multimedia & Security Workshop*, pages 3–14, Dallas, TX, September 20–21, 2007.

[48] F. Galand and G. Kabatiansky. Information hiding by coverings. In *Proceedings IEEE, Information Theory Workshop, ITW 2003*, pages 151–154, Paris, France, March 31–April 4, 2003.

[49] M. Goljan, J. Fridrich, and T. Holotyak. New blind steganalysis and its implications. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VIII*, volume 6072, pages 1–13, San Jose, CA, January 16–19, 2006.

[50] G. Gül and F. Kurugollu. A new methodology in steganalysis : Breaking highly undetactable steganograpy (HUGO). In T. Filler, T. Pevný, A. Ker, and S. Craver, editors, *Information Hiding, 13th International Conference*, Lecture Notes in Computer Science, pages 71–84, Prague, Czech Republic, May 18–20, 2011.

[51] L. Guo, J. Ni, and Y.-Q. Shi. An efficient JPEG steganographic scheme using uniform embedding. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[52] Herodotus. *The Histories*. Penguin Books, London, 1996. Translated by Aubrey de Sélincourt.

[53] T. S. Holotyak, J. Fridrich, and D. Soukal. Stochastic approach to secret message length estimation in $\pm k$ embedding steganography. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents VII*, volume 5681, pages 673–684, San Jose, CA, January 16–20, 2005.

[54] V. Holub and J. Fridrich. Designing steganographic distortion using directional filters. In *Fourth IEEE International Workshop on Information Forensics and Security*, Tenerife, Spain, December 2–5, 2012.

[55] V. Holub and J. Fridrich. Optimizing pixel predictors for steganalysis. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 09–1–09–13, San Francisco, CA, January 23–26, 2012.

[56] V. Holub and J. Fridrich. Digital image steganography using universal distortion. In W. Puech, M. Chaumont, J. Dittmann, and P. Campisi, editors, *1st ACM IH&MMSec. Workshop*, Montpellier, France, June 17–19, 2013.

[57] V. Holub and J. Fridrich. Random projections of residuals for digital image steganalysis. *IEEE Transactions on Information Forensics and Security*, 8(12):1996–2006, December 2013.

[58] V. Holub and J. Fridrich. Challenging the doctrines of jpeg steganography. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, San Francisco, CA, February 2–6, 2014.

[59] V. Holub, J. Fridrich, and T. Denemark. Universal distortion function for steganography in an arbitrary domain. *EURASIP Journal on Information Security (Revised Selected Papers of ACM IH and MMS 2013)*, 2013.

[60] F. Huang, J. Huang, and Y.-Q. Shi. New channel selection rule for JPEG steganography. *IEEE Transactions on Information Forensics and Security*, 7(4):1181–1191, August 2012.

[61] N. F. Johnson and P. Sallee. Detection of hidden information, covert channels and information flows. In John G. Voeller, editor, *Wiley Handbook of Science Technology for Homeland Security*. New York: Wiley & Sons, Inc, April 4, 2008.

[62] J. Kelley. Terrorist instructions hidden online. *USA Today*, February 5, 2001.

[63] A. D. Ker. Implementing the projected spatial rich features on a GPU.

[64] A. D. Ker. A fusion of maximal likelihood and structural steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Science, pages 204–219, Saint Malo, France, June 11–13, 2007. Springer-Verlag, Berlin.

[65] A. D. Ker and R. Böhme. Revisiting weighted stego-image steganalysis. In E. J. Delp, P. W. Wong, J. Dittmann, and N. D. Memon, editors, *Proceedings SPIE, Electronic Imaging, Security, Forensics, Steganography, and Watermarking of Multimedia Contents X*, volume 6819, pages 5 1–17, San Jose, CA, January 27–31, 2008.

[66] A. D. Ker and T. Pevný. Identifying a steganographer in realistic and heterogeneous data sets. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0N 1–13, San Francisco, CA, January 23–26, 2012.

[67] A. D. Ker, T. Pevný, J. Kodovský, and J. Fridrich. The Square Root Law of steganographic capacity. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 107–116, Oxford, UK, September 22–23, 2008.

[68] Y. Kim, Z. Duric, and D. Richards. Modified matrix encoding technique for minimal distortion steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 314–327, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[69] J. Kodovský. On dangers of cross-validation in steganalysis. Technical report, Binghamton University, August 2011.

[70] J. Kodovský and J. Fridrich. On completeness of feature spaces in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 123–132, Oxford, UK, September 22–23, 2008.

[71] J. Kodovský and J. Fridrich. Calibration revisited. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 63–74, Princeton, NJ, September 7–8, 2009.

[72] J. Kodovský and J. Fridrich. Quantitative structural steganalysis of Jsteg. *IEEE Transactions on Information Forensics and Security*, 5(4):681–693, December 2010.

[73] J. Kodovský and J. Fridrich. Steganalysis in high dimensions: Fusing classifiers built on random subspaces. In A. Alattar, N. D. Memon, E. J. Delp, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics III*, volume 7880, pages OL 1–13, San Francisco, CA, January 23–26, 2011.

