

Università degli Studi di Siena

Dipartimento di Ingegneria dell'Informazione

Ph.D. Thesis - Ciclo XXI – May 13th, 2009

New techniques for steganography and steganalysis in the pixel domain

Author: Ing. Giacomo Cancelli

Supervisor: Prof. Mauro Barni

Jury:

| Prof. | Andrea Abrardo | Università degli Studi di Siena |
|-----------|------------------|-----------------------------------|
| Prof. | Jean-Luc Dugelay | Institut Eurécom |
| Prof. | Alessandro Piva | Università degli Studi di Firenze |
| Reviewers | 5: | |
| Dr. | Gwenaël Doërr | University College London |
| Dr. | Andreas Westfeld | Technische Universität Dresden |

Contents

| 1 | Intr | oduction | 1 | |
|---|--|--|----|--|
| | 1.1 | Contributions of the thesis | 5 | |
| | | 1.1.1 ALE | 6 | |
| | | 1.1.2 Comparative Methodology in Steganalysis | 6 | |
| | | 1.1.3 MPSteg-color | 7 | |
| | 1.2 | Thesis organization | 7 | |
| | | | | |
| Ι | ± 1 | embedding steganalysis | 9 | |
| 2 | 2 Steganalysis: a classification problem | | | |
| | 2.1 | Training | 12 | |
| | 2.2 | Testing | 15 | |
| | | 2.2.1 Cross validation | 15 | |
| | | 2.2.2 Performance measures | 15 | |
| | 2.3 | Fisher Linear Discriminant Analysis | 20 | |
| 3 | ± 1 (| embedding: state of art | 23 | |
| | 3.1 | ± 1 embedding steganography $\ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots \ldots$ | 23 | |
| | 3.2 | ± 1 embedding steganalyzers \ldots \ldots \ldots \ldots \ldots \ldots | 24 | |
| | | 3.2.1 High Order Statistics of the Stego Noise (WAM) | 24 | |

| | | 3.2.2 | Center of Mass of the Histogram Characteristic Function | |
|----|------|---------|--|----|
| | | | (2D-HCFC) | 26 |
| 4 | Amj | olitude | of Local Extrema | 29 |
| | 4.1 | Improv | ving previous work on histogram domain | 29 |
| | | 4.1.1 | Removing Interferences at the Histogram Borders | 30 |
| | 4.2 | Consid | lering 2D Adjacency Histograms | 31 |
| | 4.3 | Perfor | mances of ALE | 33 |
| | | 4.3.1 | Setup | 33 |
| | | 4.3.2 | Results | 35 |
| | 4.4 | Hybric | l Algorithm | 37 |
| | 4.5 | Discus | sion | 37 |
| 5 | Exp | eriment | tal comparison among ± 1 embedding steganalysis | 39 |
| | 5.1 | Databa | ases | 39 |
| | 5.2 | Experi | mental Procedure | 41 |
| | 5.3 | Experi | mental Results | 42 |
| | | 5.3.1 | Impact of the source of imagery | 42 |
| | | 5.3.2 | Impact of the embedding rate | 46 |
| | | 5.3.3 | Performances of the steganalyzers with prior information about | |
| | | | the source of imagery | 49 |
| | | 5.3.4 | Performances of the steganalyzers without prior information | |
| | | | about the source of imagery | 51 |
| | 5.4 | Conclu | usions | 53 |
| 6 | Steg | analysi | s: remarks and future works | 55 |
| II | MI | PSteg-c | color: a new steganographic technique | 57 |
| 7 | Steg | anogra | phy at higher semantic level | 59 |
| | 7.1 | Introdu | uction to MP image decomposition | 60 |
| | 7.2 | Embec | Iding a message in the MP domain | 62 |

Contents

| 8 | An MP tailored for steganographic application | | |
|-----|---|---|-----|
| | 8.1 | Dictionary | 65 |
| | 8.2 | MP selection and update rules | 66 |
| 9 | A clo | oser look at the new MP domain | 73 |
| 10 | MPS | Steg-color | 77 |
| | 10.1 | Embedding Rule | 79 |
| | 10.2 | Improving undetectability | 80 |
| | 10.3 | Increasing the payload | 81 |
| 11 | MPS | Steg-color: experimental results | 83 |
| | 11.1 | Image Database | 83 |
| | 11.2 | Effectiveness of the proposed MP decomposition | 84 |
| | | 11.2.1 Interband correlation of decomposition path | 84 |
| | | 11.2.2 Effectiveness of the decomposition refinement step | 84 |
| | 11.3 | Undetectability analysis | 85 |
| | | 11.3.1 Targeted steganalyzers | 85 |
| | | 11.3.2 State-of-art steganalyzers | 87 |
| | | 11.3.3 Steganalysis Results | 88 |
| | | 11.3.4 Computational Complexity | 92 |
| 12 | MPS | Steg-color: remarks and future works | 99 |
| 13 | Fina | l remarks | 101 |
| Bil | bliogr | aphy | 105 |

List of Figures

| 1.1 | Relationship between steganography and related fields | 2 |
|-----|--|----|
| 2.1 | Example Receiver Operating Characteristic (ROC) curve | 18 |
| 2.2 | k = 5 individual ROC curves | 18 |
| 2.3 | Vertical averaging. | 19 |
| 2.4 | Threshold averaging. | 19 |
| 4.1 | Analysis of the impact of the border effect described in Subsection 4.1.1 on classification results. | 35 |
| 4.2 | Analysis of the impact of ALE features selection on classification results. | 36 |
| 5.1 | Impact of the source of imagery on classification performances. The embedding rate has been fixed both during the training and testing phase and set equal to 0.5 bpp. The label of the rows indicates the database used for training while the label of the columns represents | |
| | the dataset used during the testing phase | 43 |

| 5.2 | Impact of the embedding rate on classification performances. The source of imagery during both the training and testing phase is the Combined dataset. The label of the rows indicates the embedding rate used for training while the label of the columns represents the embedding rate used during the testing phase. | 47 |
|------|---|----|
| 5 2 | Charles of the set of | 47 |
| 5.3 | Classification performances when prior information about the source | |
| | to imagery is available. Training is done with stego contents ob- | |
| | and 1 hpp. The label of the rows indicates the course of imagene | |
| | and 1 opp. The label of the rows indicates the source of imagery | |
| | of the columns represents the embedding rate used during the testing | |
| | blue columns represents the embedding rate used during the testing | 50 |
| 5 4 | | 50 |
| 5.4 | Classification performances when no prior information about the | |
| | source of imagery is available. Training is done with the Com- | |
| | bined database and stego contents obtained with embedding rates | |
| | uniformly distributed across 0.2, 0.5 and 1 bpp. The label of the | |
| | rows indicates the source of imagery used during testing, while the | |
| | the testing phase | 50 |
| | | 32 |
| 8.1 | A subset of the atoms the dictionary consists of | 66 |
| 8.2 | The Selection Rule. | 68 |
| | | |
| 9.1 | Comparison between the compaction property of the DCT and MP | |
| | domains | 74 |
| 9.2 | Original gray-scale image. | 74 |
| 9.3 | Reconstructed image by using the first 2 DCT coefficients in a zig- | |
| | zag ordering for each 4×4 block. | 75 |
| 9.4 | Reconstructed image by using 2 atoms for each 4×4 block | 75 |
| | | |
| 11.1 | For each block the numbers $Z' = A + D - B - C $ and $Z'' =$ | |
| | E + H - F - G are computed | 86 |

List of Figures

| 11.2 0 | Comparison between coefficients histogram of a cover image (dashed | |
|---------------|---|----|
| li | ine) and a stego MPSteg-color image (solid line) | 87 |
| 11.3 C | Comparison between BD (a) and MPHA (b) steganalyzers: the first | |
| v | window size reported in the figure indicates the window size used by | |
| t | he steganalyzer, the second one the size of the blocks used by the | |
| e | embedder to partition the image. The payload is 0.3159 bpp | 89 |
| 11.4 F | Perceptual invisibility of the stego-message. The cover (a) and the | |
| S | stego (b) images can not be distinguished (payload = 0.3158 bpp, | |
| 4 | 4×4 partition, 51.40dB) | 94 |
| 11.5 C | Comparison between MPSteg-color with window 4×4 (solid line) | |
| а | and ± 1 embedding (dashed line) with 3 different steganalyzers at | |
| 0 | 0.3159 bpp of payload | 95 |
| 11.6 C | Comparison between MPSteg-color with window 5×5 (solid line) | |
| а | and ± 1 embedding (dashed line) with 3 different steganalyzers at | |
| 0 | 0.2002 bpp of payload | 95 |
| 11.7 C | Comparison between MPSteg-color with window 6×6 (solid line) | |
| а | and ± 1 embedding (dashed line) with 3 different steganalyzers at | |
| 0 | 0.1391 bpp of payload | 96 |
| 11.8 N | MPSteg-color detection performance on ALE at fixed embedding | |
| r | rate (0.1391 bpp) | 96 |
| 11.9 N | MPSteg-color detection performance on WAM at fixed embedding | |
| r | rate (0.1391 bpp) | 97 |
| 11.10N | MPSteg-color detection performance on 2D-HCFC at fixed embed- | |
| d | ling rate (0.1391 bpp). | 97 |

List of Tables

| 2.1 | Binary classification outcomes | 16 |
|------|---|----|
| 4.1 | Table of ALE features | 33 |
| 11.1 | PSNR values (dB) for different payload and settings. | 90 |
| 11.2 | Average execution time in seconds of embedding phases for images | |
| | 512×512 of size, window 4×4 and full payload (0.32 bpp) | 92 |
| | | |

Acknowledgements

I have a lot of acknowledgements to do for this thesis, specially because if I'm arrived to be who I am is thanks to all the people that have been around me, from those close to my desktop to those which fill up my free time.

First of all, I would like to thank my Ph.D. supervisor Mauro Barni, for his support, for his guidance and constructive criticism during these three years and half of my Ph.D.

Special thanks are due to Dr. Gwenaël Doërr and Prof. Ingemar J. Cox, who, in 2007, kindly received me in the Adastral Park Postgraduate Campus at University College London through the European Exchange Program Erasmus, for their attention and enlightening discussions. I would like to acknowledge the review efforts from Dr. Gwenaël Doërr for his precious comments on the initial manuscript of this thesis which have enabled to significantly enhance its clarity and quality and from Dr. Andreas Westfeld for his appreciation of my work.

Next, I would like to thank all the people who have been involved more or less close to my work. I want to thanks my colleague Angela, especially for her patience during our animate discussions, Guido for his generosity, and Sara for the Thursday curry dinners in the nicest Ipswich pub during my Erasmus period. Moreover, I cannot discard Riccardo, Pierluigi and Fabio for attending me for the coffe break and for any kind of break too. I also thankful to all students that during this period have enjoyed my work with their wired and uncomprehensible questions about their thesis.

I also thank all my friends who have helped me to relax during my free time and I apologize to everyone for having neglected them by spending several weekends and holidays at work: you weren't less important than my work! Moreover I appreciate my Bands, Siena and Ipswich, because they have been the melody of my studies and I thanks my Contrada which underlined this period by winning a Palio.

At the end, really special thanks are due to my brother Matteo, my dad Fabrizio, and my mum Loredana for their immeasurable support during ups and down of my life and the Ph.D. award is mainly due to their help.

I'm almost sure that I'm missing someone so... thanks to all the people whose love me too!

Chapter 1

Introduction

Steganography is the art of invisible communication. The term invisible is not linked to the meaning of the communication, as in cryptography in which the goal is to secure communications from an eavesdropper, on the contrary it refers to hiding the existence of the communication channel itself. The general idea of hiding messages in common digital contents, interests a wider class of applications that go beyond steganography. The techniques involved in such applications are collectively referred to as information hiding [1]. For example, while it is possible to add metadata about an image in special tags (exif in JPEG standard) or file headers, this information will be lost when the image is printed, because metadata inserted in tags on headers are tied to the image only as long as the image exists in digital form and are lost as soon as the image is printed. By using information hiding techniques, it is possible to fuse the digital content within the image signal regardless of the file format and the status of the image (digital or analog).

In this thesis we will refer to cover Work or equivalently to cover image, or simply cover to indicate the images that do not yet contain a secret message, while we will refer to stego Work, or stego images, or stego object to indicate an image with an embedded secret message. Moreover, we will refer to the secret message as stego-message or hidden message.

Depending on the meaning and goal of the embedded metadata, several information hiding fields can be defined, even though in literature the term 'information hiding' is often used as a synonym for steganography. In digital watermarking, for instance, the information is used for copy prevention, copy control, and copyright protection. In this case the embedded data should be robust to malicious attacks in order to preserve its goal.



Figure 1.1: Relationship between steganography and related fields.

The key difference between steganography and watermarking is the absence (in steganography) of an active adversary mainly because usually no value is associated with the act of removing the information hidden in the host content. Nevertheless, steganography may need to be robust against accidental or common distortion like compressions or color adjustment (in this case we will talk about active steganography).

On the other side, steganography wish to communicate in a completely undetectable manner which does not need to be required in watermarking. For this reason we can consider steganography also as part of cover communication science. Figure 1.1 graphically shows connections between steganography and related fields. The intersection between steganography and watermarking comprises active steganography and some kinds of watermarking for authentication applications.

From an Information Theory perspective, we can introduce steganography by adopting a slightly different point of view [2]. In [3] Shannon was the first that considered secrecy systems from the viewpoint of information theory. Shannon identified three types of secret communications which he described as

- 1. 'concealment systems, including such methods as invisible ink, concealing a message in an innocent text, or in a fake covering cryptogram, or other methods in which the existence of the message is concealed from the enemy',
- 2. privacy systems,

3. cryptographic systems.

With regards to concealment systems, i.e. steganography, Shannon stated that such *'systems are primarily a psychological problem'* and did not consider them further.

Afterwards the concept of steganography was recovered by Simmons [4] in his famous explanation of steganography described by mean of the prisoners' problem. According to the prisoners' scenario two accomplices in a crime have been arrested and are about to be locked in widely separated cells. Their only means of communication after they are locked up is by way of messages conveyed for them by trustees - who are known to be agents of the warden. The warden is willing to allow the prisoners to exchange messages. However, since he has every reason to suspect that the prisoners want to coordinate an escape plan, the warden will only permit the exchanges to occur if the information contained in the messages is completely open to him and presumably innocuous. The prisoners, on the other hand, are willing to accept some risk of deception in order to be able to communicate at all, since they need to coordinate their plans. To do this they have to deceive the warden by finding a way of communicating secretely in the exchanges, i.e., of establishing an hidden channel between them in full view of the warden, even though the message themselves contain no secret (to the warden) information.

Today steganography is also seen as a way of ensuring freedom of speech in military dictatorship countries or connected to homeland security. Steganography has also been supposed to be used by terrorists to design terroristic attacks. Example about the terrorism are the technical jihad manual [5] that is part of a terrorist manual and the color of the Osama Bin Laden's beard in its clips: military investigators think that secret messages are associated each color of the beard to coordinate terrorist cells.

Another topical target of steganography is computer warfare. New worms and spywares stole a lot of information about users and then they have to find a way to carry out this data by preventing any suspicion of transmission existence by antivirus, firewall or data stream analysis.

From a different viewpoint, we sometimes know that there are some forbidden transmissions [6] and we want to know who is sending secret information, for ex-

ample, to the press. Apparently, during the 1980's, British Prime Minister Margaret Thatcher became so irritated at press leaks of cabinet documents that she had the word processors programmed to encode the identity of secretaries in the word spacing of documents, so that disloyal ministers could be traced. Later, steganography has being used by some HP and Xerox printers [7] which embed small yellow dots during the printing phase, by writing a coded message in which the serial number of the printer and the print time is embedded. This security has been initially forced onto printer manufacturers by the Federal Government because American dollar bills were easily forged with such printers (one of the weakest currency at the time).

During the last few years image steganography research has raised an increasingly interest. A variety of techniques have been proposed especially for a given image file format like gif, jpeg or images represented in the pixel domain. In fact, the main idea behind steganography undetectability is: less embedding changes to the cover Work means a less detectable stego object. Even though this statement is not completely true (as it shown in [8]), it represents a good starting point to develop and to improve initial steganographic techniques proposed in the literature. Moreover, new channel coding techniques have been proposed to reduce the embedding changes as the introduction of matrix embedding [9, 10] and Wet Paper Coding [11]. Other techniques [12, 13], specially in JPEG domain, use a subset of support to adjust in some way image statistics that are changed by the message embedding. Recently in [14] authors try to estimate the payload upperbound for a perfect undetectability by using common JPEG steganalysis.

The dual goal of steganography pertains to steganalysis whose goal is to discover the presence of secret communication channels (secret messages) established by steganography. For each steganographic method, several techniques (i.e. *target steganalysis*) [15, 16, 17, 18, 19] have been proposed, however the current state of art is moving to *blind steganalysis* [20, 21, 22, 15], i.e. techniques that are designed to detect the widest possible range of steganography.

Modern steganalyzers summarize the image by a set of features which are able to reveal the presence or the absence of a secret message embedded within the Work, then these features are used to train a classifier like a Linear Discriminant classifier or a Support Vector Machine. After the training phase, the whole system based on a feature extraction and a classification step is ready to use. This feature summarization is highly dependent on the image itself, so it depends on image source and hence pre-embedding processing and experimental settings of a technique should be carefully described. The high dependence between steganalysis and images used in experimental results can be explained by the follow considerations. Some steganalyzers which work on high order statistics are highly dependent on high support frequencies, but these frequencies change a lot depending on image source (camera CCD, or scanner CCD) and the presence of lossy compression, i.e. a low pass filtering, that can be applied to the image before the potential steganography [23].

The detectability of a hidden message highly depends on the payload, i.e. the ratio between the length of the secret message and the size of the cover in which it is embedded. In a real case we should consider that no a priori information is given about the message length that could be embedded within the analyzed Work. Moreover, in [24, 25], authors show that the detectability of a stego image is linked to square root ratio between the payload and the image size.

When a new steganalyzer is proposed, all the above issues should be take into account. Moreover, authors should share all their experimental settings, including the image database used for the test, to permit to validate and to make their work reproducible. Unfortunately, steganographic literature usually lacks good comparisons and reproducible research, so in this thesis we tried to adopt a fully reproducible methodology applied both to steganography and steganalysis. In the next section, a detailed description of the main contributions of the thesis is given.

1.1 Contributions of the thesis

The contribution of this thesis is threefold. From a steganalysis point of view we introduce a new steganalysis method called ALE^1 which outperforms previously proposed pixel domain method. As a second contribution we introduce a comparative methodology for the comparison of different steganalyzers and we apply it

¹Amplitude of Local Extrema

to compare ALE with the state-of-art steganalyzers. The third contribution of the thesis regards steganography, since we introduce a new embedding domain and a corresponding method, called MPSteg-color, which outperforms, in terms of unde-tectability, classical embedding methods. Next, we briefly describe each contribution.

1.1.1 ALE

Recently Zhang *et al.* [26] have introduced an algorithm for the detection of ± 1 LSB steganography in the pixel domain based on the statistics of the amplitudes of local extrema in the grey-level histogram. Experimental results demonstrated performance comparable or superior to other state-of-the-art algorithms. In this thesis, we describe improvements to Zhang's algorithm (i) to reduce the noise associated with border effects in the histogram, and (ii) to extend the analysis to amplitude of local extrema in the 2D adjacency histogram.

Experimental results on a composite database of 7125 images, averaged over a 20-fold cross validation, with classification based on Fisher linear discriminants, demonstrated that the improved algorithm exhibits significantly better performance for the given dataset. The new algorithm, called ALE, uses 10 features derived in a very efficient way from the 1D and 2D histograms, so it is also executable in a real scenario in which the steganalysis results have to be given in realtime.

1.1.2 Comparative Methodology in Steganalysis

As a second contribution we discuss a variety of issues associated with comparison of different steganalyzers and highlight some of these issues with a case study comparing four steganalysis algorithms designed to detect ± 1 embedding. In particular, we discuss issues related to the creation of the training and testing sets. We emphasize that for steganalysis, it is very unlikely that the assumptions used to create the training set will match conditions used during deployment. Consequently, it is imperative that testing also investigates how performance degrades as the test set deviates from the training data. The subsequent empirical evaluation of four algorithms on four different test sets revealed that algorithm performance is highly variable, and strongly dependent on the training and test imagery. Experimental results clearly demonstrate that the performance is strongly image-dependent, and that further work is needed to establish more comprehensive databases. It is also common to assume that the embedding rate is known during testing and training, but this is unlikely to be the case in practice. Once again, significant performance degradation is observed. Experimental results also suggest that the common practice of training at a low embedding rate in order to deal with a wide range of embedding rates during testing is not as effective as training with a mixture of embedding rates.

1.1.3 MPSteg-color

The third contribution regards steganography for color images. Specifically, we propose a new steganographic method that tries to use the fail-safe of steganalyzers to improve the undetectability of the stego-message. In fact, although steganalyzers do not know the hidden message, they rely on a statistical analysis to understand whether a given signal contains hidden data or not. However this analysis disregards the semantic content of the cover signal. We argue that, from a steganographic point of view it is preferable to embed the secret message at higher semantic levels of the image, e.g. by modifying structural elements of the cover image like lines, edges or flat areas.

By the above consideration, we propose a new steganographic method, called MPSteg-color, that hides the stego-message into some selected coefficients obtained through a high redundant basis decomposition of the color image. The decomposition is efficiently obtained by using a Matching Pursuit (MP) algorithm. In this way the hidden message is embedded at a higher semantic level and hence it is more difficult for a steganalyzer to detect it.

1.2 Thesis organization

This thesis is organized in two parts regarding steganalysis and steganography in the pixel domain. The first part deals with steganalysis by introducing it as classification problem in Chapter 2 and by showing the state-of-art of steganalysis in the pixel domain in Chapter 3. Moreover, in Chapter 3 we describe a simple steganography benchmark called ± 1 embedding. In Chapter 4 we propose a new steganalyzer, called ALE, which improves the ± 1 embedding detection especially for images with high frequency noise in the histogram. Chapter 5 investigates experimental issue in steganalysis by proposing a methodology to fully compare steganalyzer performances. In the same chapter, we also compare the ALE steganalyzer with other three state-of-art steganalyzers. Some considerations and future works are drawn in Chapter 6.

In Part II we develop a new steganography which is less detectable than ± 1 steganography. To do so we embed the message at a higher semantic level with respect to the pixel domain by using the high redundant basis domain described in Chapter 7. Due to the impossibility to use the MP algorithm as it is used in image compression, we define an MP suitable approach for steganalysis in Chapter 8 and we fully describe the proposed technique, MPSteg-color, in Chapter 10. The undetectability of MPSteg-color is investigated in Chapter 11 both against target and general purpose steganalyzers. Chapter 12 presents some conclusions and future works on MPSteg-color.

Part I

± 1 embedding steganalysis

Chapter 2

Steganalysis: a classification problem

In this part of the thesis we will consider the steganalysis of ± 1 embedding technique by introducing some steganalysis concepts and by describing the steganalyzers that are available in literature. Moreover we propose a new steganalyzer and we compare it with the state-of-art steganalyzers in the pixel domain. While performing this comparison we also describe a full benchmark methodology.

A steganalysis algorithm receives a Work and must decide whether it is a cover or stego Work. Some steganalysis algorithms go further, attempting to estimate the size of the embedded message and even the content of the message. In this thesis, we are only concerned with the first decision step, and as such, we view steganalysis as a binary classification problem, i.e. the Work is, or is not a stego Work.

Classification has a long history and we assume that the reader is familiar with the basics of classification. It is not our intention to provide a detailed tutorial on the subject of classification and the reader is directed to [27] for further information.

Blind steganalysis refers to algorithms that do not assume knowledge of the underlying steganographic algorithm [28]. As such, these algorithms are intended to detect the presence of a hidden message embedded with a wide variety of algorithms, perhaps including unknown algorithms. Conversely, *targeted* steganalysis assumes knowledge of the underlying steganographic algorithm, and as such, is intended for the detection of a specific steganographic algorithm [28]. In this thesis, we are concerned with targeted steganalysis, specifically the detection of ± 1 embedding.

As said, steganalysis is a classification problem, hence, building a steganalyzer can be viewed as a three step procedure:

1. For each image in a training set containing both cover and stego Works, extract a feature vector,

- 2. With the available training feature vectors, train a binary classifier for the classification of stego and non-stego Works,
- 3. Vary the decision parameters of the classifier, e.g. a threshold, to obtain the receiver operating characteristic (ROC) curve for the training data and set the value of this parameter to achieve the desired performance in terms of false positive or true positives.

Most steganalysis algorithms can be described by (i) their feature set, and (ii) the associated classification algorithm. The feature set is often handcrafted, and may be derived from an analysis of one or more steganographic algorithms. In this Chapter, we assume that the feature set is given and focus our attention on general issues related to classification, while the problem of define a significant set of features will be addressed in the next chapter. We do not consider the relative merits of various classification algorithms, e.g. *k*-nearest neighbors (*k*-NN), Fisher linear discriminant (FLD) analysis, support vector machines (SVM), etc. Instead, we consider generic issues that are applicable to all classification algorithms. Specifically, we consider two phases in the design of a classification system, namely the training phase and the test phase. We now consider each in turn.

2.1 Training

During the training phase, the classification algorithm is presented with a set of labeled data, i.e. images that are known to be either stego Works or cover Works. The classification algorithm uses this information to adjust its associated parameters in order to minimize the number of false positives and false negatives it classifies.

In steganalysis, a false positive corresponds to classifying a cover Work as a stego Work. Similarly, a false negative corresponds to classifying a stego Work as a cover Work. Both errors are important, but the relative cost of each error may depend on the application. For example, if steganalysis is applied to the detection of covert terrorist communication, a false negative may be more costly than a false positive. Such an application may therefore accept a higher false positive rate, in order to ensure a lower false negative rate. Of course, resources must then be available to analyze the data classified as stego Works, and more resources will be needed because of the higher level of false positive. If resources are severely constrained, as for example may be the case for police surveillance of hidden child pornography¹, then a different compromise may be sought that seeks to reduce the number of false positives, even though this will be at the expense of increasing the number of false negatives, i.e. failing to detect actual cases.

Labeled examples of both cover images and stego images are needed. Cover images are in abundance. They are available from cameras, the Internet and standardized databases. However, in order for experimental results to be reproducible, the dataset must be publicly available. And for the experimental results to be comparable, it is necessary to use the same database for various algorithms, otherwise variations in performance may be attributable to variations in the database rather than in the algorithm. The steganalysis community has recognized this and a number of databases have become de facto standards for experimentation. These databases are described in Chapter 5.

The type of imagery contained in these databases varies considerably. It is derived from a variety of sources, i.e. cameras, outdoor scenes, indoor scenes, etc, and is stored in a variety of different formats, i.e. images may have never been compressed or have been compressed using a number of lossy compression algorithms that introduce a variety of statistical artifacts. The effect of these variations has not been discussed in detail. However, experimental results described in Chapter 5 clearly indicate that the performance of a single algorithm can vary greatly, depending on the database.

Since performance is so affected by the database, it is imperative to (i) characterize each database and understand what characteristics affect performance, (ii) test on multiple standardized databases in order to quantify the variation in performance due to the dataset, and (iii) develop new databases that contain a wider variety of training imagery.

¹Note that while child pornography is often cited as an application for steganalysis, we are unaware of any documented case of this. To the best of our knowledge, the closest case is the "twirl face" pedophile in Thailand [29] which is a long shot away from any kind of steganography.

For targeted steganalysis, the labeled stego images are usually generated from the cover images by applying the known steganographic algorithm to the cover images. For blind steganalysis, a set of known steganographic algorithms can be used to generate a labeled training set. In this case, the hope is that the resulting classifier will at least learn to classify stego Works generated by this set of algorithms. And perhaps will even generalize to previously unseen algorithms. Alternatively, one can try to devise a model of cover content and detect whenever the content under test deviates from this model [30].

Even in the case of target steganalysis, generation of the labeled set is not straightforward. In particular, every steganographic algorithm will have a variety of parameter setting. What values should be used to generate the stego images? There is no definitive answer to this question. Rather, it depends on the particular application scenario. In an ideal situation, the steganalyst would have information about the parameter settings used by the adversary. However, such a scenario is very unlikely. In the absence of this knowledge, it is necessary to deal with all possibilities.

Let us consider the *embedding rate*, which is a parameter common to all steganographic algorithms. The embedding rate, also referred to as the relative message length, is the ratio of the covert message length (in bits) to the number of samples in the cover Work. It is well-known that the lower the embedding rate, the more difficult it is to reliably detect a stego Work. Despite the fact that the embedding rate is unknown and also likely to vary, it is common to train using a single embedding rate (and to test with the same). Clearly this represents a best-case scenario that is unlikely to be achieved in practice. However, if sufficient resources are available, then it may be possible to run multiple steganalysis algorithms, each trained for a specific set of parameter settings. If the number of parameters is small, this may be practical. If not, then it is necessary to train (and test) using a range of parameter settings².

²This issue is examined further in Chapter 5.

2.2 Testing

Once the training phase is complete, the classification system must be tested. Clearly, the test data must be different from the training data. After all, when the steganalysis system is deployed, it will be analyzing previously unseen data. We therefore need to be confident that the system does not suffer from over-learning. Testing on the training set does not provide us with this confidence (surprisingly, a number of papers on steganalysis do not follow this rule and classification rates sometimes are only reported on the training data).

2.2.1 Cross validation

A database of images must be divided into both a training and a test set. Ideally, this partitioning should be made by randomly assigning images to one or other of the two sets, in order to avoid any bias. The size of the two sets does not need to be equal. To simulate real world conditions, it may be desirable to have a much smaller training set to account for the fact that there is much more content available worldwide than any database being used in a lab. Of course, this may introduce strong performance variations depending on the content selected for training. To address this problem, it is a common practice to repeat the training and testing multiple times. This is referred to as k-fold cross validation. One can then assess the stability of the steganalysis system by analyzing the detection performances statistics.

2.2.2 Performance measures

There are a number of performance measures that are of interest in steganalysis. The most common measures are the false positive and false negative rates. Since these two measures are intimately coupled, it is also common to depict these rates in the form of a receiver operating characteristic (ROC) curve. A limitation of such measures is that they do not provide a single numerical figure of merit. To address this, the area under the ROC curve is occasionally used as such.

| | | True Class | |
|----------------|---|------------------------|-----------------------|
| | | р | n |
| Hypothesized | р | true positives (TP) | false positives (FP) |
| Class | n | false negatives (FN) | true negatives (TN) |
| Column totals: | | Р | N |

Table 2.1: Binary classification outcomes.

False positives and negatives

The steganalysis problem is a binary classification problem - is or isn't the test instance (image) a stego image? As such, there are four possible outcomes, which are illustrated in Table 2.1. These are:

- 1. True positives, i.e. test instances that are correctly labeled as stego Works;
- 2. True negatives, i.e. test instances that are correctly labeled as non-stego Works;
- 3. False negatives, i.e. test instances that are incorrectly labeled as non-stego Works;
- 4. False positives, i.e. test instances that are incorrectly labeled as stego Works.

If P and N denote the real number of positive and negative instances, and TP and FP denote the predicted number of true positives and false positives, respectively, then the true positive rate, t_p is defined as

$$t_p = \frac{TP}{P},\tag{2.1}$$

and the false positive rate, f_p as:

$$f_p = \frac{FP}{N}.$$
(2.2)

Common performance metrics which can be derived from these include precision, recall, accuracy and F-measure:

$$Precision = \frac{TP}{TP + FP},$$
(2.3)

$$\operatorname{Recall} = \frac{TP}{P}, \tag{2.4}$$

Accuracy =
$$\frac{TP + TN}{P + N}$$
, (2.5)

$$F - measure = \frac{2}{1/precision + 1/recall}$$
 (2.6)

Receiver Operating Characteristic

The four classification outcomes, true and false positives, and true and false negatives, are coupled. For example, it is trivial to achieve a true positive rate of 100% by labeling all test instances as positive. Of course, this is at the cost of a 100% false positive rate. To better understand this coupled relationship, the receiver operating characteristic (ROC) curve plots the true positive rate against false positive rate. A typical ROC curve is illustrated in Figure 2.1.

A detailed discussion of the receiver operating characteristic can be found in [31]. A brief summary of some key points are now provided.

In a real scenario, a given classifier produces a single point on a ROC curve. However, all classifiers have some form of implicit or explicit decision threshold, and by varying this threshold it is possible to generate a full ROC curve. Random guessing will produce points along the diagonal line. A curve below the diagonal implies that simply inverting the binary decision would give a better classifier.

When k-fold cross validation is performed, we essentially have k such ROC curves, which we must merge in some way. There are a number of ways in which this can be done.

The most straightforward way is to merge the results for the k-trials into one single "trial" and plot the associated ROC curve as before. A limitation of this procedure is that it does not provide an associated variance measure for each point.

Given the k-trials, we have k corresponding ROC curves. If we consider the



Figure 2.1: Example Receiver Operating Characteristic (ROC) curve.



Figure 2.2: k = 5 individual ROC curves.



Figure 2.3: Vertical averaging.



Figure 2.4: Threshold averaging.

x-axis, i.e. the false positive rate, as an independent parameter that is under our control, then for a given fixed false positive rate, we can average the true positive rates, as depicted in Figure 2.3. The vertical lines at each point depict the uncertainty associated with the average. The length of the line can represent a percentile range, or the minimum and maximum values of the true positive rate for the given false positive rate. In this thesis, we show minimum and maximum values.

In practice, the false positive rate is not directly under our control, but rather is a function of a threshold, t, that controls both the true and false positive rates. Thus, for a fixed threshold, t, we can determine both the true and false positive rates for each of the k ROC curves and average these together, as depicted in Figure 2.4. Now the uncertainty associated with each point is two-dimensional, reflecting the variation in both the true and false positive rates for each of the k curves.

Area under the ROC curve

It is sometimes desirable to have a single scalar value to describe the performance of an algorithm. One method for doing so is to calculate the area under the ROC curve, (AUC). The AUC has a value form 0 to 1, but since the diagonal line, reflecting random performance, has an area of 0.5, the AUC typically ranges from 0.5 to 1. Fawcett [31] points out that (i) the AUC measures "the probability that the classifier will rank a randomly chosen positive instance higher than a randomly chosen negative instance", and (ii) it is closely related to the Gini coefficient [32].

2.3 Fisher Linear Discriminant Analysis

In this thesis we focus the attention on steganalyzer features, instead of taking into account the classifier. For this reason we decided to use a linear classifier. Even though we can obtain better results with Support Vector Machines (SVM) or other classifiers (which have a lot of settings), we prefer to give to the reader a fully reproducible approach.

Fisher Linear Discriminant (FLD) analysis seeks directions that are efficient for discrimination. The goal is to find an orientation **u** for which the samples in the

dataset, once projected onto it, are well separated. Let us assume that a dataset \mathcal{D} is made of N d-dimensional samples $\mathbf{x}_1, \ldots, \mathbf{x}_N$, N_1 being in a subset \mathcal{D}_1 corresponding to one class and N_2 being in a subset \mathcal{D}_2 corresponding to the other class. The first step of FLD analysis consists in computing the d-dimensional sample mean of each class:

$$\mathbf{m}_i = \frac{1}{N_i} \sum_{\mathbf{x} \in \mathcal{D}_i} \mathbf{x}.$$
 (2.7)

Next, the scatter matrix $S_W = S_1 + S_2$ is computed using the following definitions:

$$\mathbf{S}_{i} = \sum_{\mathbf{x} \in \mathcal{D}_{i}} (\mathbf{x} - \mathbf{m}_{i}) (\mathbf{x} - \mathbf{m}_{i})^{t}.$$
(2.8)

Finally, the direction of projection **u** is given by:

$$\mathbf{u} = \mathbf{S}_W^{-1}(\mathbf{m}_1 - \mathbf{m}_2). \tag{2.9}$$

This vector **u** defines a linear function $y = \mathbf{u}^t \mathbf{x}$ which yields the maximum ratio of between-class scatter to within-class scatter. The interested reader is redirected to [27] for further details (pp. 117–121).
Chapter 3

± 1 embedding: state of art

In this chapter we describe the scenario this thesis is working on. Mainly we introduce a common steganographic algorithm known as ± 1 embedding, also called LSB matching, which is a common used technique to embed messages in the pixel domain. Due to its simplicity, its efficiency, and its undetectability, ± 1 embedding is often used as a benchmark for steganalysis and steganography. This simple evolution from classical LSB is highly undetectable specially when the length of the embedded message is smaller than the length of the embedding support.

We also introduce two state of art steganalyzers, by describing their feature extraction method. The first one is a blind method, while the second steganalyzer is a simple feature steganalyzer developed by analyzing artifacts specific to ± 1 embedding.

3.1 ± 1 embedding steganography

The simplest technique used in steganography is the Least Significative Bit (LSB) also called LSB replacement. To illustrate LSB replacement, let us consider grayscale images with pixels values in the range $0 \dots 255$ as cover Works. LSB steganography replaces the least significant bit of each pixel value in the image with the corresponding bit of the message to be hidden. When LSB flipping is used, an even-valued pixel will either retain its value or be incremented by one. However, it will never be decremented. The converse is true for odd-valued pixels. This asymmetry introduces a statistical anomaly into the intensity histogram – pairs of intensity values, specifically 0-1, 2-3 etc., will, on average, exhibit the same frequency if the image is a stego Work. This can be exploited for steganalysis purposes, as described in [33, 34, 35].