[74] J. Kodovský and J. Fridrich. Steganalysis of JPEG images using rich models. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0A 1–13, San Francisco, CA, January 23–26, 2012.

[75] J. Kodovský and J. Fridrich. Steganalysis in resized images. In *Proc. of IEEE ICASSP*, Vancouver, Canada, May 26–31, 2013.

[76] J. Kodovský, J. Fridrich, and V. Holub. On dangers of overtraining steganography to incomplete cover model. In J. Dittmann, S. Craver, and C. Heitzenrater, editors, *Proceedings of the 13th ACM Multimedia & Security Workshop*, pages 69–76, Niagara Falls, NY, September 29–30, 2011.

[77] J. Kodovský, J. Fridrich, and V. Holub. Ensemble classifiers for steganalysis of digital media. *IEEE Transactions on Information Forensics and Security*, 7(2):432–444, 2012.

[78] G. Kolata. Veiled messages of terror may lurk in cyberspace. *The New York Times*, October 30, 2001.

[79] Liyun Li, H. T. Sencar, and N. Memon. A cost-effective decision tree based approach to steganalysis. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2013*, volume 8665, pages 0P 1–7, San Francisco, CA, February 5–7, 2013.

[80] Q. Liu. Steganalysis of DCT-embedding based adaptive steganography and yass. In J. Dittmann, S. Craver, and C. Heitzenrater, editors, *Proceedings of the 13th ACM Multimedia & Security Workshop*, pages 77–86, Niagara Falls, NY, September 29–30, 2011.

[81] I. Lubenko and A. D. Ker. Going from small to large data sets in steganalysis. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages OM 1–10, San Francisco, CA, January 23–26, 2012.

[82] W. Luo, F. Huang, and J. Huang. Edge adaptive image steganography based on LSB matching revisited. *IEEE Transactions on Information Forensics and Security*, 5(2):201–214, June 2010.

[83] S. Lyu and H. Farid. Steganalysis using higher-order image statistics. *IEEE Transactions on Information Forensics and Security*, 1(1):111–119, 2006.

[84] D. Montgomery. Arrests of alleged spies draws attention to long obscure field of steganography. *The Washington Post*, June 30, 2010.

[85] J. Nocedal and S. Wright. *Numerical Optimization*. Springer, 2nd edition edition, 2006.

[86] T. Pevný. Detecting messages of unknown length. In A. Alattar, N. D. Memon, E. J. Delp, and J. Dittmann, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security and Forensics III*, volume 7880, pages OT 1–12, San Francisco, CA, January 23–26, 2011.

[87] T. Pevný. Co-occurrence steganalysis in high dimension. In A. Alattar, N. D. Memon, and E. J. Delp, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2012*, volume 8303, pages 0B 1–13, San Francisco, CA, January 23–26, 2012.

[88] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 75–84, Princeton, NJ, September 7–8, 2009.

[89] T. Pevný, P. Bas, and J. Fridrich. Steganalysis by subtractive pixel adjacency matrix. *IEEE Transactions on Information Forensics and Security*, 5(2):215–224, June 2010.

[90] T. Pevný, T. Filler, and P. Bas. Using high-dimensional image models to perform highly undetectable steganography. In R. Böhme and R. Safavi-Naini, editors, *Information Hiding, 12th International Conference*, volume 6387 of Lecture Notes in Computer Science, pages 161–177, Calgary, Canada, June 28–30, 2010. Springer-Verlag, New York.

[91] T. Pevný and J. Fridrich. Merging Markov and DCT features for multi-class JPEG steganalysis. In E. J. Delp and P. W. Wong, editors, *Proceedings SPIE, Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, volume 6505, pages 3 1–3 14, San Jose, CA, January 29–February 1, 2007.

[92] T. Pevný and J. Fridrich. Novelty detection in blind steganalysis. In A. D. Ker, J. Dittmann, and J. Fridrich, editors, *Proceedings of the 10th ACM Multimedia & Security Workshop*, pages 167–176, Oxford, UK, September 22–23, 2008.

[93] N. Provos. Defending against statistical steganalysis. In *10th USENIX Security Symposium*, pages 323–335, Washington, DC, August 13–17, 2001.

[94] J. A. Reeds. Solved: The ciphers in Book III of Trithemius's S*teganographia*. *Cryptologia*, 22:291–319, October 1998.

[95] V. Sachnev, H. J. Kim, and R. Zhang. Less detectable JPEG steganography method based on heuristic optimization and BCH syndrome coding. In J. Dittmann, S. Craver, and J. Fridrich, editors, *Proceedings of the 11th ACM Multimedia & Security Workshop*, pages 131–140, Princeton, NJ, September 7–8, 2009.