LSB matching, also known as ± 1 embedding is a slightly more sophisticated version of least significant bit (LSB) embedding. Rather than simply replacing the LSB with the desired message bit, the corresponding pixel value is randomly incremented or decremented whenever the LSB value needs to be changed¹. By so doing, the asymmetry present in LSB flipping is almost eliminated². Luckily for the steganalyzer, other statistical anomalies are created that still permit discrimination between cover and stego Works. However, these anomalies are more subtle and discrimination accuracy is significantly lower than for LSB embedding.

In formulas, ± 1 embedding can be described as follows:

$$p_{s} = \begin{cases} p_{c} + 1, & \text{if } b \neq \text{LSB}(p_{c}) \text{ and } (\kappa > 0 \text{ or } p_{c} = 0) \\ p_{c} - 1, & \text{if } b \neq \text{LSB}(p_{c}) \text{ and } (\kappa < 0 \text{ or } p_{c} = 255) \\ p_{c}, & \text{if } b = \text{LSB}(p_{c}) \end{cases}$$
(3.1)

where κ is an i.i.d. random variable with uniform distribution in $\{-1, +1\}$, and p_c and p_s are respectively the pixel value of the cover and the pixel value of the stego image. This process can be applied to all the pixels in the image or only for a pseudo-randomly chosen image portion, when the embedding rate, ρ , is less than one, i.e. the length of the hidden message is less than the number of pixels in the image.

3.2 ± 1 embedding steganalyzers

The next sections describe a blind and a target steganalyzer which are the state of art of steganalysis in the pixel domain.

3.2.1 High Order Statistics of the Stego Noise (WAM)

Since ± 1 embedding is simply a matter of adding or subtracting 1 to a subset of pixel values, it can be modeled as the addition of high frequency noise. In [10],

¹Note that this strategy may affect bit-planes other than the LSB plane. For example, if the secret bit is a "0", and the original 8-bit pixel value is 01111111, then incrementing this value results in 10000000.

²The ± 1 embedding has asymmetries only for 0 and 255 pixel values in which no random choice can be applied due the lowerbound and upperbound borders.

Goljan *et al.* suggested estimating the stego noise and characterizing it with some central absolute moments. While their algorithm is a blind steganalysis algorithm, i.e. it is not designed to specifically detect ± 1 embedding, it seems well suited to do so.

The algorithm starts by computing the first level wavelet decomposition of the input image with the 8-tap Daubechies filter. The resulting three frequency subbands (vertical \mathbf{v} , horizontal \mathbf{h} , and diagonal \mathbf{d}) are then denoised with a Wiener filter, as follows:

$$\mathbf{b}_{den}(i,j) = \frac{\hat{\sigma}_{\mathbf{b}}^2(i,j)}{\hat{\sigma}_{\mathbf{b}}^2(i,j) + \sigma_0^2} \mathbf{b}(i,j), \quad (i,j) \in \mathcal{I}$$
(3.2)

where **b** is one of the three subbands, \mathcal{I} is a bidimensional index set used to run through the whole subband, and $\sigma_0^2 = 0.5$. The local variance, $\hat{\sigma}_{\mathbf{b}}^2(i, j)$, at position (i, j) in the subband **b** is estimated by:

$$\hat{\sigma}_{\mathbf{b}}^{2}(i,j) = \min_{N \in \{3,5,7,9\}} \max\left(0, \frac{1}{N^{2}} \sum_{(i,j) \in \mathcal{N}_{i,j}^{N}} \mathbf{b}^{2}(i,j) - \sigma_{0}^{2}\right), \quad (3.3)$$

where $\mathcal{N}_{i,j}^N$ is the square $N \times N$ neighborhood centered at pixel location (i, j). The noise residual, $\mathbf{r}_{\mathbf{b}} = \mathbf{b} - \mathbf{b}_{den}$, is then computed, together with its first p absolute central moments. Specifically,

$$m_{\mathbf{b}}^{p} = \frac{1}{|\mathcal{I}|} \sum_{(i,j)\in\mathcal{I}} |\mathbf{r}_{\mathbf{b}}(i,j) - \bar{\mathbf{r}}_{\mathbf{b}}|^{p}, \qquad (3.4)$$

where $\bar{\mathbf{r}}_{\mathbf{b}}$ is the mean value of the estimated stego noise in subband b. The first 9 central moments, i.e. $p = 1 \cdots 9$, for each of the three subbands are calculated to obtain a 27-dimensional feature vector, \mathbf{f}_{WAM} , that is used for steganalysis:

$$\mathbf{f}_{\text{WAM}} = \left\{ m_{\mathbf{b}}^{p} \mid \mathbf{b} \in \{\mathbf{v}, \mathbf{h}, \mathbf{d}\}, \ p \in [1, 9] \right\}.$$
(3.5)

Due to its construction, this system is referred to as Wavelet Absolute Moment (WAM) steganalysis. Further details can be found in [10]. It should be noted that

this method is not specific to ± 1 steganography and can therefore be used to detect other steganographic techniques. Authors shows in [10] that by using a 0.5bpp of payload, WAM produces only 1.77% false positives at 50% of detection rate, and the AUC value is above 0.95.

Even though WAM algorithm provides a rather good classification accuracy, it has main three weaknesses. The first one is that it looks for a fingerprint of the steganography in the noisy region of the image. For a good detection, the ratio between the steganography fingerprint and the image noise should be high. The second one is that the feature vector has 27 elements, but for a given scenario (i.e. by analyzing images that come from a specific source and by using the same steganography with a fixed payload) only a subset of these are useful to detect stego image. Moreover, by changing the scenario, it changes the feature subset too. This behavior is not good when the steganalyzer works in a real scenario in which there is no knowledge about the images under analysis. The last one is the computational complexity for the feature extraction, i.e. a wavelet full frame decomposition and the calculation of several high order statistics on an huge amount of wavelet coefficients. When a steganalysis system have to work with a big image database or an Internet image streaming, it is onerous to apply a real time analysis by using WAM.

3.2.2 Center of Mass of the Histogram Characteristic Function (2D-HCFC)

In [36], Harmsen and Pearlman noted that ± 1 embedding steganography induces a low-pass filtering of the intensity/color histogram h_1 of the image³. They showed that, when looking at the intensity histogram, ± 1 steganography reduces to a filtering operation with the kernel:



where ρ is the embedding rate. This means that the histogram of a stego Work contains less high-frequency power than the histogram of the corresponding cover

³In this thesis, all histograms will be considered to be implicitly normalized by the total number of samples.

image. In other words, the Fourier transform H_1 of the intensity histogram, also referred to as the Histogram Characteristic Function (HCF), is likely to be significantly affected by ± 1 embedding steganography. In fact, its center of mass, defined as

$$c_1(\mathbf{H}_1) = \frac{\sum_{k=0}^{127} k \|\mathbf{H}_1(k)\|}{\sum_{k=0}^{127} \|\mathbf{H}_1(k)\|}$$
(3.6)

will be shifted toward the origin. In eq.(3.6) summations are from k = 0 to 127 to avoid the symmetric parts of the Fourier transform. This approach can be extended to multidimensional signals, e.g. RGB images, by using a multidimensional Fourier transform and computing a multidimensional center of mass. Experimental results [23] have shown that the HCF strategy performs better with RGB images than with grayscale images.

Ker [23] suggested that this difference in performance is due to a lack of sparsity in the histogram of grayscale images. To address this issue, Ker proposed using a two-dimensional adjacency histogram, $h_2(k, l)$, which tabulates how often each pixel intensity is observed next to another:

$$\mathbf{h}_{2}(k,l) = \left| \left\{ (i,j) \in \mathcal{I} \mid \mathbf{p}(i,j) = k, \ \mathbf{p}(i,j+1) = l \right\} \right|$$
(3.7)

where $\mathbf{p}(i, j)$ is the pixel value at location (i, j) in the input image, and \mathcal{I} is a bidimensional index set which runs through all pixel locations in the image. Since adjacent pixels have in general close intensity values, this histogram is sparse off the diagonal. ± 1 embedding steganography reduces to low-pass filtering the adjacency histogram with the following kernel:

| $\left(\frac{\rho}{4}\right)^2$ | $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ | $\left(\frac{\rho}{4}\right)^2$ |
|---|---|---|
| $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ | $\left(1-\frac{\rho}{2}\right)^2$ | $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ |
| $\left(\frac{\rho}{4}\right)^2$ | $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ | $\left(\frac{\rho}{4}\right)^2$ |

As a result, in the same way as in the 1D case, the center of mass of the 2-D histogram characteristic function, H_2 , obtained with a 2-D Fourier transform, is shifted toward the origin. However, to obtain a scalar feature, Ker suggested to use the center of

mass of the 2D-HCF projected onto the first diagonal:

$$c_{2}(\mathbf{H}_{2}) = \frac{\sum_{k=0}^{127} \sum_{l=0}^{127} (k+l) \|\mathbf{H}_{2}(k,l)\|}{\sum_{k=0}^{127} \sum_{l=0}^{127} \|\mathbf{H}_{2}(k,l)\|}.$$
(3.8)

This alternative feature has been reported to significantly outperform the center of mass calculated from a one-dimensional HCF [23], by decreasing from 34.8% to 7.8% the false positives at 50% of detection rate, by using a 0.5 bpp of payload.

Finally, to reduce the variability of this feature across images, Ker recommended applying a calibration procedure, so that the final feature vector, $\mathbf{f}_{2D-HCFC}$ is given by:

$$\mathbf{f}_{2\mathrm{D-HCFC}} = \frac{c_2(\mathbf{H}_2)}{c_2(\mathbf{H}_2')},\tag{3.9}$$

where \mathbf{H}_2' is the 2-D histogram characteristic function of a downsampled version of the image. The image is downsampled by a factor of 2 using a straightforward 2×2 averaging filter. Experimental results have demonstrated that this ratio is close to 1 for original cover Works and lower than 1 for stego Works, hence permitting efficient steganalysis. In contrast with the previous method, this steganalyzer, referred to as 2D-HCFC, is targeted for ± 1 steganography. Nothing suggests that it could be useful to detect other steganographic techniques.

The 2D-HCFC feature itself, in comparison with 27 features by WAM, is able to be used for a good stego-cover classification. Unfortunately, the big weakness is that it mainly works well on images which are compressed before the embedding phase. In this case, images have poor high frequency contents and the presence of the steganography fingerprint - an additional low pass filtering - can be discriminated easier then using never-compressed images.

By analyzing the above steganalysis, specially 2D-HCFC, and the ± 1 embedding artefacts, we developed a new target steganalyzer with a low complexity feature extraction algorithm. The proposed steganalyzer, based on the Amplitude of Local Extrema (ALE) is fully described in the next chapter. Moreover, in Chapter 5 we will compare the above steganalysis with the new one that we are proposing.

Chapter 4

Amplitude of Local Extrema

In this chapter, we describe a new steganalysis algorithm that significantly improves upon previous results. It is based on work by Zhang *et al.* and it works on the statistical properties of the amplitudes of local extrema (ALE). The extension to the algorithm presented in [26] is described in Section 4.1. Specifically, we first describe a modification to the algorithm that reduces noise associated with border effects, i.e. pixel values with intensities of either 0 or 255. Section 4.2 then describes the extension of the amplitudes of local extrema to 2D adjacency histograms. These enhancements result in a collection of 10 features whose classification performances are evaluated in Section 4.3 through experimental validation. The results clearly demonstrate significantly improved classification compared to the original steganalyzer by Zhang *et al.* [26]. Moreover in Section 4.4 we design a Hybrid steganalyzer that takes into account state-of-art and ALE steganalyzers. At the end of the chapter, in Section 4.5, some consideration are drawn.

4.1 Improving previous work on histogram domain

In [36], the authors noted that ± 1 embedding steganography induces a low-pass filtering of the intensity/colour histogram \mathbf{h}_1 of the image. Indeed, it is easy to show that, when looking at the intensity histogram, ± 1 steganography is equivalent to a filtering operation with the kernel:

| $\frac{\rho}{4}$ | $1 - \frac{\rho}{2}$ | $\frac{\rho}{4}$ |
|------------------|----------------------|------------------|
| | | |

where ρ is the embedding rate. This implies that the histogram of a stego Work contains less high-frequency power than the histogram of the corresponding cover image.

Based on this idea, Zhang *et al.* [26] proposed to observe what happens in the surrounding of local extrema of the histogram [26]. Since ± 1 embedding is equivalent to low pass filtering the intensity histogram, then the filtering operation will reduce the amplitude of local extrema (ALE). This motivated the introduction of a new feature, which is basically the sum of the amplitudes of local extrema in the intensity histogram, as defined below:

$$A_1(\mathbf{h}_1) = \sum_{n \in \mathcal{E}_1} \left| 2\mathbf{h}_1(k) - \mathbf{h}_1(k-1) - \mathbf{h}_1(k+1) \right|$$
(4.1)

where $\mathcal{E}_1 \subset [1, 254]$ is the set of local extrema in the histogram given by:

$$k \in \mathcal{E}_1 \Leftrightarrow \left(\mathbf{h}_1(k) - \mathbf{h}_1(k-1)\right) \left(\mathbf{h}_1(k) - \mathbf{h}_1(k+1)\right) > 0.$$
(4.2)

Experimental results reported in [26] confirmed that the feature A_1 is statistically larger for original cover Works than for stego Works. Moreover, using this feature in conjunction with a classifier based on Fisher linear discriminant (FLD) [27] analysis, resulted in much better classification results compared with other state-of-the-art steganalyzers, such as WAM [10] or HCF-COM [36, 23].

4.1.1 Removing Interferences at the Histogram Borders

Embedding based on Equation (3.1) introduces a minor asymmetry: 0-valued pixels will *always* be changed to 1 if their LSB needs to be modified. Similarly, 255-valued pixels will *always* be changed to 254. This asymmetry in the histogram can cause interferences with the extracted feature in eq. (4.1). To avoid this problem, Equation (4.1) is modified, as follows:

$$A_1(\mathbf{h}_1) = \sum_{n \in \mathcal{E}'_1} \left| 2\mathbf{h}_1(k) - \mathbf{h}_1(k-1) - \mathbf{h}_1(k+1) \right|$$
(4.3)

where the set of local extrema \mathcal{E}'_1 is now reduced to be within [3, 252]. In other words, the positions $\{1, 2, 253, 254\}$ are not considered as potential local extrema. Nevertheless, to account the bound values of the histogram, the following additional

feature is defined:

$$d_1(\mathbf{h}_1) = \sum_{k \in \mathcal{E}_1^*} \left| 2\mathbf{h}_1(k) - \mathbf{h}_1(k-1) - \mathbf{h}_1(k+1) \right|$$
(4.4)

where $\mathcal{E}_1^* \subset \{1, 2, 253, 254\}$ is a set of local extrema as defined by Equation (4.2).

4.2 Considering 2D Adjacency Histograms

Inspired by [23], the analysis of local extrema has been extended to 2D adjacency histograms [37], $h_2(k, l)$, which tabulates how often each pixel intensity is observed next to another in the horizontal direction $h_2(k, l)$, as defined in Equation (3.7). Since adjacent pixels have, in general, close intensity values, this histogram is sparse off the diagonal. It should be noted that the histogram defined by Equation (3.7) can be slightly modified to obtain 3 other adjacency histograms for other directions (vertical, main diagonal, and minor diagonal). For clarity we will use the apex h, v, D, d, respectively for horizontal, vertical, main diagonal, minor diagonal, to the adjacency function $h_2(k, l)$ in order to specify, if necessary, the kind of adjacency, otherwise $h_2(k, l)$ is referred to a generic kind of adjacency matrix. In particular, we define again the four kinds of adjacency matrix:

$$\mathbf{h}_{2}^{h}(k,l) = \left| \left\{ (i,j) \in \mathcal{I} \mid \mathbf{p}(i,j) = k, \ \mathbf{p}(i,j+1) = l \right\} \right|$$
(4.5)

$$\mathbf{h}_{2}^{v}(k,l) = \left| \left\{ (i,j) \in \mathcal{I} \mid \mathbf{p}(i,j) = k, \ \mathbf{p}(i+1,j) = l \right\} \right|$$
(4.6)

$$\mathbf{h}_{2}^{D}(k,l) = \left| \left\{ (i,j) \in \mathcal{I} \mid \mathbf{p}(i,j) = k, \ \mathbf{p}(i+1,j+1) = l \right\} \right|$$
(4.7)

$$\mathbf{h}_{2}^{d}(k,l) = \left| \left\{ (i,j) \in \mathcal{I} \mid \mathbf{p}(i,j) = k, \ \mathbf{p}(i+1,j-1) = l \right\} \right|$$
(4.8)

where $\mathbf{p}(i, j)$ is the pixel value at location (i, j) in the input image, and \mathcal{I} is a bidimensional index set which runs through all pixel locations in the image.

Moreover, we can extend previous considerations about the ± 1 embedding artefacts on the histogram domain by using the adjacency matrix. In this case, by using ± 1 embedding with payload ρ , we obtain a 2-D low pass filtering with the following kernel:

| $\left(\frac{\rho}{4}\right)^2$ | $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ | $\left(\frac{\rho}{4}\right)^2$ |
|---|---|---|
| $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ | $\left(1-\frac{\rho}{2}\right)^2$ | $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ |
| $\left(\frac{\rho}{4}\right)^2$ | $\frac{\rho}{4}\left(1-\frac{\rho}{2}\right)$ | $\left(\frac{\rho}{4}\right)^2$ |

Consequently, it should also be possible to distinguish between cover and stego Works by examining local amplitude extrema in the 2D adjacency histogram. The set of local extrema in an adjacency histogram $\mathcal{E}_2 \subset [0, 255]^2$ is defined as:

$$\mathbf{p} = (k, l) \in \mathcal{E}_2 \Leftrightarrow \begin{cases} \exists \epsilon \in \{-1, 1\}, \, \forall \mathbf{n} \in \mathcal{N}_+ \\ \operatorname{sign} (\mathbf{h}_2(\mathbf{p}) - \mathbf{h}_2(\mathbf{p} + \mathbf{n})) = \epsilon \end{cases}$$
(4.9)

where $\mathcal{N}_{+} = \{(-1,0), (1,0), (0,-1), (0,1)\}$ is used to define a cross-shaped neighborhood and $\mathbf{h}_{2}(\cdot)$ is the generical adjacency matrix. However, many of these extrema have a small amplitude and are thus highly sensitive to changes of the cover Work. To achieve higher stability, this set is further reduced to:

$$\mathbf{p} = (k, l) \in \mathcal{E}'_2 \Leftrightarrow (k, l) \in \mathcal{E}_2 \text{ and } (l, k) \in \mathcal{E}_2$$
(4.10)

In other words, only pairs of extrema symmetrical with respect to the main diagonal are retained. Empirical observations have revealed that such extrema have significantly higher amplitude and are thus more stable. The resulting generical feature is defined by,

$$A_2(\mathbf{h}_2) = \sum_{\mathbf{p}\in\mathcal{E}'_2} \left| 4\mathbf{h}_2(\mathbf{p}) - \sum_{\mathbf{n}\in\mathcal{N}_+} \mathbf{h}_2(\mathbf{p}+\mathbf{n}) \right|$$
(4.11)

which is the sum of the amplitude of extrema located at positions in \mathcal{E}'_2 .