[96] V. Schwamberger and M.O. Franz. Simple algorithmic modifications for improving blind steganalysis performance. In J. Dittmann, S. Craver, and P. Campisi, editors, *Proceedings of the 12th ACM Multimedia & Security Workshop*, pages 225–230, Rome, Italy, September 9–10, 2010.

[97] C. E. Shannon. Coding theorems for a discrete source with a fidelity criterion. *IRE Nat. Conv. Rec.*, 4:142–163, 1959.

[98] T. Sharp. An implementation of key-based digital signal steganography. In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of Lecture Notes in Computer Science, pages 13–26, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[99] Y. Q. Shi, C. Chen, and W. Chen. A Markov process based approach to effective attacking JPEG steganography. In J. L. Camenisch, C. S. Collberg, N. F. Johnson, and P. Sallee, editors, *Information Hiding, 8th International Workshop*, volume 4437 of Lecture Notes in Computer Science, pages 249–264, Alexandria, VA, July 10–12, 2006. Springer-Verlag, New York.

[100] Y.-Q. Shi, P. Sutthiwan, and L. Chen. Textural features for steganalysis. In M. Kirchner and D. Ghosal, editors, *Information Hiding, 14th International Conference*, volume 7692 of Lecture Notes in Computer Science, pages 63–77, Berkeley, California, May 15–18, 2012.

[101] G. J. Simmons. The prisoner's problem and the subliminal channel. In D. Chaum, editor, *Advances in Cryptology, CRYPTO '83*, pages 51–67, Santa Barbara, CA, August 22–24, 1983. New York: Plenum Press.

[102] K. Solanki, A. Sarkar, and B. S. Manjunath. YASS: Yet another steganographic scheme that resists blind steganalysis. In T. Furon, F. Cayre, G. Doërr, and P. Bas, editors, *Information Hiding, 9th International Workshop*, volume 4567 of Lecture Notes in Computer Science, pages 16–31, Saint Malo, France, June 11–13, 2007. Springer-Verlag, New York.

[103] J. Fridrich T. Denemark and V. Holub. Further study on security of s-uniward. In A. Alattar, N. D. Memon, and C. Heitzenrater, editors, *Proceedings SPIE, Electronic Imaging, Media Watermarking, Security, and Forensics 2014*, volume 9028, San Francisco, CA, February 2–6, 2014.

[104] T. Thai, R. Cogranne, and F. Retraint. Statistical model of quantized DCT coefficients: Application in the steganalysis of Jsteg algorithm. *Image Processing, IEEE Transactions on*, 23(5):1–14, May 2014.

[105] D. Upham. Steganographic algorithm JSteg. Software available at http://zooid.org/~paul/crypto/jsteg.

[106] M. Vetterli and J. Kovacevic. *Wavelets and Subband Coding*. Prentice Hall Signal Processing Series, 1995.

[107] C. Wang and J. Ni. An efficient JPEG steganographic scheme based on the block–entropy of DCT coefficents. In *Proc. of IEEE ICASSP*, Kyoto, Japan, March 25–30, 2012.

[108] A. Westfeld. High capacity despite better steganalysis (F5 – a steganographic algorithm). In I. S. Moskowitz, editor, *Information Hiding, 4th International Workshop*, volume 2137 of Lecture Notes in Computer Science, pages 289–302, Pittsburgh, PA, April 25–27, 2001. Springer-Verlag, New York.

[109] A. Westfeld. Generic adoption of spatial steganalysis to transformed domain. In K. Solanki, K. Sullivan, and U. Madhow, editors, *Information Hiding, 10th International Workshop*, volume 5284 of Lecture Notes in Computer Science, pages 161–177, Santa Barbara, CA, June 19–21, 2007. Springer-Verlag, New York.

[110] A. Westfeld and A. Pfitzmann. Attacks on steganographic systems. In A. Pfitzmann, editor, *Information Hiding, 3rd International Workshop*, volume 1768 of Lecture Notes in Computer Science, pages 61–75, Dresden, Germany, September 29–October 1, 1999. Springer-Verlag, New York.

[111] G. Winkler. *Image Analysis, Random Fields and Markov Chain Monte Carlo Methods: A Mathematical Introduction (Stochastic Modelling and Applied Probability)*. Springer-Verlag, Berlin Heidelberg, 2nd edition, 2003.

[112] T. Zhang and X. Ping. A fast and effective steganalytic technique against Jsteg-like algorithms. In *Proceedings of the ACM Symposium on Applied Computing*, pages 307–311, Melbourne, FL, March 9–12, 2003.

[113] C. Zitzmann, R. Cogranne, L. Fillatre, I. Nikiforov, F. Retraint, and P. Cornu. Hidden information detection based on quantized Laplacian distribution. In *Proc. IEEE ICASSP*, Kyoto, Japan, March 25-30, 2012.

[114] D. Zou, Y. Q. Shi, W. Su, and G. Xuan. Steganalysis based on Markov model of thresholded prediction-error image. In *Proceedings IEEE, International Conference on Multimedia and Expo*, pages 1365–1368, Toronto, Canada, July 9–12, 2006.