In addition to eq. 4.11 feature, empirical experiments have demonstrated that the sum of all the elements on the diagonal of a 2D adjacency histogram, defined as follows:

$$d_2(\mathbf{h}_2) = \sum_{k=0}^{255} \mathbf{h}_2(k,k)$$
(4.12)

| 1 | $A_1(\mathbf{h}_1)$ |
|----|--|
| 2 | $d_1(\mathbf{h}_1)$ |
| 3 | $A_2(\mathbf{h}_2^h)$ (horizontal direction) |
| 4 | $A_2(\mathbf{h}_2^v)$ (vertical direction) |
| 5 | $A_2(\mathbf{h}_2^D)$ (main diagonal direction) |
| 6 | $A_2(\mathbf{h}_2^d)$ (minor diagonal direction) |
| 7 | $d_2(\mathbf{h}_2^h)$ (horizontal direction) |
| 8 | $d_2(\mathbf{h}_2^v)$ (vertical direction) |
| 9 | $d_2(\mathbf{h}_2^D)$ (main diagonal direction) |
| 10 | $d_2(\mathbf{h}_2^d)$ (minor diagonal direction) |

Table 4.1: Table of ALE features

could also be exploited to improve classification results. Indeed, ± 1 steganography decreases the value of this feature and its variations can be used in the decision process.

Altogether, the above observations result in a collection of 10 features features which are listed in Table 4.1.

4.3 Performances of ALE

In this Section we describe a number of experiments that we carried out to investigate the impact of the various features on classification performance.

4.3.1 Setup

The experiments were run on a database composed of images originating from three different sources. Specifically:

- 2,375 images from the NRCS Photo Gallery [38]. The photos are of natural scenery, e.g. landscape, cornfields, etc. There is no indication of how these photos were acquired. This database has been previously used in [23].
- 2,375 images captured using 24 different digital cameras (Canon, Kodak, Nikon, Olympus and Sony) previously used in [10]. They include photographs of nat-

ural landscapes, buildings and object details. All images have been stored in a raw format i.e. the images have *never* undergone lossy compression.

• 2,375 images from the Corel database [39]. They include images of natural landscapes, people, animals, instruments, buildings, artwork, etc. Although there is no indication of how these images have been acquired, they are very likely to have been scanned from a variety of photos and slides. This database has been previously used in [26].

The above image sets result in a composite database of 7125 images. Where necessary, all images have been converted to grayscale. Moreover, a central cropping operation of size 512×512 was applied to all images to obtain images of the same dimension across all three source databases. Cropping was preferred over resampling with interpolation, in order to avoid any interference with the source signal.

The motivation for using more than one source database is to account for the variability in steganalyzers' performances across different databases [40, 41]. In the next chapter we fully investigate this variability across image sources. It is hoped that this set of databases will become a reference for subsequent works in steganalysis research.

Given the composite database, the stego images are built by using ± 1 embedding at 0.5 bpp of payload, thus obtaining the stego database. Then, for every image ALE features are extracted and we randomly separated the cover-features database \mathcal{D}_{ALE} and stego features database \mathcal{D}_{ALE}^* into a training set (20% of the database size), and a test set (the remaining 80% of the database) and we built a ROC curve by using Fisher Discriminant classifier on a training set and by projecting all the test feature vectors onto the trained projection vector **u**. To apply a cross validation on the obtained results, we repeat 20 times the above procedure with a different randomization of the train and test datasets. At the end we joined the 20 ROCs by the vertical averaging scheme described in Chapter 2.

The overall performance of the steganalyzer is then measured by computing the area under the ROC curve (AUC).



Figure 4.1: Analysis of the impact of the border effect described in Subsection 4.1.1 on classification results.

4.3.2 Results

Since similar results were observed for various embedding rates, we only report classification results for $\rho = 0.5$.

Figure 4.1 shows the improvements in classification resulting from elimination of border effects. The original algorithm of Zhang *et al.* is compared with a system based on feature 1 of Table 4.1 (ALE 1), and features 1 and 2 (ALE 1-2). The error bars on each plot indicate the minimum and maximum values observed during the 20 cross-validation runs. First of all, we note the unexpectedly poor performances of all three algorithms, i.e. the ROC curves are very close to the diagonal. This is due to the wide variety of images present in of composite database.

Despite the poor performance of all three algorithms, the two algorithms based on new ALE features (ALE 1 and ALE 1-2) exhibit a slight improvement in classification performances. The system using the first two ALE features (ALE 1-2) achieves the highest performances based on area under the ROC curve (AUC), with a score of 0.59, and is therefore used as a reference in the next experiment.



Figure 4.2: Analysis of the impact of ALE features selection on classification results.

Figure 4.2 reports the classification performances achieved when using ALE features computed from the 2D adjacency histogram. Four sets of ALE features are investigated:

- ALE 3-6 i.e. the amplitude of the local extrema in the adjacency histograms,
- ALE 7-10 i.e. the amplitude of the diagonal in the adjacency histograms,
- ALE 3-10 i.e. all features from the adjacency histograms,
- ALE 1-10 i.e. all features from the intensity histogram and the adjacency histograms.

All 4 systems perform at least as well as the reference classification system considered above (ALE 1-2). ALE 3-6 features perform significantly better than ALE 7-10 features. Nevertheless, when these two sets of features are combined (ALE 3-10), the resulting steganalyzer outperforms the systems that rely on a single set of features computed from adjacency histograms. However, the best classification performance

is achieved when all ALE features are combined (ALE 1-10). Compared to the original steganalyzer [26], the area under the ROC curve (AUC) value increases from 0.57 to 0.77, which is a significant improvement.

4.4 Hybrid Algorithm

Experimental issues in steganalysis usually reveal that when the experimental setup is not ideally built in the lab, i.e. no information about payload, image sources and image preprocessing are known, no algorithm has a superior performance over *all* scenarios. Consequently, we also implemented a hybrid steganalysis system that combines the features from all three previously described algorithms.

Let us assume that there are S different steganalyzers $\{S_1, \ldots, S_S\}$ available to perform ± 1 embedding steganalysis. Each steganalyzer S_i relies on some feature vector \mathbf{f}_i , which may have different dimensionality depending on the consider steganalyzer. A commonly used strategy to combine this collection of systems consists in merging all information available, e.g. by concatenating all feature vectors in a single meta feature vector \mathbf{f} as follows:

$$\mathbf{f} = \mathbf{f}_1 | \mathbf{f}_2 | \dots | \mathbf{f}_S \tag{4.13}$$

where | denotes the concatenation operation.

Then applying a classifier on this meta feature vector is expected to increase classification performances. For instance, combining WAM (Chapter 3.2.1), 2D-HCFC (Chapter 3.2.2) and the above ALE results in a 38-dimensional feature vector **f**.

4.5 Discussion

Now it could be interesting to evaluate the performance of ALE in a wider scenario. Unfortunately in steganalysis no evaluation benchmark has ever been designed to this aim as, for example, Stirmark benchmark [42] makes for watermarking applications. However, every proposed steganalyzer¹ should be fully evaluated especially on a real case scenario, by using comparisons with the current state-of-art steganalyzers and the advanced steganography. Unfortunately common comparisons are made between old techniques or specific lab tests in which the image database and the a priori steganalyzer knowledge as used payload or used dataset is really far away from the practical case in which nothing is known. Usually, it could be that a steganalyzer seems to be the best because it obtains good accuracy classification scores in the proposed experimental settings, but at the same time it could be the worst if we use different comparison settings. These considerations are obviously true even for our steganalyzer.

Even though ALE seems to behave very well, an appropriate comparison procedure should be designed to compare ALE behavior against state-of-art classifiers. Specifically, we should investigate how ALE performance vary by changing the experimental conditions by changing both the image database and the payload. Due to the importance of experimental settings and comparison with other steganalyzers like WAM and 2D-HCFC, we will investigate the ALE performance and comparison in the next chapter.

The performance variation across databases, or more in general, a full analysis about ALE and its comparison with the state-of-art steganalysis is shown in Chapter 5. Moreover, the next Chapter describes a new methodology approach for steganalysis comparisons which should be take into account in further steganalysis works.

Chapter 5

Experimental comparison among ± 1 embedding steganalysis

In this chapter we fully investigate ALE performances in comparison with WAM and 2D-HCFC (see Chapter 3). To do so, we define a new benchmark methodology which takes into account the widest possible experimental setting. In this way the obtained results should be as close as possible to a real work steganalysis scenario.

Detection of ± 1 embedding is known to be much more difficult than detecting LSB replacement. Nevertheless, a number of algorithms have been developed for this purpose. Unfortunately, in literature experimental issues did not receive enough attention and often authors do not consider the real constraints set by scenarios that are completely different from those applying to steganalysis or steganography working on a predefined image set or with a predefined payload. An additional problem is that sometimes such a highly controlled scenario may not be reproducible specially when the image database is not shared or it is not carefully described. In these biased situations results are not significant and no comparison between techniques can be made.

In this chapter we would like to propose a comparative steganalysis methodology by showing how results change when the experimental setup changes. To do so we use a FLD classifier and we test ALE, WAM, 2D-HCFC and Hybrid steganalyzers.

5.1 Databases

In our study we used three different databases that have been previously used in the context of steganography and watermarking. The three databases not only contain different images, but, more importantly, the image sources are significantly different, as discussed shortly. The motivation for using more than one database was to determine any variability in performance across databases. A fourth database was created as the concatenation of these three primary databases. It is hoped that this set of databases will become a reference for subsequent works in steganalysis research¹.

The four image databases are:

- 1. NRCS Photo Gallery: This image database is provided by the United States Department of Agriculture [38]. It contains 2,375 photos related to natural resources and conservation from across the USA, e.g. landscape, cornfields, etc. Typically, the image formats are in 32-bit CMYK space color and in high resolution, i.e. 1500×2100 . Unfortunately, there is no indication of how these photos were acquired. This image database has first been used in [23].
- Camera Images: This image database is a collection of 3,164 images captured using 24 different digital cameras (Canon, Kodak, Nikon, Olympus and Sony). It includes photographs of natural landscapes, buildings and object details. All images have been stored in a raw format i.e. the images have *not* undergone lossy compression. A subset of these images was previously used in [10].
- 3. Corel database: The Corel image database consists of a large collection of uncompressed images [39]. They include natural landscape, people, animals, instruments, buildings, artwork, etc. Although there is no indication of how these images have been acquired, they are very likely to have been scanned from a variety of photos and slides. Moreover, a close inspection of the grayscale histogram of several pictures tend to suggest that the images have been submitted to some kind of histogram equalization technique. This process introduced significant artifacts in the histogram which, as a by-product, significantly boost the performances of the ALE steganalyzer as will be detailed late. A subset of 8,185 images has been extracted from the database with dimension 512×768 .

¹To encourage the use of this database, it is accessible on the website [43].

4. **Combined database:** A fourth database was created by concatenating 2,375 randomly selected images from each of the three databases.

Where necessary, all images have been converted to 8-bit depth grayscale. Moreover, a central cropping operation of size 512×512 was applied to all images to obtain images of the same dimension across all three databases. Cropping was preferred over resampling with interpolation, in order to avoid introducing artifacts due to signal processing.

5.2 **Experimental Procedure**

For each one of the four databases (NRCS, Camera, Corel, Combined), the following procedure was performed for every steganalyzer under study (WAM, 2D-HCFC, ALE, Hybrid):

- Apply LSB embedding with embedding rate ρ to all images in the database D to obtain the database of stego images D*;
- Separate both databases into a training set, {D(U), D*(U)}, and a test set, {D(U_J), D*(U_J)}, where U is a subset of the image indexes and U_J is its complement. The size of the training set was set to be equal to 20% of the database size;
- For the steganalyzer under test, compute the associated feature vector for all images in the training set and perform FLD analysis to obtain the trained projection vector u;
- 4. For the steganalyzer under test, compute the associated feature vector for all images in the test set, and project the feature vector onto **u**;
- 5. Compare the resulting scalar values to a threshold τ and record the probabilities of false positives and true positives for different values of the threshold in order to obtain the Receiver Operating Characteristic (ROC) curve of the system.

Steps 2 to 5 were repeated 20 times for cross-validation [27] and the ROC curves vertically averaged. That is, for a fixed false positive value, the corresponding true positive rates for each curve were averaged. The confidence level at each false positive point depicted in the resulting curves indicates the minimum and maximum true positive rates form the set of ROC curves.

Thresholding averaging of the ROC curves is also possible, as previously discussed. For example, for the ALE algorithm and a given threshold, we obtain k = 20 points corresponding to the true and false positive rates for the k-trials, and these points lie in reasonably close proximity to one another. However, for the WAM algorithm, and consequently the hybrid algorithm as well, these k = 20 points are dispersed across the ROC curve, i.e. the variances are very large.

Although we have not considered them in our study, alternative performances metrics have been suggested in the literature e.g. the detection reliability which is simply derived from the AUC [44], the false positive rate at 50% (80%) detection rate [10], and others.

5.3 Experimental Results

In an attempt to obtain a better understanding of the different steganalyzers under study, we first examine the impact of the source of imagery used during training and testing, and in particular the consequences of using mismatching imagery. Next, we investigate the influence of the embedding rate depending on the testing conditions. Based on this analysis, we then further detail the performances of the individual steganalyzers depending on whether or not some prior about the source of imagery is available before the steganalyzer is run. Such a priori information could for instance be obtained thanks to forensic tools.

5.3.1 Impact of the source of imagery

In the first batch of experiments, the embedding rate is fixed and set equal to ρ =0.5 bit per pixel (bpp), both during training and testing. Similar behavior was observed for other embedding rates but data is omitted for brevity and clarity. Each



Figure 5.1: Impact of the source of imagery on classification performances. The embedding rate has been fixed both during the training and testing phase and set equal to 0.5 bpp. The label of the rows indicates the database used for training while the label of the columns represents the dataset used during the testing phase.

individual steganalyzer has been successively trained using images from one of the four databases considered in this study (NRCS, Camera, Corel, Combined). Subsequently, each trained steganalyzer is benchmarked with each individual database. It results in $4 \times 4 = 16$ possible combinations for training and testing conditions. For each scenario, the average ROC curve of each steganalyzer has been computed as described in Section 5.2 and the results are reported in Figure 5.1². The label of the

²In order to remove as much redundant information as possible and therefore facilitate the reading of the plots, all axis labels and ticks have been removed in these plots and the following ones. All plots

rows indicates the database used for training and the label of the columns the one used for testing. As a result, plots on the diagonal have matching training and testing conditions. They have been framed to clearly highlight them.

Let us first focus on the 3×3 block of figures in the top left corner. According to expectation, the best performances are achieved when the training and testing conditions match (plots on the diagonal). However, even in these conditions, it is clear that the absolute performance of the four algorithms varies considerably across the three primary image databases. Additionally, the relative performance is also seen to vary. For the NRCS and Camera testsets, the WAM algorithm exhibits best performance. However, even across these two testsets, the absolute performance varies significantly. For example, we observe that for a false positive rate of 10%, the WAM algorithm has a true positive rate of 30% and 60% for NRCS and Camera respectively. There are similar variations for the other two algorithms. For the Corel testset, the ALE algorithm performs much better. Also, interestingly, we observe a very strange behavior from the 2D-HCFC algorithm, where the true positive rate remains almost constant as the false positive rate increases from 20% to 70%. As might be expected, the hybrid algorithm exhibits the best performance for each of the three individual databases.

As soon as we deviate from the diagonal, i.e. when training and testing conditions no longer match, we observe drastic performance degradation for all four algorithms and significant increase in variability. This indicates that each individual database is specific and is not representative of the other two databases. As a result, it illustrates the importance of training with a dataset that is representative of the classes of images that will be observed in real life. If this is not done, then the performance of algorithms is likely to be worse than expected. For instance, if the Hybrid algorithm is trained with NRCS images whereas it will only encounter Camera images, then its performances is significantly reduced compared to if it has been trained with Camera images, with an AUC score reducing from 0.89 to 0.58. As a matter of fact, it no longer the best performing algorithm, but is in fact the

share the same axis, i.e. false positive vs. true positive with all axis running from 0 to 1 with a linear scale.

worse performer. This may sound rather counter-intuitive at first. Indeed, one would expect that the hybrid system performs at least as well as the others e.g. by taking the projection vector used for the best performing steganalyzer and padding it with zeros. And this is true! However, the projection vector used for classification is given by the FLD analysis described in Chapter 2.3. During this process, two parameters are optimized at the same time: the within-class scatter (S_W) and the between-class scatter (S_B). In other words, the optimization process for the within-class scatter may come into the way of maximizing the separability between the two classes, hence resulting in degraded classification performances. This is clearly a limitation of FLD analysis which has motivated the use of Support Vector Machines (SVM) in recent steganalysis systems.

Additionally, the increased variability of the ROC curve observed during crossvalidation clearly indicates that there is no guaranteed performances with such mismatching training strategies. As a result, a straightforward rule of thumb is 'if you know the source of imagery (whatever this means) that you will encounter in your application, you should train with it'.

Now, let us assume that the steganalyzers have been trained with a single source of imagery but that they actually have to deal with a variety of images in practice (3 first rows of the last column). Since there is still a significant mismatch between the training and testing conditions, we observe a significant reduction of performances and a huge increase of variability. As a result, this calls for training the classifiers with a variety of images (last row). Such strategy usually slightly hampers performances when testing on individual databases compared to the classification results achieved with matching training and testing conditions. This seems to be particularly true for the NRCS database where, for instance, the WAM steganalyzer sees its AUC value drop from 70% to 49%, i.e. nearly random guessing. Still, the variability of the ROC curves during cross-validation is now significantly reduced compared to the situation where the training and testing databases do not match. This low variability is crucial in order to be able to guarantee performances. Moreover, the figure in the bottom right corner clearly shows that training steganalyzers, when the system ac-

tually encounters varied sources of imagery in practice.

In summary, the previous observations clearly indicate that, if the steganalyst has some a priori information about the source of imagery that the system will encounter in practice, the steganalyzer should be trained with that specific source of imagery. For instance, one could imagine that a forensics module could be placed at the beginning of the system to accurately switch to the most appropriate steganalyzer for each input test image. On the other hand, if the steganalyst has no a priori knowledge, then the system should be trained with the most varied sources of imagery as possible in order to maintain performances.

5.3.2 Impact of the embedding rate

The previous results refer to the case where both training and testing were conducted for a known, fixed embedding rate of ρ =0.5 bpp. In practice, the steganalyst is unlikely to have knowledge of the embedding rate used by the steganographer. Thus, it is necessary to design a steganalysis algorithm that performs well for a variety of embedding rates.

In the second round of experiments, both training and testing have been conducted by using the Combined database, since the previous observations strongly hinted that it was the most relevant training strategy. Each individual steganalyzer has been successively trained using stego content obtained with an embedding rate ρ equal to 0.2, 0.5, 1 bpp³ or a uniform mix of these embedding rates. Subsequently, each trained steganalyzer is benchmarked with stego content obtained, again, with an embedding rate equal to 0.2, 0.5, 1 bpp or uniform mix of these embedding rates. It results in $4 \times 4 = 16$ possible combinations for training and testing conditions. For each scenario, the average ROC curve of each steganalyzer has been computed as described in Section 5.2 and the results are reported in Figure 5.2. The label of the rows indicates the embedding rate used for training and the label of the columns the one used for testing. As a result, plots on the diagonal have matching training and testing conditions. They have been framed to clearly highlight them.

³More exhaustive tests were conducted over a wider range of embedding rates. However, the behavior is the same.



Figure 5.2: Impact of the embedding rate on classification performances. The source of imagery during both the training and testing phase is the Combined dataset. The label of the rows indicates the embedding rate used for training while the label of the columns represents the embedding rate used during the testing phase.

Again, let us first focus on the 3×3 top left figures. Intuitively, one would expect that a steganalyzer trained to spot steganography at some embedding rate should be able to better detect, to some extent, steganography at higher embedding rates. And, conversely, a steganalyzer trained at some embedding rate should miss more stego contents if they are produced with a lower embedding rate. This intuitive rule seems to hold generally i.e. classification performance increase when going to the right of the the diagonal and decreases when going to the left. However, it is not always the case for the WAM algorithm, and by inheritance for the Hybrid algorithm. For

instance, when training is done at 0.2 bpp (first row), the classification performance for WAM first slightly raises from an AUC value of 63% when testing on ρ =0.2 bpp, the same embedding rate as for training, to an AUC value of 64% when testing on ρ =0.5 bpp; however, it then collapses down to an AUC value of 55% when testing on ρ =1 bpp. Similarly, still for the WAM algorithm, when trained at 0.5 bpp, the AUC value reduces from 68% for the 0.5 bpp testset to 63% for the 1 bpp testset. This peculiar behavior seems to suggest that the WAM algorithm, and also the Hybrid algorithm to some extent, learn different features for different embedding rates and is therefore unable to cope with stego contents obtained with embedding rates not considered during training.

The above phenomena could be annoying when the embedding rate used by the steganographer is unknown (last column), which is actually the most likely case in a realistic scenario. For instance, the WAM algorithm, when trained at 0.2 bpp is the only steganalyzer whose classification performance is worse with a testset containing mixed embedding rates rather than a single embedding rate ρ =0.2 bpp, i.e. the same as during training. As a result, it is no longer straightforward to state "train your steganalyzer with a low embedding rate and as a by-product it will also be able to detect all other payloads". As a matter of fact, the behavior of the other algorithms tends to suggest that steganalyzers trained at 0.5 bpp achieve slightly better performances. This should not be reduced to a universal rule: this *optimal* embedding rate for training, if there is any, is most likely to be dependent on the distribution of the embedding rates used by the steganographer, which may be difficult to figure out or estimate. In our case, since a uniform distribution over the embedding rates has been assumed, training at 0.5 bpp might be the best since stego contents are then closer in average to the training embedding rate compared to 0.2 or 1 bpp.

Now, let us assume that the steganalyst is able to figure out the distribution of embedding rates used by steganographers across the world and that he uses this knowledge to train the different classifiers with the same distribution (last row). In this setup, we observe that we achieve classification performances very similar to the one obtained while training at 0.5 bpp, i.e. the best performances so far. This suggests that, should the distribution of embedding rates used by steganographers be known, it should be exploited during the training of the steganalysis algorithms.

5.3.3 Performances of the steganalyzers with prior information about the source of imagery

Although the experimental results of the previous section clearly suggest that training should be performed with the same distribution of embedding rates as the one encountered during testing, it is still not clear how a steganalysis algorithm should be trained in general. Should we use a single source of imagery or a combination of all known sources? As already mentioned, the answer it heavily depends on whether or not, during the testing phase, we have some additional tools, e.g. multimedia forensics techniques, which gives some a priori information about the source of imagery of the tested content. In such a case, the steganalyst can switch to the relevantly trained classifier accordingly. In this section, we will assume that we do have access to such a priori information and will review the detailed performances of the different algorithms under study.

Each steganalyzer is trained on each of the three available databases (NRCS, Camera and Corel). The stego contents used for training are obtained using embedding rates uniformly distributed across 0.2, 0.5 and 1 bpp as suggested by prior findings. Since we do assume to have prior information, we then test each classifier with contents taken from the same database as the one using during training. Still, to get a better understanding of their classification performances, the steganalyzers are successively tested with stego contents obtained with an embedding rate equal to 0.2, 0.5, 1 bpp or a uniform mix of these. It results in 3×4 training and testing scenarios and, for each one of them, the average ROC curve of all steganalyzers has been computed as described in Section 5.2. All the plots have been gathered in Figure 5.3. The label of the rows indicates the database used both during training and testing and the label of the columns the embedding rate used for testing.

Intuitively, one could expect that, for some training conditions i.e. for a given row, stego contents obtained with large embedding rates should be detected more easily than those with smaller ones. This rule seems to hold in most cases except, again, for the WAM algorithm on the Camera database. For this particular dataset,



Figure 5.3: Classification performances when prior information about the source of imagery is available. Training is done with stego contents obtained with embedding rates uniformly distributed across 0.2, 0.5 and 1 bpp. The label of the rows indicates the source of imagery used both during training and testing. On the other hand, the label of the columns represents the embedding rate used during the testing phase.

the AUC value indeed raises from 73% for 0.2 bpp to 86% for 0.5 bpp before decreasing down to 80% for 1 bpp. The fact that this behavior is only observed for the Camera databases may be due to the fact that this dataset is actually composed of images taken with different cameras i.e. different sources of imagery strictly speaking. In any case, it does highlight the extreme sensitivity of the WAM algorithm. The plots for mixed payload give some kind of average of the ROC curves along the rows. However, these plots completely mask specific behaviors for different embedding rates. For instance with the NRCS database, efforts should be focused on detecting low embedding rates since performances are really bad in that case. It might indeed be easier to raise the AUC of really poor ROC curves than further enhance not so bad ROC curves. This calls for reporting more than only the average curve (mix) to get a better understanding of the system. Also worth mention, classification performances appear to be heavily dependent on the source of imagery. In average for instance, the AUC value of the best performing algorithm is equal to 75% for NRCS, 85% for Camera and 93% for Corel, hence clearly that some type of images might be more difficult to tackle than others. Additionally, setting apart the Hybrid algorithm, the best performing steganalysis algorithm seems to change from one dataset to the other. ALE definitely outperforms the others on the Corel database, being matched by 2D-HCFC only for high embedding rates. With the Camera dataset, the situation is more contrasted: WAM is better for low embedding rates but is being outmatched by ALE for high embedding rates. Finally, WAM and ALE are side by side for the NRCS database. Still, even in that case, combining the feature sets of both algorithms succeeds to significantly improve performances hence demonstrating the complementarity of these two systems.

5.3.4 Performances of the steganalyzers without prior information about the source of imagery

In contrast with the previous subsection, we now assume that the steganalyst is unable to figure out the source of the content which is to be tested. In other words, he can no longer switch pertinently between several specifically trained classifiers. This scenario is significantly more realistic in practice since one can hardly tell how many sources of imagery should be considered to be close to the real world. As a result, we conducted a final batch of experiments to address this specific situation.

In this scenario, the steganalyst trains the different steganalysis systems with the Combined database and stego contents obtained with an embedding rate uniformly distributed across 0.2, 0.5 and 1 bpp. Still, the obtained classifiers are benchmarked for individual databases (NRCS, Camera, Corel, combined) and individual embedding rates (0.2, 0.5, 1 bpp and mixed payloads). It results in 4×4 testing scenarios. For each of them, the average ROC curve of all steganalyzers has been computed as detailed in Section 5.2 and the resulting plots are depicted in Figure 5.4. The label of the rows indicates the dataset used for testing and the label of the columns the embedding rate.

Again, classification performances improve when the embedding rate increases.



Figure 5.4: Classification performances when no prior information about the source of imagery is available. Training is done with the Combined database and stego contents obtained with embedding rates uniformly distributed across 0.2, 0.5 and 1 bpp. The label of the rows indicates the source of imagery used during testing, while the label of the columns represents the embedding rate used also during the testing phase.

Even the peculiar behavior previously observed for the WAM algorithm on the Camera database is significantly attenuated. On the other hand, we can observe than in average performances are significantly hampered compared to the previous situation where training was performed for specific sources of imagery. For instance, on the NRCS and Camera data sets, the AUC values of the ROC curves can be decreased by up to 10%. The only exception is of course the 2D-HCFC algorithm since it does not involve any kind of training (1D feature space) and therefore is not affected by this change of training setup. Additionally, we can observe a mild increase in the variability of the ROC curves, thus indicated increased instability for the different steganalysis systems. This drop of performances suggests that it might be utopian to train a single classifier to address all situations and that it might be more relevant to pertinently switch between specifically trained classifiers.

One could argue that the only relevant plot is the one in the bottom right corner as it reports the performances of the different steganalysis system in conditions close to the real world. In that case, one would realize that classifications performances are average at best. One could also be surprised by the suggested ranking of the different classifier, the well-known WAM being the worst (most likely due to its loss of stability) and the underrated 2D-HCFC scoring not so badly. However, focusing on this single plot kind of hide the most important information: where should efforts be targeted to further improve these performances. It is clear for instance that very little improvement is likely to be achieved on the Corel dataset. On the other hand, the NRCS dataset offers huge room for further improvement even at high embedding rate.

5.4 Conclusions

We compared four steganalysis algorithms applied to the detection of ± 1 embedding.

We stressed out that during training, it is necessary to provide a labeled set of cover and stego Works. The stego Works are usually derived from the application of known steganographic algorithms. However, even for the case of targeted steganalysis there are a range of free parameters available to the steganographer, but usually unknown to the steganalyst. The most common such parameter is the embedding rate. It is quite usual to report results assuming exact knowledge of the embedding rate, i.e. training and testing are for a fixed embedding rate. Even though there are some works that try to estimate the message length [45, 46, 47, 48], in practice, no-body will know the embedding payload. If training and testing are conducted over a range of embedding rates, we can expect performance to degrade. Our study also revealed this. It is common to train with a low embedding rate in order to cope with a wide range of embedding rates used during testing. However, experimental results suggest that this is less effective than training with a mixture of embedding rates.

In summary, our experiments revealed that (i) the performance of all algorithms varied significantly depending on the database, (ii) no one algorithm was superior across all databases. In particular, we have seen that training in real world conditions, e.g. mixed embedding rates (and combined database), may have a significant impact on performances compared to tightly controlled situation.

Chapter 6

Steganalysis: remarks and future works

In the first part of the thesis we have discussed steganalysis in the pixel domain by proposing the steganalyzer ALE and a methodology to experimentally evaluate the performance of a steganalysis algorithms.

With regard to ALE, we modified the algorithm by Zhang *et al.* to deal with (i) border effects associated with the 1D intensity histogram, and (ii) extended it to include statistics associated to the amplitude of local extrema in the 2D adjacency histogram.

Experimental results demonstrated the positive impact of eliminating the border effects and showed substantial improvements in classification accuracy when features derived from the 2D adjacency histogram were included. Moreover, the proposed steganalysis system proved to outperform other state-of-the-art steganalyzers such as WAM [10] and 2D-HCFC [23].

As future works, we could improve further the performances of ALE by using a calibrated version of the features as suggested by Ker in [23].

We have also discussed a number of issues in training and testing steganalysis algorithms and illustrated these issues by comparing four algorithms for the detection of ± 1 embedding.

While the community recognizes the importance of standardized training and test sets, it is clear that current databases are inadequate. In particular, we observed very significant variations in performance across the four databases used in the test for all the algorithms under evaluation. This indicates that no one database is currently sufficiently representative of the variety of imagery that may be encountered in the real world. More research is needed to (i) understand how various databases differ from one another, and (ii) develop more comprehensive databases that better represent the variation in real world imagery.

Performance results are usually reported in the form of a receiver operating characteristic curve, and we have followed this convention. Cross-fold validation generates a family of ROC curves which must be merged. Vertical averaging and threshold averaging are two approaches for doing so. Both have the advantage of providing a measure of the uncertainty associated with the ROC values. In our analysis, we observed an extreme range of uncertainty when attempting to apply threshold averaging to the WAM (and hybrid) algorithms. Consequently, we chose to perform vertical averaging. Probably this uncertainty is due directly to the cardinality of features. A steganalysis open question regards about the best feature strategy: is a big amount of features [49, 50, 51] preferred to few features [52, 53, 54] in a real scenario?

Future works should investigate this more closely and it would be beneficial if the community agreed to standardize on one or other approach.

In many situations, it is useful to summarize the performance of an algorithm with a single scalar value. One such value is the area under the ROC curve (AUC). While this is a common measure for summarizing ROC performance, further discussion is needed to decide whether the AUC is adequate and/or sufficient for comparing steganalysis algorithms.

Part II

MPSteg-color: a new steganographic technique
Chapter 7

Steganography at higher semantic level

Common steganalyzers like those described in Chapter 3 and Chapter 4 rely on a statistical analysis to understand whether a given signal contains hidden data or not, however, this analysis disregards the semantic content of the cover signal. For this reason it may be argued that, from a steganographic point of view, it is preferable to embed the stego-messages at the highest possible semantic level, e.g. by modifying structural elements of the host signal like lines, edges or flat areas in the case of still images.

Following a similar need arising from image compression applications¹ [55, 56], a new class of image representation methods has been recently developed that relies on redundant bases decomposition. In practice, a dictionary with a large number of elementary signals (called atoms) is built, trying to ensure that, for each image (or image block), a subset of few atoms exists that permits to represent the image efficiently. The main problems with redundant basis decomposition of images are the construction of the dictionary and, more importantly, the definition of an efficient procedure to select the best subset of atoms for each image. The most common approach to solve the latter problem, consists to resort to Matching Pursuit (MP) techniques, that use a greedy algorithm to select a subset of atoms capable of representing the to-be-decomposed image efficiently [57].

As a main result MP schemes permit to decompose images efficiently by describing the main features of picture's semantic.

Similar ideas about the usage of a semantic layer as message support are widely investigated in watermarking field [58, 59, 60] in which the robustness and the invis-

¹As a matter of fact, the goal of any compression algorithm is to describe the main semantic features of the image without considering noise-like details.

ibility of the watermark is required.

We propose a new steganographic method, called MPSteg-color, that hides the stego-message into some selected coefficients of the MP representation of the cover color image. In this way the hidden message is embedded at a higher semantic level and hence it should be more difficult for a steganalyzer to detect it. To actually build a steganographic technique based on MP decomposition several problems need to be solved including: i) the choice of a suitable dictionary, ii) the setting up of MP rules which permit to correctly embed and extract the message in the MP domain, iii) the implementation of security aspects in order to prevent the detectability of the proposed technique.

In the sequel we show how we investigates and solved all the above problems.

7.1 Introduction to MP image decomposition

Given a vector space V, a high redundant basis is a set of elements of V whose number greatly exceeds the dimension of V. The main idea behind the use of redundant basis for signal representation is that for any given signal it is likely that we can find a small subset of elements within the basis which are enough to represent the signal up to a certain accuracy level. Indeed, the more elements are contained in the basis the more likely the representing set will be small. Of course, since the number of signals in the basis exceeds the size of the space the host signal belongs to, the elements of the basis will no longer be orthogonal as in standard signal decomposition. At the same time, the availability of many degrees of freedom in the design of the redundant basis permits to include signals with specific semantic meaning.

In the following, the elements of the redundant basis will be called atoms, and the redundant basis the dictionary. The dictionary is usually indicated as D:

$$\mathcal{D} = \{g_k\}_{k \in 1, \dots, N},\tag{7.1}$$

where g_k is the k-th atom. If \mathcal{I} is a generic signal (hereafter an image), we can

describe it as the sum of a subset of elements of \mathcal{D} :

$$\mathcal{I} = \sum_{k=1}^{N} c_k g_k,\tag{7.2}$$

where c_k is the specific weight of the k-th atom, and where as many c_k as possible are zero. There are no particular requirements concerning the dictionary: in fact, the main advantage of this approach is the complete freedom in designing \mathcal{D} which can then be efficiently tailored to closely match signal structures. Due to the nonorthogonality of the atoms, the decomposition in equation (7.2) is not unique, hence one could ask which is the best possible way of decomposing \mathcal{I} . Several meanings can be given to the term *best decomposition*. In compression applications, for instance, it is necessary that a suitable approximation in terms of human perceptible distortion of the image \mathcal{I} is obtained. In this case, it is convenient to restate the decomposition problem as follows. Let $\gamma = {\gamma_1, \gamma_2 \dots \gamma_N}$ be a decomposition path, with γ_k indicating the index of the k-th atom of the decomposition. Let us also define the residual signal \mathcal{R}^n as the difference between the original image \mathcal{I} and the approximation obtained by considering only n atoms of the dictionary. We have:

$$\mathcal{I}^n = \sum_{k=1}^n c_k g_{\gamma_k},\tag{7.3}$$

$$\mathcal{R}^n = \mathcal{I} - \mathcal{I}^n, \tag{7.4}$$

where γ_k ties the atom identifier to the k-th position of the decomposition sum.

Given the above definitions, the best approximation problem can be restated as follows:

$$\min_{\gamma, c_k: \|\mathcal{R}^n\|^2 \le \varepsilon} n \tag{7.5}$$

where ε is suitable approximation error. Unfortunately, the above minimization is an NP-hard problem, due to the non-orthogonality of the dictionary [61]. Matching Pursuit is a greedy method that, by looking for a suboptimal solution, permits to overtake the above NP problem with a polynomial complexity algorithm [61], by looking for a step by step minimization of the current residual \mathcal{R}^k . While MP finds the best solution at each step, it generally does not find the global optimum.

In the following, we find convenient to rephrase MP as a two-step algorithm. The first step is defined through a selection function that, given the residual \mathcal{R}^{k-1} , selects the appropriate element of \mathcal{D} and its weight:

$$[c_k, g_{\gamma_k}] = \mathcal{S}(\mathcal{R}^{k-1}, \mathcal{D}), \tag{7.6}$$

where $\mathcal{S}(\cdot)$ is a particular selection operator. At the second step, the residual is updated

$$\mathcal{R}^{k} = \mathcal{U}(\mathcal{R}^{k-1}, c_{k}, g_{\gamma_{k}}).$$
(7.7)

As it can be seen, for a complete definition of the MP framework several specifications must be given including the definition of the dictionary, the selection and the update rules. To do so, we must first investigate the requirements set by the particular framework in which we will apply the MP algorithm, i.e. image steganography.

7.2 Embedding a message in the MP domain

Given the representation formula

$$\mathcal{I} = \sum_{k=1}^{n} c_k \cdot g_{\gamma_k} + \mathcal{R}^n, \tag{7.8}$$

there are different ways of embedding a message within \mathcal{I} . In [62], for instance, the stego-messages is hidden in the particular decomposition path used to represent the image, whereas in [57] and [8], the message is hidden by modifying the decomposition coefficients c_k . In this thesis, we adopt the latter approach, due to the difficulties of applying the former strategy in a blind detection framework (indeed the scheme described in [62] requires non-blind detection). However, this strategy requires several problems to be addressed.

First of all, it is necessary that the transition from the pixel domain to the MP domain and then back to the pixel domain does not introduce approximation errors

that could prevent the correct decoding of the stego-messages. The easiest way of achieving this result consists in requiring that all the operations are performed in integer arithmetic with no need to quantize the stego image when the transformation from the MP to the pixel domain is performed.

The second requirement stems from the very goal of all our work, that is to embed the stego-message at as high semantic level as possible, hence the dictionary should be as semantically meaningful as possible.

The third and the most fundamental requirement, regards the stability of the MP decomposition. MP instability has two different facets:

- Decomposition path instability: this source of instability is due to the fact that the insertion of the message may change the order in which the atoms are chosen by the MP algorithm. As a matter of fact, if this is the case, the decoder will fail to read the hidden message correctly (note that in image compression, where the image is reconstructed from a list of weighed atoms, the fact that a successive decomposition generates a different list of atoms is not a problem).
- Coefficient instability: the second source of instability derives from the nonorthogonality of the dictionary: if we modify one single coefficient c_{k^*} , reconstruct the modified image and apply the MP algorithm again, even if we do not change the order in which the atoms are selected, it may well be the case that all the coefficients will have different values. Even worse, there is no guarantee that the coefficient of the k^* -th atom will be equal to the value we set it to. It is easy to show that this is the case, for example, if the selection and update rules are based on the classical projection operator.

As a last observation, we note that, even though MP decreases the decomposition problem to polynomial complexity, the computational burden may still be prohibitive, especially if MP is applied to large image blocks. For this reason we decided to apply MP to small non-overlapping blocks rather than to consider the whole image. Note, however, that in principle, the subsequent discussion can be indifferently applied to the whole host image or to subparts of it.

In the next two chapters we describe how the above constraints are satisfied by

MPSteg color. We first describe the dictionary, then we introduce new selection and update rules explicitly designed to avoid coefficient instability.

Chapter 8 An MP tailored for steganographic application

In this chapter we introduce the MP domain in which we will embed the message. To do so we introduce the used dictionary, and we define proper selection and update rules. The designed domain is then analyzed from the embedding point of view by defining constraints which permit to correctly extract the embedded message and the semantic point of view describing performances to validate the theoretical semantic approach.

8.1 Dictionary

There are several ways of building the dictionary. Discrete- or real-valued atoms can be used and atoms can be generated manually or by means of a generating function. In classical MP techniques, applied to still images [55], the dictionary is built by starting from a small set of generating functions that generate real-valued atoms. A problem with real-valued atoms is that when the modified coefficients are used to reconstruct the image in the pixel domain, non-integer values may be produced, thus resulting in a quantization error when the grey levels are expressed in the standard 8-bit format. This is a problem in steganographic applications where the hidden message is so weak that the quantization error may prevent its correct decoding. For this reason, and to prevent instability problems, we decided to work with binary-valued atoms for which only the 0 and 1 values are allowed.

The most important property of the dictionary is that it should be able to describe each type of image with a linear combination of few atoms. To simplify the construction of the dictionary and to keep the computational burden of the MP decomposition low, we decided to work on a block by block basis, applying the MP algorithm to



Figure 8.1: A subset of the atoms the dictionary consists of.

 4×4 blocks. At this level, each block may be seen as the composition of few fundamental geometric structures like flat regions, lines, edges and corners. Specifically, we designed the dictionary by considering elements which describe uniform areas, contours, lines, edges, C-junctions, H-junctions, L-junctions, T-junctions and X-junctions. In Figure 8.1 the basic (non-shifted) atoms forming the dictionary are shown. The complete dictionary is built by considering the atoms reported in Figure 8.1 and their cropped 4×4 version when the center of the zero-padding atom - at coordinate (2,2) - is shifted around the 4×4 crop window. The whole dictionary is formed by 324 distinct atoms.

8.2 MP selection and update rules

In order to avoid that quantization errors prevent the correct decoding of the hidden message, let us observe that the stego-messages will be embedded in the MP domain by modifying the coefficients c_k in equation (7.3), however, after embedding, the modified image must be brought back into the pixel domain. If we want to avoid the introduction of quantization errors it is necessary that the reconstructed image

belongs to the Image class. The Image class is defined by the following property:

Property 1. Let \mathcal{I} be a generic gray image¹ in the pixel domain and let $n \times m$ be its size. Let $\mathcal{I}(x, y)$ be the value of the image \mathcal{I} at x-row and y-column. We say that \mathcal{I} belongs to the Image class if:

$$\forall x \in 1, \dots, n, \forall y \in 1, \dots, m \\ 0 \le \mathcal{I}(x, y) \le 255 \quad and \quad \mathcal{I}(x, y) \in \mathbb{N},$$

the value 255 is used by considering an 8 bit color depth for each color band.

The necessity of ensuring that at each step the approximated image and the residual belong to the *Image class* already suggested us to consider binary-valued atoms, now we also impose that atom coefficients take non-negative integer values. In this way, we ensure that the reconstructed image belongs to the *Image class*²

Coefficient instability is more difficult to deal with, especially when coupled with the requirement that the decomposition path includes atoms matching the structural content of the image. MPSteg-color achieves the above result by defining the selection rule as follows. At each decomposition step k let

$$\mathcal{S}(\mathcal{R}^{k-1}, \mathcal{D}) = [c_k^*, g_{\gamma_k^*}] \tag{8.1}$$

with

$$\gamma_k^* = \arg\min_{\gamma_k \in \{1, 2, \dots, |\mathcal{D}|\}} \sum_{i, j} \|\mathcal{R}_{\gamma_k}^k(i, j)\|^2$$
(8.2)

and

$$\mathcal{R}^k_{\gamma_k} = \mathcal{R}^{k-1} - c^*_k g_{\gamma_k},\tag{8.3}$$

¹It is possible to extend this definition to RGB images by considering each color band as a gray image.

 $^{^{2}}$ Actually we must also ensure that no underflow or overflow errors occur. We will consider this problem later on in Chapter 10.



Figure 8.2: The Selection Rule.

where the notation $\mathcal{R}^k_{\gamma_k}(i, j)$ makes explicit the dependence of the residual at the k-th step on the selected atom, and where c_k^* is computed as follows:

$$c_k^* = \max\{c \ge 0 : \mathcal{R}^{k-1} - cg_{\gamma_k} \ge 0 \quad \text{for every pixel}\}.$$
(8.4)

An illustration of the behavior of the selection rule is given in Figure 8.2, where the choice of c_k is shown in the one-dimensional case. By starting from the residual R^{k-1} (solid line) and the selected atom g_{γ_k} (dashed), the weight c_k is calculated as the maximum integer for which $c_k g_{\gamma_k}$ is lower than or equal to R^{k-1} (the dotted line in the figure). Note that given that the atoms take only 0 or 1 values, at each step the inclusion of a new non-null term in the MP decomposition permits to set to zero at least one pixel of the residual. Note also that the partial residual \mathcal{R}^k continues to stay in the *Image class*.

We must now determine whether the selection rule described above is able to avoid the instability of MP coefficients. This is indeed the case, if we assume that the decomposition path is fixed and that only non-zero coefficients are selected for embedding, as it is shown by the following theorem. **Theorem 1.** Let $\mathcal{I} = \mathcal{R}^0$ be an image and let $\vec{g}_{\gamma} = (g_{\gamma_1}, \ldots, g_{\gamma_n})$ be a decomposition path. We suppose that the atoms are binary valued, i.e. they take only values 0 or 1. Assume that the MP decomposition coefficients are computed iteratively by means of the following operations:

$$c_k = \max\{c \ge 0 : \mathcal{R}^{k-1} - cg_{\gamma_k} \ge 0$$

for every pixel} (8.5)

$$\mathcal{R}^k = \mathcal{R}^{k-1} - c_k g_{\gamma_k}, \tag{8.6}$$

and let $\vec{c} = (c_1, c_2, ..., c_n)$ be the coefficient vector built after *n* iterations. Let c_k be an element of \vec{c} with $c_k \neq 0$, and let \vec{c}' be a modified version of \vec{c} where c_k has been replaced by c'_k . If we apply the MP decomposition to the modified image

$$\mathcal{I}' = \sum_{i=1, i \neq k}^{n} c_i \cdot g_{\gamma_i} + c'_k g_{\gamma_k} + \mathcal{R}^n$$
(8.7)

by using the decomposition path \vec{g}_{γ} , we re-obtain exactly the same vector \vec{c}' and the same residual \mathcal{R}^n .

Proof. To prove the theorem we introduce some notations. We indicate by $S(g_{\gamma_k})$ the support of the atom $(\gamma_k)^3$. This notation, and the fact that

$$g_{\gamma_k}(x,y) \in \{0,1\} \ \forall (x,y),$$

permits us to rewrite the rule for the computation of c_k as follows:

$$c_k = \min_{(x,y) \in S(g_{\gamma_k})} \mathcal{R}^{k-1}(x,y).$$
(8.8)

We indicate by j_k the coordinates for which the above minimum is reached, i.e.:

$$j_k = \arg\min_{(x,y)\in S(g_{\gamma_k})} \mathcal{R}^{k-1}(x,y).$$
(8.9)

³The support of an atom is defined as the set of coordinates (x, y) for which $g_{\gamma_k}(x, y) \neq 0$

Note that after the update we will always have $\mathcal{R}^k(j_k) = 0$. We also find it useful to define the set $\mathcal{J}_k = \bigcup_{i=1}^k j_i$, with $\mathcal{J}_0 = \emptyset$. In the following we will indicate with \mathcal{R} the residuals computed by applying the decomposition path \vec{g}_{γ} to \mathcal{I} , while we will indicate with \mathcal{R}' the residuals obtained by applying the same decomposition path to \mathcal{I}' . A similar notation applies to the other symbols we have defined. Let now c_k be a non-zero element of \vec{c} . We surely have $S(g_{\gamma_k}) \cap \mathcal{J}_{k-1} = \emptyset$ since otherwise we would have $c_k = 0$. Let us show first that by applying the MP to \mathcal{I}' the coefficients of the atoms g_{γ_h} with h < k do not change. Without loss of generality let h be the first element for which c_h may have changed. Two cases are possible: $S(g_{\gamma_k}) \cap S(g_{\gamma_h}) = \emptyset$ or $S(g_{\gamma_k}) \cap S(g_{\gamma_h}) \neq \emptyset$. In the first case it is evident that the weight c_h can not change, since a modification of the weight assigned to g_{γ_k} cannot have any impact on (8.8) given that the minimization is performed on $S(g_{\gamma_h})$.

When the intersection between $S(g_{\gamma_h})$ and $S(g_{\gamma_k})$ is non-empty the proof is split in two parts, the former considers the case $c'_k > c_k$, the latter the case $c'_k < c_k$. When $c'_k > c_k$ some of the values in \mathcal{R}'^{h-1} are increased, however $\mathcal{R}'^{h-1}(j_h)$ does not change since $S(g_{\gamma_k}) \cap \mathcal{J}_{k-1} = \emptyset$, hence leaving the choice of j_h and the computation of the weight c_h unchanged.

If $c'_k < c_k$, some values in \mathcal{R}'^{h-1} are decreased while leaving $\mathcal{R}'^{h-1}(j_h)$ unchanged. However, $\forall (x, y) \in S(g_{\gamma_k}) \cap S(g_{\gamma_h})$ we have $\mathcal{R}^{k-1}(x, y) \leq \mathcal{R}^h(x, y)$ since due to the particular update rule we adopted, at each iteration the values in the residual cannot increase. For this reason at the *h*-th selection step, the modification of the *k*-th coefficient cannot decrease the residual by more than $\mathcal{R}^{h-1} - c_h$ (remember that $c_h = \mathcal{R}^{h-1}(j_h)$). In other words, $\mathcal{R}'^{h-1}(x, y)$ computed on the modified image \mathcal{I}' will satisfy the relation $\mathcal{R}'^{h-1}(x, y) \geq \mathcal{R}'^{h-1}(j_h)$ hence ensuring that $c'_h = c_h$.

We must now show that the components $h \ge k$ of the vector \vec{c} do not change as well. Let us start with the case h = k. Since no coefficient has changed until position k, when the MP is applied to the image \mathcal{I}' we have

$$c_k'' = \min_{(x,y)\in S(g_{\gamma_k})} \left[\mathcal{R}^{k-1}(x,y) + (c_k' - c_k)g_{\gamma_k}(x,y) \right].$$
 (8.10)

From equation (8.10) it is evident that

$$c_k'' = c_k' = \min_{(x,y) \in S(g_{\gamma_k})} \mathcal{R}^{k-1}(x,y),$$
(8.11)

since the term $(c'_k - c_k)g_{\gamma_k}$ introduces a constant bias on all the points of $S(g_{\gamma_k})$.

As to the case h > k it is trivial to show that $c'_h = c_h$ given that the residual after the k-th step will be the same for \mathcal{I} and \mathcal{I}' .

Theorem 1 can be applied recursively to deal with the case in which more than one coefficient in \vec{c} is changed. In the following we show how the stability result stated in Theorem 1 can be used to build the MPSteg-color algorithm.

Chapter 9

A closer look at the new MP domain

One may wonder whether the particular dictionary, selection and update rules we used, which are the result of the requirements set in the previous chapter, maintain the compaction properties of high-redundant basis. This is indeed the case as it is witnessed by Figure 9.1 and exemplified in Figures 9.2, 9.3, and 9.4. Specifically, in Figure 9.1 the reconstruction error is plotted (in log scale) as a function of the number of basis elements considered for the reconstruction (the results have been obtained by averaging the plots relative to 25 images), as it can be seen when very few coefficients are used the DCT decomposition performs better. This is due to the decision we made to design the update rule in such a way that the residual image is always positive (while the DCT coefficients are chosen in such a way to minimize the error energy). However, when the number of basis elements increases the MP capacity of fully describing the image with a lower number of elements is evident. Indeed in the DCT case all the 16 coefficients of the orthogonal basis are needed to bring the reconstruction error to zero, while in the MP case only 9.63 atoms are needed (on the average).

From a different perspective, the higher semantic level MP operates at is exemplified in Figures 9.2, 9.3, and 9.4. The original image (Figure 9.2) is first decomposed by applying a 4×4 DCT and reconstructed by using only the DC and the first AC coefficient, yielding the result depicted in Figure 9.3. The same approach is applied in Figure 9.4 where the image is generated by using only the first 2 atoms of the MP decomposition. Though the reconstruction error is larger in the MP case (in accordance with the plot of Figure 9.1), the perceived quality of the image obtained through MP decomposition is better than that obtained with DCT, since the selected atoms permit to better represent the geometric structures contained in the image.



Figure 9.1: Comparison between the compaction property of the DCT and MP domains.



Figure 9.2: Original gray-scale image.



Figure 9.3: Reconstructed image by using the first 2 DCT coefficients in a zig-zag ordering for each 4×4 block.



Figure 9.4: Reconstructed image by using 2 atoms for each 4×4 block.

Chapter 10

MPSteg-color

In this chapter we give a detailed description of the MPSteg-color algorithm. We first introduce the main structure of the algorithm, then we describe how we can achieve security against targeted steganalyzers and increase the stego-message payload.

Theorem 1 ensures that by using the selection rule described in equations (8.1) through (8.4), it is possible to correctly write and read a message hidden in the MP coefficients if the decomposition path \vec{g}_{γ} is known. In order to cope with decomposition path instability, we exploit the availability of three color bands. To explain how, let us introduce the following notation:

$$\mathcal{I} = \left(\begin{array}{c} \mathcal{I}_r \\ \mathcal{I}_g \\ \mathcal{I}_b \end{array}\right)$$

where $\mathcal{I}_r, \mathcal{I}_g$ and \mathcal{I}_b are the RGB bands of a traditional color image.

MPSteg-color works on a non-overlapping, 4×4 block-wise partition of the original image, however, for simplicity we continue to refer to image decomposition instead of block decomposition, the use of blocks, in fact, is only an implementation detail, not a conceptual strategy.

The main idea behind MPSteg-color is to use the correlation of the three color bands to stabilize the decomposition path. Specifically the decomposition path is calculated on a color band and then used to decompose the other two bands (the validity of such an argument will be tested in Section 11.2.1). Due to the high correlation between color bands, we argue that the structural elements found in a band will also be present in the other two. Suppose, for instance, that the decomposition path is computed on the \mathcal{I}_r band, we decompose the original image as follows

$$\mathcal{I} = \begin{pmatrix} \sum_{k=1}^{n} c_{r,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_{r}^{n} \\ \sum_{k=1}^{n} c_{g,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_{g}^{n} \\ \sum_{k=1}^{n} c_{b,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_{b}^{n} \end{pmatrix}$$
(10.1)

where $g_{\gamma_{r,k}}$ are the atoms selected on the red band, $c_{r,k}, c_{g,k}$ and $c_{b,k}$ are the atom weights of each band and $\mathcal{R}_r^n, \mathcal{R}_g^n$ and \mathcal{R}_b^n are the partial residuals. By using eq. (10.1) we do not obtain the optimum decomposition of \mathcal{I} for the green and blue bands, but this decomposition has a good property: if the red band is not modified then the decoder may apply the selection function $\mathcal{S}(\cdot)$ to the red band and use it to retrieve the decomposition path used by the embedder to hide the message in the other two bands.

In Section 10.2 we cope with the security aspect by adding for each block a random choice between the reference and embeddable bands.

By assuming, for instance, that the decomposition path is computed on the red band, then MPSteg-color can embed the stego-message by operating on the vector with the decomposition weights of the green and blue bands, i.e. the vector

$$\vec{c}_{gb} = (c_{g,1}, c_{b,1}, \dots, c_{g,n}, c_{b,n}).$$
 (10.2)

According to Theorem 1, we know that the stego-messages can be correctly embedded by changing the coefficients of the MP decomposition vector \vec{c}_{gb} , however, for this result to hold it is necessary that only non-zero coefficients are modified. In fact, given that the decomposition path is computed on one band and the message embedded in the other two, it may be the case that the coefficients of some atoms of the decomposition path are zero, i.e. the vector \vec{c}_{gb} may contain some null coefficients. This issue will be considered in the next section, where the embedding rule used by MPSteg-color is described.

10.1 Embedding Rule

We now describe the embedding rule used to embed the stego-message within \vec{c}_{gb} . Given that the coefficients of \vec{c}_{gb} are non-negative integers, we can apply any method that is usually applied to embed a message in the pixel domain. However, we must consider that the embedder cannot modify zero coefficients (due to Theorem 1 assumptions), but in principle it could set to zero some non-zero coefficients. If this is the case a de-synchronization would be introduced between the embedder and the decoder since the decoder will not know which coefficients have been used to convey the stego-message. In the steganographic literature this is known as the channel selection problem, for which an elegant solution exists, namely the Wet Paper Code strategy introduced by Fridrich *et al.* in [11]. However, our aim was to analyze the capability of the MP domain as a cover domain, hence will not consider any procedure to redirect the embedding changes of the basic MPSteg algorithm¹. In fact, the same procedures could be applied to pixel domain methods, and are not related to the particular domain in which the message is embedded.

For this reason, we adopted the standard ± 1 embedding, that is described in Chapter 3.1, to embed the message in the non-null weights.

In order to avoid the channel selection problem, we add 2 to all the coefficients for which the ± 1 embedding rule yields a null value. By indicating with

$$\vec{c}_{gb}^w = (c_{g,1}^w, c_{b,1}^w, \dots, c_{g,n}^w, c_{b,n}^w)$$

the marked coefficient vector, then we build the stego image \mathcal{I}^s :

$$\mathcal{I}^{s} = \begin{pmatrix} \sum_{k=1}^{n} c_{r,k} \cdot g_{\gamma_{r,k}} + \mathcal{R}_{r}^{n} \\ \sum_{k=1}^{n} c_{g,k}^{s} \cdot g_{\gamma_{r,k}} + \mathcal{R}_{g}^{n} \\ \sum_{k=1}^{n} c_{b,k}^{s} \cdot g_{\gamma_{r,k}} + \mathcal{R}_{b}^{n} \end{pmatrix}.$$
(10.3)

¹Similarly we will not consider matrix embedding [10], since it can be used to boost the performance of any steganographic scheme, regardless of the embedding domain.

While the application of ± 1 embedding rule to MP coefficients guarantees that the modified coefficients lie in the [0,255] interval, it is possible that some pixels of the reconstructed image exceed the 255 limit. If this happens, the coefficients larger than 2 are decreased by 2 until the overflow error disappears. In this way the embedding distortion is slightly augmented, however, such an effect is completely negligible since overflow errors are extremely rare.

10.2 Improving undetectability

While the undetectability of the above scheme against general purpose steganalyzers can be easily proved [8], undetectability against targeted steganalysis may be a problem. First of all, if the dictionary is assumed to be known, a steganalyzer may look for specific artifacts introduced by MPSteg-color directly in the MP domain. Secondarily, even if the dictionary is kept secret, the particular nature of atoms and the application of the MP algorithm at a block level, may introduce blocking artifacts that could be used by a targeted steganalyzer to detect the presence of a stegomessage. As it will be shown in section 11.3 this is indeed the case, hence some countermeasures need to be taken.

First of all we decided to avoid using the first decomposition coefficient as support of the secret message. Usually such a coefficient is able to describe most of the image energy compared to the remaining atoms. For this reason, any modification to the first atom is likely to introduce significant blocking artifacts, hence we decide to keep such an atom unchanged.

The second and more important countermeasure we took, is randomization of the embedding process. Randomization is applied at two different levels. At the first level randomization affects the image decomposition into blocks. By following an approach similar to that proposed by Solanki *et al.* in [63], the image is partitioned into disjoint and contiguous windows of size 5×5 or 6×6 , and MP decomposition is applied to 4×4 blocks randomly chosen within the larger 5×5 (or 6×6) windows². By doing so, we reduce and randomize the blocking artifacts introduced

 $^{^{2}}$ Randomization is achieved by changing the offset of the 4x4 window within the larger 5x5 or 6x6

by MPSteg-color that will be more difficult to detect. In addition, even by knowing the MP dictionary, the MP domain used by a possible adversary will be spatially de-synchronized with respect to the one used by the embedder, thus making steganalysis in the MP domain more difficult. Of course a compromise between payload and undetectability must be found here, given that the larger the window size the better the undetectability at the expense of payload (given that the number of pixels not touched by MPSteg-color will increase).

The second randomization level regards the choice of the reference color band that is used to calculate the MP decomposition path. Specifically, a secret key is used as a seed for a random number generator that decides on a block by block basis which color band is used to calculate the decomposition path. The MP decomposition is applied to the chosen band, while the secret message is embedded within the other bands.

As it will be seen in Chapter 11, through randomization, especially block position randomization, it is possible to resist to attacks brought by targeted steganalysis.

10.3 Increasing the payload

An undesirable effect of block position randomization is that the payload is (slightly) decreased, all the more that the capacity³ of MP domain is intrinsically lower than that of the spatial domain (see [57, 8]). A possible way to improve (slightly) the payload of messages hidden by MPSteg-color stems from the observation that though the color bands are highly correlated, the decomposition path calculated on one of them in general is not able to lead to a zero residual on the other two bands. For some of the atoms selected in the reference band, in fact, a null coefficient is obtained in the other bands, thus diminishing the number of coefficients available for embedding. For this reason, after that the decomposition path computed on the reference band is applied to the other two bands, the residual of one of the these two bands is further decomposed to provide an additional list of

window.

³We are using the term capacity in a loose sense, without any reference to the corresponding information theoretic concept.

atoms that are used on the remaining band to provide additional coefficients to embed some more bits. In the sequel we will refer to this second decomposition step as the *decomposition refinement step*. The actual payload increase obtained thanks to the decomposition refinement step will be evaluated experimentally in the next chapter.

Chapter 11

MPSteg-color: experimental results

In this chapter we report experimental results that demonstrate the undetectability of the new MPSteg-color algorithm and validate the main assumptions behind it. First of all in Section 11.1 the image database used for the experiments is described. Afterwards, in Section 11.2 we take a closer look at the MP domain to support the hypothesis that the decomposition path calculated in one color band can be used with little loss for the other bands. We also evaluate the gain in terms of payload that is brought by the decomposition refinement step.

After that, in Section 11.3, we carefully analyze the undetectability of the proposed technique, with particular attention to the effectiveness of partition randomization as a countermeasure to targeted steganalysis. For this reason the undetectability of the stego-message is tested first again two targeted steganalyzers explicitly developed to detect MPSteg-color messages, then against general purpose steganalyzers.

11.1 Image Database

For the experimental validation we used a database of 2564 raw color images of 512×512 size, which is a color version subset of the camera database described in 5.1.

Images are the cropped version of the original ones which are taken in a RAW format from several kinds of common cameras. The images in the database show a wide range of scenarios including countryside, houses, people, faces, man-made objects, etc.

11.2 Effectiveness of the proposed MP decomposition

We first validate the conjecture that, due to the correlation between RGB color bands, computing the decomposition path on one band and using it on the other two does not impair the capability of the MP algorithm to extract the most important features of image blocks. Moreover we give a measure of the payload allowed by the MP domain and the payload gain allowed by the decomposition refinement step. On one side this is a good result showing a high degree of correlation, on the other side it shows that the decomposition path calculated on one band is not capable of fully describe the content of the other bands, thus justifying the resort to a decomposition refinement step.

11.2.1 Interband correlation of decomposition path

MPSteg-color relies on the assumption that the color bands are highly correlated. To experimentally validate the above conjecture, we decomposed a random color band until a null residual is obtained, then with the same decomposition path we decomposed one of the remaining bands. After this second decomposition, we usually obtain a non-null residual that will be null only if the decomposition path calculated on the first band fits the content of the second band. At this point we applied a matching pursuit decomposition to the non-null residual and we measured its length. By averaging the results obtained on all the images of the test database, we found that about 3.7 additional atoms are needed to decompose the second and the third band residuals whose energy is about 40,80dB (while about 9.63 atoms were necessary for the reference band).

11.2.2 Effectiveness of the decomposition refinement step

The goal of the decomposition refinement step is to decompose the residuals of the two remaining bands after that the decomposition path computed on the reference band is applied to them. In this way some extra non-zero coefficients are obtained thus contributing to increase the payload of MPSteg-color. Specifically, we found that the number of available coefficients for embedding is increased by 12.29% on

average. In terms of payload this means that if we embed one bit per non-null coefficient then we are able to increase the size of the secret message by a 12.29% factor.

11.3 Undetectability analysis

The most important requirement for any steganographic technique is undetectability. In this section, we report the results that we obtained by applying four state-ofthe-art steganalyzers to detect ± 1 embedding applied in the MP domain and the pixel domains. Before doing that, however, we test the effectiveness of block partition randomization to combat targeted steganalyzers. In the following, we briefly describe the steganalyzers we used by grouping them into two main sets.

The first set comprises target steganalyzers. It will be used to show the weakness of MPSteg-color without the block-windows randomization. The second set of steganalyzers is composed by steganalyzers proposed until now.

All the steganalyzers are used as feature extractors, however, we decide to always use a simple linear classifier, namely the Fisher Linear Discriminant (FLD) that is described in Chapter 2.3, to compare the goodness of each tool even though in the original version some of them are associated with an SVM classifier. We chose to compare all the steganographic algorithms by using a FLD classifier in order to highlight the capability of the various types of features to detect the presence of a hidden MPSteg message.

11.3.1 Targeted steganalyzers

The first targeted steganalyzer we used is built on the simple blocking artifacts detector (BD) described in [64]. This technique was originally developed for detecting JPEG block artifacts, however, we adapted it to detect the artifacts introduced by MPSteg-color and use them as a feature to detect the presence of a stego-message. The algorithm is very simple: we split the image into blocks whose size should be matched to that used by the MP algorithm. Regardless of the block partition strategy the steganalyzer assumes that blocks are located on a grid aligned with the top-left



Figure 11.1: For each block the numbers Z' = |A + D - B - C| and Z'' = |E + H - F - G| are computed.

corner of the image. For each block we calculate Z and Z' as follows:

$$Z' = |A + D - B - C|$$
$$Z'' = |E + H - F - G|$$

where A, B, C, D, E, F, G and H are taken as shown in Figure 11.1 in the case of 4×4 blocks, the extension to larger blocks being trivial. Next the normalized histograms vectors h'(n) and h''(n) are computed respectively for Z' and Z'' and the following feature is calculated:

$$\mathbf{f}_{BD} = \sum_{n=0}^{255} |h'(n) - h''(n)|.$$

The above procedure is repeated for the three color bands producing a three-dimensional feature vector that is given as input to the FLD classifier.

The second steganalyzer we developed relies on the knowledge of the histogram of MP coefficients. For this to be possible, we assume that the steganalyzer knows the MP dictionary but it does not know the reference band that is used to calculate the decomposition path (hence a random band is used as a reference by the steganalyzer). Figure 11.2 shows a typical histogram of a cover image and a stego MPSteg-color



Figure 11.2: Comparison between coefficients histogram of a cover image (dashed line) and a stego MPSteg-color image (solid line).

image. Due to the embedding asymmetry applied to coefficients having value equal to 1 - that are either left unchanged or incremented by one - a flat step appears in the leftmost part of the histogram, while this effect does not appear in the cover image. By considering this effect, we propose to use the following feature:

$$\mathbf{f}_{MPHA} = h(2) - \frac{h(1) + h(3)}{2} \tag{11.1}$$

where h is the histogram function. In the sequel we will refer to this technique as MPHA.

11.3.2 State-of-art steganalyzers

The first steganalyzer of the second group is ALE based on the artifacts introduced by ± 1 embedding in the image histogram and described in Chapter 4.

The second algorithm we used in this set is WAM steganalyzer [10] that we described in Chapter 3.2.1. It works in the wavelet domain and the extracted features

are central moments that are calculated in the three detail bands of first order wavelet decomposition. This steganalyzer is a blind steganalyzer because it is not explicitly developed to detect any particular kind of messages.

The third steganalyzer is 2D-HCFC algorithm introduced by Ker in [23] and described in Chapter 3.2.2. It builds on some considerations made in [36] about artifacts generated in the histogram domain by ± 1 embedding. In particular we used the concatenated features from the histogram analysis and the adjacency matrix analysis.

Starting from the initial gray scale steganalyzers, we implemented a color version by joining the 3 RGB band feature vectors in a unique vector with triple components. In this way we worked with 30 features for ALE, three features for 2D-HCFC and 81 features for WAM.

11.3.3 Steganalysis Results

For our experiments we embedded in each image a random message by using a secret unique key.

For MPSteg-color we used three window sizes in the experimental tests: 4×4 , 5×5 and 6×6 . The comparison between different methods was always made by using the maximum payload allowed by the techniques involved in the comparison, for instance when comparing MPSteg-color versions with different window sizes the payload imposed by the largest window is used ¹.

The cover and stego images produced as described above were used to build a training and a test set, both containing 50% cover and 50% stego images. The size of the training set was equal to 20% of the 2564 images, the remaining 80% forming the test set. The training and the test sets were built randomly, however, to avoid any dependence of the results upon the specific training and test sets, the experiments were repeated 20 times, each time with a different training and test set. In this way we obtained 20 ROC curves that were vertically averaged to produce the final plots shown in the following. In the plots the minimum and maximum bound of the beam of ROC curves is shown.

¹The payload is expressed in bit per pixel, by considering $512 \times 512 \times 3$ the number of the pixels.



(b) MPHA - MPSteg-color payload: 0.3159 bpp

Figure 11.3: Comparison between BD (a) and MPHA (b) steganalyzers: the first window size reported in the figure indicates the window size used by the steganalyzer, the second one the size of the blocks used by the embedder to partition the image. The payload is 0.3159 bpp.

| Payload | MPSteg-color | | | ± 1 embedding |
|----------|---------------------|--------------------|---------------------|-------------------|
| | 6×6 window | 5×5 window | 4×4 window | |
| 0.14 bpp | 55.20 | 55.25 | 55.30 | 59.70 |
| 0.20 bpp | - | 53.62 | 53.69 | 58.12 |
| 0.32 bpp | - | - | 51.64 | 56.14 |

Table 11.1: PSNR values (dB) for different payload and settings.

Figure 11.3 shows the performance of the two target steganalyzers described in Section 11.3.1. We considered several scenarios: in a first optimistic (for the steganalyzer) situation the steganalyzer knows the window size used by the embedder, though it does not know the particular randomization key used. In this case the steganalyzer simply picks a random 4x4 block out of the larger window, with a probability of guessing the right position equal to 1/4 for 5×5 windows and 1/9 for 6×6 windows. In the second case the steganalyzer is tuned on 4×4 blocks, but the actual partitioning window used by the embedder is larger.

The results shown in the Figure 11.3 indicate that both BD and MPHA steganalyzers perform better when no block randomization is used (MPSteg-color window size = 4×4) and the steganalyzers are tuned on the a 4×4 window size. At the same time, the undetectability of MPSteg-color increases dramatically when block position is randomized, even when the steganalyzers are tuned to the exact window sizes. This is especially true for the BD steganalyzer whose performance are bad already for a 5×5 randomization window and very close to a random guess for the 6×6 case. Since the MPHA performs slightly better, we also investigated its performance when it does not adopt the correct window size (a case that is closer to reality since in general the steganalyzer may not know the size of the randomization window used by the embedder). Specifically part (b) of Figure 11.3 reports the case of an MPHA steganalyzer tuned on a 5×5 or 6×6 window applied to stego-images containing messages embedded by using a 4×4 partition. As it can be seen the performance of the steganalyzer decreases significantly. Similar results are obtained when the embedder uses a larger randomization window. We now turn the attention to non-targeted steganalysis and to the comparison between MPSteg-color and ± 1 embedding applied in the pixel domain

Before presenting the ROC curves, it is instructive to consider the PSNR obtained by applying ± 1 -steg in the pixel and in the MP domains. Such results are given in Table 11.1 for different MPSteg-color window sizes and different payloads. The average PSNR is obtained by taking the average on the linear quantities and then passing to the logarithmic scale. As expected, by considering that the atoms of the MP decomposition has a support larger than a single pixel, MPSteg-color results in a lower PSNR, hence suggesting that any advantage in terms of undetectability (if any) will be due to the better hiding properties of the MP domain.

Despite the lower PSNR, the presence of the stego-messages can not be noticed perceptually as it is exemplified in Figure 11.4 where the stego-image (right) cannot be distinguished from the original one (left) even if the largest possible payload is used (0.3687bpp) for a PSNR of 51.22dB.

Figures 11.5, 11.6, and 11.7 compares the detectability of MPSteg-color with that of ± 1 embedding, for three different window sizes (and different payloads). In the legend, the Area Under Curve (AUC) value is also given for each steganalyzer as an overall measure of classification accuracy.

We can see that WAM and ALE are capable to distinguish the stego-images with a significant level of accuracy. In WAM case, though, the message embedded in the MP domain is less detectable than the one embedded in the pixel domain, while ALE works better with MPSteg-color than ± 1 embedding.

We can see that WAM is the only steganalyzer capable to distinguish the stegoimages with a significant level of accuracy. Even in this case, though, the message embedded in the MP domain is less detectable than the one embedded in the pixel domain.

Slightly better results (from the steganalyzer point of view) are obtained for a 4×4 window (larger payload), however, the general behavior of the various algorithms does not change.

In order to evaluate the dependence of MPSteg-color detectability on the size of the randomization window, the ROC curves obtained for different sizes are plotted Table 11.2: Average execution time in seconds of embedding phases for images 512×512 of size, window 4×4 and full payload (0.32 bpp).

| Decomposition | Embedding | Reconstruction |
|---------------|-----------|----------------|
| 13830 | 14.78 | 2.5 |

in Figures 11.8, 11.9, and 11.10. In this case we pay our attention to a specific steganalyzer, and we use the maximum admissible payload for all the used windows (i.e. those attainable with the 6×6 windows) that is 0.1391 bpp. The ALE, WAM and 2D-HCFC steganalyzers are respectively shows in Figures 11.8, 11.9, and 11.10. We see in Figure 11.10 that 2D-HCFC steganalyzer is not able to detect MPSteg-color. Instead, the performance of ALE and WAM steganalyzers do not depend on the size of the partitioning window. A possible explanation for this behavior is that for the 6×6 case we are using the maximum admissible payload, hence approximately half of the MP coefficients are changed, while this is not the case with the 4×4 window. In addition, the additional randomization allowed by the 6×6 window is a way to improve the undetectability against targeted steganalyzers - as it is shown in Figure 11.3 - explicitly designed to detected a message embedded in the MP domain, the same advantage is not expected for other steganalyzers.

11.3.4 Computational Complexity

Although particular attention has been paid to reduce the execution time, the MP exhaustive search to define the decomposition path at each step is really onerous and it is the bottleneck of the whole system. We developed the prototype of our scheme in MATLAB and we used a c-MEX function in the kernel of exhaustive search in order to reduce as much as possible the computational time. Table 11.2 shows the execution time of the embedding phase (decomposition step, message embedding and image reconstruction) when the MATLAB code is executed on an Intel Xeon at 3GHz.

Even though the source code could be improved and a different language could be chosen, the decomposition step - that is used to the receiver side too - is the most critical part of the proposed steganography.



(a) Cover image



(b) Stego image

Figure 11.4: Perceptual invisibility of the stego-message. The cover (a) and the stego (b) images can not be distinguished (payload = 0.3158 bpp, 4×4 partition, 51.40dB).


Figure 11.5: Comparison between MPSteg-color with window 4×4 (solid line) and ± 1 embedding (dashed line) with 3 different steganalyzers at 0.3159 bpp of payload.



Figure 11.6: Comparison between MPSteg-color with window 5×5 (solid line) and ± 1 embedding (dashed line) with 3 different steganalyzers at 0.2002 bpp of payload.



Figure 11.7: Comparison between MPSteg-color with window 6×6 (solid line) and ± 1 embedding (dashed line) with 3 different steganalyzers at 0.1391 bpp of payload.



Figure 11.8: MPSteg-color detection performance on ALE at fixed embedding rate (0.1391 bpp).



Figure 11.9: MPSteg-color detection performance on WAM at fixed embedding rate (0.1391 bpp).



Figure 11.10: MPSteg-color detection performance on 2D-HCFC at fixed embedding rate (0.1391 bpp).

Chapter 12

MPSteg-color: remarks and future works

In the second part of the thesis a new algorithm for embedding a stego-message into color images represented by means of high redundant basis decomposition has been presented. The problems of previous schemes proposed in this sense have been solved, with particular attention to undetectability against targeted steganalyzers. Indeed, we have shown that without proper countermeasures, messages hidden in the MP domain are easily detectable.

The undetectability of MPSteg-color has been extensively tested against both targeted and general purpose steganalyzers, showing the validity of the proposed approach. In particular, the good hiding properties of the MP domain are demonstrated by comparing the undetectability of a ± 1 embedding message embedded in the pixel with that of a ± 1 embedding message embedded in the MP domain, with the latter being less detectable than the former despite a higher embedding distortion.

Further experimental investigations are needed in order to apply the full methodology benchmark proposed in the first part of the thesis. Actually, we have not been able to build a full MPSteg-color dataset with several payloads, due to the computational complexity of the proposed technique.

Some further studies can consider the role of the dictionary, by analyzing the undetectability dependence on the used atoms in order to design a powerful dictionary which is more able to make MPSteg-color undetectable. By enlarging the cardinality of the dictionary, we could also study a randomized dictionary, i.e. a subset of a very big dictionary, which can be unknown to the steganalyzer. In fact, a weakness of MPSteg-color is that the steganalyzer knows the dictionary in which the message is embedded and this fact can help in the design of a targeted method.

Another weakness of the proposed approach is that we are loosing one band just

to correctly recover the decomposition path. In this way we lose about 1/3 of coefficients which cannot be used as embedding support and the MPSteg-color maximum payload goes down. As future works it could be possible to develop a new scheme with a blind decomposition path recovery or by embedding the message within the atom indexes instead of the atom weights. In this scenario, a great attention should be given to synchronizing problems.

A few additional improvements of the proposed scheme are possible, either to augment the payload or diminish the detectability. Specifically, the wet paper coding approach may be applied to remove the constraint that message embedding cannot produce zero coefficients, and matrix embedding can be applied to decrease the embedding distortion.

Chapter 13

Final remarks

In this thesis we have taken into account steganography and steganalysis in the pixel domain. Pixel domain steganography has been deeply investigated by many research groups the last ten years so we can consider it as a rather mature field. Several application scenarios could be interested to use either steganographies or steganalysis techniques for digital images especially thanks to the widest connection infrastructure: Internet. Even though steganography is usually linked to malevolent applications such as industrial espionage or coordination between terroristic cells and steganalysis as a benevolent tool thought to reinforce homeland security, these are not the unique application scenarios. For example in countries suffering from military dictatorship, steganography can be seen as the only way of ensuring freedom of speech while steganalysis need each other to avoid the supremacy of the adversary and for this reason research has lately been directed towards the investigation of the ultimate limits of these techniques [45].

Today several steganographic and steganalysis methods ensure good results especially in controlled scenarios with fixed payload, image sources, and image sizes but they usually do not extend these good performance to practical case. Moreover, in the literature the performance of steganographic techniques are rarely based on a test set never analyzed by the classifier. In the last years, to obtain undetectability the steganographic methods have been mainly concerned with the minimization of embedding changes, but this methodology is not the unique possible strategy and researchers should investigate other strategies as well.

In the above framework, the contribution of this thesis is threefold. The first one is a new ± 1 embedding steganalyzer, called ALE, which is able to detect ± 1

embedding artifacts. The second one is the proposal of a comparison methodology to be applied to fairly benchmark the merits and drawbacks of different steganalyzers. Finally, the last one is a new steganographic method, called MPSteg-color which shows how embedding the hidden message at a higher semantic level could result in a less detectable stego message than a strategy that minimizes the embedding changes without takes into account the embedding domain.

The first part of the thesis focuses on the importance of reproducible research in terms of performance validation and weakness analysis. A lot of works describe steganography and steganalysis tools in a good way, however the reader has always difficulties to implement again these tools especially because the experimental results are not obtained by using a standard procedure. We have shown in this thesis that a proper design of the image database is a crucial path to obtain reproducible results. Moreover, recent studies [25] show that by fixing the payload, the results change through a square root law depending on image size and could be interesting to extend our analysis by taking into account the image size.

Our analysis has also shown that when a steganalyzer does not know the payload, its performance in terms of AUC are far from those obtained in the best case in which they are known and this fact constitutes a big weakness of the current state of art. In practical applications, in fact, a steganalysis system can check the image size, and by using a forensic tool it can get information about the image sources (camera or scanner), but it never knows the message length.

Besides, we have shown that concatenating good features in a new steganalyzer, as hybrid steganalyzers do, it is usually a good way to improve performances, but if we do not know the practical scenario, i.e. the steganalyzer is trained with a train set which is very different than the future test set, the concatenation may not be the best strategy: it just increases the classification uncertainty. The ALE, the steganalyzer proposed in this thesis, seems to be more stable in terms of overall AUC performances than WAM, which is the current state of art steganalyzer in the pixel domain thanks to the low number of features it uses which are about one third of those used by WAM.

In the second part of this thesis we considered the steganography point of view

by showing that the weakness of steganalyzers could be used by steganographers to develop more efficient and secure techniques. Specifically, we have proposed a new technique based on high redundant basis decomposition that shows how a embedding the message at a higher semantic level could result in a less undetectable stego message, when the steganalyzer is based on high-order statistics analysis. Of course, the proposed technique is just a prototype, however it clearly shows that embedding messages at a more semantic level could be a good way to achieve undetectability. In the future, we could extend our research by fully analyzing the MP embedding domain, especially by investigating the relationship between detectability and the design and cardinality of the dictionary.

Bibliography

- M. Kharrazi, H. Sencar, and N. Memon, "Image Steganography: Concepts and Practice," *Lecture Note Series, Institute for Mathematical Sciences, National University of Singapore*, 2004.
- [2] I. Cox, T. Kalker, G. Pakura, and M. Scheel, "Information transmission and steganography," *Lecture Notes in Computer Science*, vol. 3710, p. 15, 2005.
- [3] C. Shannon, "Communication Theory of Secrecy Systems," *Bell System technical Journal*, vol. 28, pp. 656–715, 1954.
- [4] G. J. Simmons, "The prisoners' problem and the subliminal channel," in *Advances in Cryptology: Proceedings of CRYPTO'83*. Plenum Pub Corp, 1984, pp. 51–67.
- [5] R. Givner-Forbes, "Steganography: Information Technology in the Service of Jihad," *The international Centre for Political Violence and Terrorism Research*, March 2007. [Online]. Available: www.pvtr.org
- [6] R. Anderson, "Stretching the limits of steganography," *Lecture Notes in Computer Science*, vol. 1174, pp. 39–48, 1996.
- [7] WikipediA The Free Encyclopedia. [Online]. Available: http://en.wikipedia.org/wiki/Printer_steganography
- [8] G. Cancelli and M. Barni, "MPSteg-color: A new steganographic technique for color images," *Information Hiding: 9th International Workshop, Ih 2007, Saint Malo, France, June 11-13*, vol. 4567, pp. 1–15, 2007.

- [9] A. Westfeld, "F5-a steganographic algorithm: High capacity despite better steganalysis," in *Information Hiding: 4th International Workshop, IH 2001, Pittsburgh, PA, USA, April 25-27, 2001: Proceedings.* Springer, 2001, p. 289.
- [10] M. Goljan, J. Fridrich, and T. Holotyak, "New blind steganalysis and its implications," *Proceedings of SPIE*, vol. 6072, pp. 1–13, 2006.
- [11] J. Fridrich, M. Goljan, P. Lisonek, and D. Soukal, "Writing on wet paper," *IEEE Transactions on Signal Processing*, vol. 53, no. 10 Part 2, pp. 3923–3935, 2005.
- [12] J. Fridrich, M. Goljan, and D. Hogea, "Attacking the outguess," in Proc. of the ACM Workshop on Multimedia and Security, 2002.
- [13] P. Sallee, "Model-based methods for steganography and steganalysis," *International Journal of Image and Graphics*, vol. 5, no. 1, pp. 167–189, 2005.
- [14] J. Fridrich, T. Pevnỳ, and J. Kodovskỳ, "Statistically undetectable jpeg steganography: dead ends challenges, and opportunities," in *Proceedings of the 9th workshop on Multimedia & security.* ACM New York, NY, USA, 2007, pp. 3–14.
- [15] B. Roue and J. Chassery, "Improving LSB steganalysis using marginal and joint probabilistic distributions," in *Proceedings of the 2004 workshop on Multimedia and security.* ACM New York, NY, USA, 2004, pp. 75–80.
- [16] S. Dumitrescu, X. Wu, and Z. Wang, "Detection of LSB steganography via sample pair analysis," *IEEE transactions on Signal Processing*, vol. 51, no. 7, pp. 1995–2007, 2003.
- [17] J. Fridrich and M. Long, "Steganalysis of LSB encoding in color images," in 2000 IEEE International Conference on Multimedia and Expo, 2000. ICME 2000, vol. 3, 2000.
- [18] A. Ker, "Quantitative evaluation of pairs and RS steganalysis," Security, Steganography, and Watermarking of Multimedia Contents VI, vol. 5306, pp. 83–97, 2004.
- [19] P. Lu, X. Luo, Q. Tang, and L. Shen, "An improved sample pairs method for detection of LSB embedding," in *Proc. 6th Information Hiding Workshop*, vol. 3200. Springer, 2004, pp. 116–127.
- [20] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Transac*tions on Information Forensics and Security, vol. 1, no. 1, pp. 111–119, 2006.

- [21] R. Bohme and A. Westfeld, "Exploiting preserved statistics for steganalysis," in *Sixth Workshop on Information Hiding, Toronto, Canada (2004, May).* Springer, 2004.
- [22] J. Fridrich, M. Goljan, D. Hogea, and D. Soukal, "Quantitative steganalysis of digital images: estimating the secret message length," *Multimedia Systems*, vol. 9, no. 3, pp. 288–302, 2003.
- [23] A. D. Ker, "Steganalysis of LSB matching in grayscale images," *IEEE Signal Process-ing Letters*, vol. 12, no. 6, pp. 441–444, 2005.
- [24] A. Ker, T. Pevný, J. Kodovský, and J. Fridrich, "The square root law of steganographic capacity," in *Proceedings of the 10th ACM workshop on Multimedia and security*. ACM New York, NY, USA, 2008, pp. 107–116.
- [25] T. Filler, A. Ker, and J. Fridrich, "The square root law of steganographic capacity for Markov covers," in *Proceedings of SPIE*, vol. 7254, 2009, p. 725408.
- [26] J. Zhang, I. J. Cox, and G. Doërr, "Steganalysis for LSB Matching in images with high-frequency noise," *IEEE 9th Workshop on Multimedia Signal Processing (MMSP)*, 2007, pp. 385–388, 2007.
- [27] R. Duda, P. Hart, and D. Stork, Pattern classification. Wiley-Interscience, 2000.
- [28] I. J. Cox, M. Miller, J. Bloom, J. Fridrich, and T. Kalker, *Digital watermarking and steganography*. Morgan Kaufmann, 2007.
- [29] M. Tran, "Thai police name suspected web paedophile," The Gardian, October 2007.
- [30] T. Pevný and J. Fridrich, "Novelty detection in blind steganalysis," in *Proceedings of the 10th ACM workshop on Multimedia and security*. ACM New York, NY, USA, 2008, pp. 167–176.
- [31] T. Fawcett, "ROC graphs: Notes and practical considerations for researchers," *Techni-cal Report HPL-2003-4*, 2003.
- [32] L. Breiman, Classification and Regression Trees. Chapman & Hall/CRC, 1998.
- [33] M. Goljan, J. Fridrich, and T. Soukal, "Higher-order statistical steganalysis of palette images," *Proceedings of SPIE*, vol. 5020, pp. 178–190, 2003.
- [34] O. Dabeer, K. Sullivan, U. Madhow, S. Chandrasekaran, and B. S. Manjunath, "Detection of hiding in the least significant bit," *IEEE Transactions on Signal Processing*, vol. 52, no. 10, pp. 3046–3058, October 2004.

- [35] A. D. Ker, "A general framework for structural analysis of LSB replacement," *Proceedings of the 7th Information Hiding Workshop, LNCS*, vol. 3727, pp. 296–311, June 2005.
- [36] J. Harmsen, "Steganalysis of additive noise modelable information hiding," Ph.D. dissertation, Ph.d. thesis at Rensselaer Polytechnic Institute, 2003.
- [37] G. Cancelli, G. Doerr, I. J. Cox, and M. Barni, "Detection of ±1 LSB steganography based on the amplitude of histogram local extrema," in *Image Processing*, 2008. *ICIP* 2008. 15th IEEE International Conference on, 2008, pp. 1288–1291.
- [38] U. S. D. of Agriculture, "Natural Resources Conservation Service photo gallery," 2002. [Online]. Available: http://photogallery.nrcs.usda.gov
- [39] C. Corporation, "Corel Stock Photo Library 3," Ontario, Canada.
- [40] G. Cancelli, G. Doërr, M. Barni, and I. J. Cox, "A comparative study of ±1 steganalyzers," *Multimedia Signal Processing*, 2008 IEEE 10th Workshop on, pp. 791–796, Oct. 2008.
- [41] G. Cancelli, G. Doërr, M. Barni, and I. J. Cox, "Comparing steganalyzers: A case study with ±1 steganography," *IEEE Transactions on Information Forensics and Security*, vol. submitted for publication, 2009.
- [42] F. Petitcolas, R. Anderson, and M. Kuhn, "Attacks on copyright marking systems," in Proceedings of the Second International Workshop on Information Hiding. Springer, 1998, pp. 218–238.
- [43] G. Doërr, "Image database for steganalysis studies." [Online]. Available: http://www.adastral.ucl.ac.uk/ gwendoer/steganalysis
- [44] J. Fridrich, "Feature-based steganalysis for JPEG images and its implications for future design of steganographic schemes," in *Information Hiding*, 6th International Workshop, vol. 3200. Springer, 2004, pp. 67–81.
- [45] T. Pevny, J. Fridrich, and A. Ker, "From blind to quantitative steganalysis," in *Proceed-ings of SPIE*, vol. 7254, 2009, p. 72540C.
- [46] C. Liu and S. Liao, "High-performance JPEG steganography using complementary embedding strategy," *Pattern Recognition*, 2008.
- [47] M. Kharrazi, H. Sencar, N. Memon *et al.*, "Benchmarking steganographic and steganalysis techniques," *EI SPIE San Jose, CA*, 2005.

- [48] D. Lou, C. Liu, and C. Lin, "Message estimation for universal steganalysis using multiclassification support vector machine," *Computer Standards & Interfaces*, vol. 31, no. 2, pp. 420–427, 2009.
- [49] T. Pevny and J. Fridrich, "Towards multi-class blind steganalyzer for JPEG images," *Lecture notes in computer science*, vol. 3710, p. 39, 2005.
- [50] T. Pevny and J. Fridrich, "Merging Markov and DCT features for multi-class JPEG steganalysis," *IS&T/SPIE EI*, vol. 6505, 2007.
- [51] T. Pevny and J. Fridrich, "Multiclass Detector of Current Steganographic Methods for JPEG Format," *IEEE Transactions on Information Forensics and Security*, vol. 3, no. 4, pp. 635–650, 2008.
- [52] Y. Miche, P. Bas, A. Lendasse, C. Jutten, and O. Simula, "Reliable Steganalysis Using a Minimum Set of Samples and Features," *EURASIP Journal on Information Security*, 2009.
- [53] C. Jutten and O. Simula, "Advantages of Using Feature Selection Techniques on Steganalysis Schemes," in *Computational and Ambient Intelligence 9th International Work-Conference on Artificial Neural Networks, IWANN 2007, San Sebastian, Spain, June 20-22, 2007: Proceedings.* Springer, 2007, p. 606.
- [54] Y. Miche, B. Roue, A. Lendasse, and P. Bas, "A feature selection methodology for steganalysis," *Lecture Notes in Computer Science*, vol. 4105, p. 49, 2006.
- [55] P. Vandergheynst and P. Frossard, "Image coding using redundant dictionaries," *Document And Image Compression*, p. 207, 2006.
- [56] R. i Ventura, P. Vandergheynst, and P. Frossard, "Low-rate and flexible image coding with redundant representations," *IEEE Transactions on Image Processing*, vol. 15, no. 3, 2006.
- [57] G. Cancelli, M. Barni, and G. Menegaz, "Mpsteg: hiding a message in the matching pursuit domain," in *Proceedings of SPIE*, vol. 6072, 2006, p. 60720P.
- [58] T. Hien, I. Kei, H. Harak, Y. Chen, Y. Nagata, and Z. Nakao, "Curvelet-Domain Image Watermarking Based on Edge-Embedding," *Lecture Notes in Computer Science*, vol. 4693, p. 311, 2007.
- [59] S. Lee, C. Yoo, and T. Kalker, "Reversible image watermarking based on integer-tointeger wavelet transform," *IEEE Transactions on Information Forensics and Security*, vol. 2, no. 3 Part 1, pp. 321–330, 2007.

- [60] X. Zhang and K. Li, "Comments on" An SVD-based watermarking scheme for protecting rightful Ownership," *IEEE Transactions on Multimedia*, vol. 7, no. 3, pp. 593–594, 2005.
- [61] S. Mallat and Z. Zhang, "Matching pursuit with time-frequency dictionaries," *IEEE Transactions on Signal Processing*, vol. 41, no. 12, pp. 3397–3415, 1993.
- [62] P. Jost, P. Vandergheynst, and P. Frossard, "Redundant image representations in security applications," *International Conference on Image Processing (ICIP), 2004*, vol. 4, 2004.
- [63] K. Solanki, A. Sarkar, and B. Manjunath, "YASS: yet another steganographic scheme that resists blind steganalysis," *Information Hiding: 9th International Workshop, IH* 2007, Saint Malo, France, June 11-13, vol. 4567, pp. 16–31, 2007.
- [64] Z. Fan and R. de Queiroz, "Identification of bitmap compression history: JPEG detection and quantizer estimation," *IEEE Transactions on Image Processing*, vol. 12, no. 2, pp. 230–235, 2003.