Steganography and Steganalysis of Raw and Compressed Image Data

BY

MEHDI SHARIFZADEH
B.Sc., Sharif University of Technology, 2012
M.Sc., University of Illinois at Chicago, 2018

THESIS

Submitted as partial fulfillment of the requirements
for the degree of Doctor of Philosophy in Electrical and Computer Engineering
in the Graduate College of the
University of Illinois at Chicago, 2019

Chicago, Illinois

Defense Committee:

Dan Schonfeld, Chair and Advisor
Rashid Ansari
Hulya Seferoglu
Mojtaba Soltanalian
Brian Ziebart, Computer Science

to Bahareh, Maryam, Zahra, and Mohsen

for their unconditional love and support

ACKNOWLEDGMENTS

First and foremost, I would like to express my heartfelt gratitude towards my advisor, Dr. Dan Schonfeld, for his continuous support, inspiration, encouragement, immense knowledge, insightful guidance, and his intellectual contribution to my development as a scientist.

I would like to express my appreciation to my dissertation committee, Dr. Rashid Ansari, Dr. Mojtaba Soltanalian, Dr. Hulya Seferoglu, and Dr. Brian Ziebart, for serving on my dissertation committee and for their guidance and inputs throughout this process.

I want to extend my thanks to my colleagues and fellow labmates Mohammed Aloraini, Chirag Agarwal, and Dr. Mehdi Salarian for constructive collaboration. I would also like to thank my friends for their continued support and suggestions: Dr. Armin Sarabi, Dr. Amirhassan Kermanshah, Dr. Pouria Naderi, Mehrtash Haghighat, Saman Farhat, Alireza Kardouni, Alireza Bahramirad, Dr. Soroush Khaleghi, Dr. Sepideh Roghanchi, Dr. Ashkan Mahdavi and Amir Katoozian.

Most of all, I want to thank my family, my beloved parents, Zahra and Mohsen, and my only sister, Dr. Maryam, for their endless love, and for inspiring me to follow my dreams. This journey would not be possible without you.

In the end, I would like to thank my wife, Dr. Bahareh, for the love that she keeps giving me, always believing in me and my capabilities and for giving me the courage to try. Thank you for coming into my life.

MS

CONTRIBUTION OF AUTHORS

The content of Chapter 3 was published in (Sharifzadeh et al., 2019a; Sharifzadeh et al., 2019b). The content of Chapter 4 is submitted as a journal paper to IEEE Transactions on Circuits and Systems for Video Technology. In all the mentioned publications, my adviser, Dr. Dan Schonfeld, was the lead investigator. I was the main contributor, responsible for coming up with the main ideas, implementations of all the experiments and simulations, and writing of the papers. Mohammed Aloraini contributed in developing the idea and editing the manuscript.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

CHAPTER **PAGE**

LIST OF TABLES

LIST OF TABLES (Continued)

TABLE **PAGE**

LIST OF FIGURES

SUMMARY

Steganography is the art of hiding data in a cover medium without arousing suspicion of the warden (Simmons, 1984). In this thesis, we focus on the most popular and studied cover medium for steganography, digital images. In digital image steganography, the statistical model of an image is essential for hiding data in less detectable regions and achieving better security. This has been addressed in the literature, where different cost-based and statistical model-based approaches were proposed. However, due to the usage of heuristically defined distortions or statistical models resulting in numerically solvable equations, there is no closed-form expression for security as a function of payload. The closed-form expression is crucial for a better insight into image steganography, batch steganography, and pool steganalysis problems. Besides, it is also required for improving the security of steganography and batch steganography algorithms against single image and pool steganalysis. Towards this goal, our research is focused on four problems.

1) We develop a general spatial image steganography embedding model that can utilize embedding costs and residual variances for embedding the hidden message and achieves state-of-the-art performance. 2) We extend the embedding model to JPEG steganography, which is also generalized in the sense that it can accomplish embedding using any spatial or DCT embedding cost as well as residual variances. Employing the proposed model improves the security of previous works and outperforms the state-of-the-art JPEG steganography algorithms. 3) We derive the closed-form expression for steganalysis error of batch steganography. The expression allows us to study the effect of batch size on security

which results in a novel batch steganography method, **Ada**ptive **B**atch size **I**mage **M**erging steganographer (*AdaBIM*). 4) We further extend the closed-form expression of single image steganalysis to pool steganalysis for an optimal omniscience detector. The developed analytical model is validated by its ability to accurately estimate empirical results of pool steganalysis and predict the behavior of empirical pool steganalysis error variance.

CHAPTER 1

INTRODUCTION

1.1 Motivation and background

Steganography is the art of concealing data within an innocent cover medium such as multimedia files. This problem was first defined as the prisoners' problem in which two prisoners, Alice and Bob, want to communicate a secret message in the presence of warden, Wendy (Simmons, 1984). Alice creates a stego medium by embedding the secret message in a cover medium using a private or public key, and Bob decodes the message from the stego medium. Steganography is closely related to watermarking and cryptography. However, they all have different methodologies and applications. In cryptography, information is converted from a readable state to incomprehensible state. Regardless of how complex and unbreakable a cryptography method is, it arises suspicion. In watermarking, a marker is embedded for authenticity purposes and tracing infringements. As a result, robustness is a top priority. In contrast to these two concepts, steganography hides the very existence of the hidden content, and not rising suspicion is the top priority. Furthermore, by applying encryption first, then steganography, the content of the hidden message is protected in addition to concealing its existence (Sadek et al., 2015).

Due to the high redundancy of digital images, they are the most common medium for steganography. Early methods of image steganography do not take into account the underlying distribution and correlation of image elements and, therefore, hide the same amount of hidden message in every element.

Examples of such non-adaptive methods for spatial domain are (Cheddad et al., 2010; Johnson and Jajodia, 1998) and for DCT domain are (Upham, 1993; Westfeld, 2001; Fridrich et al., 2007). Treating all the cover elements similarly results in embedding hidden message in highly correlated regions in which even small perturbations are easily detectable (Fridrich et al., 2001; Fridrich et al., 2007). To tackle this issue and increase security, one must do image steganography adaptively by considering the detectability of cover elements and embedding more hidden data in less detectable ones. This problem is modeled by an optimization problem for minimizing the distortion caused by embedding data and formulated to source coding with a fidelity criterion (Shannon, 1959). A general solution for this optimization problem is Syndrome Trellis Codes (STC), which performs embedding according to cover elements embedding costs (Filler and Fridrich, 2010; Filler et al., 2011). STC hides more data in low-cost elements and less in high-cost ones.

STC has led to many studies on computing embedding costs for image steganography for both spatial and JPEG domains. Well-known examples of such studies for spatial domain are as follows (Pevnỳ et al., 2010b; Holub and Fridrich, 2012; Fridrich and Kodovskỳ, 2013; Li et al., 2014; Holub et al., 2014; Sedighi et al., 2015; Sedighi et al., 2016). For JPEG domain steganography, they are (Guo et al., 2014; Holub et al., 2014; Guo et al., 2015; Pan et al., 2016; Denemark and Fridrich, 2017). These examples can be clustered into two categories according to their approach, one is cost-based methods that compute costs using purely heuristic approaches, and the other one is model-based method, which calculates costs based on statistical models for cover and stego mediums. Because of using heuristics or statistical models resulting in numerically solvable equations, there are no closed-form solutions for spatial and JPEG image steganography. Obtaining a closed-form solution would allow us to better

understand image steganography as well as batch steganography and pool steganalysis and enable us to predict their behavior.

Batch steganography is the extension of steganography in which the embedder spreads the hidden message among multiple cover objects. On the other hand, pool steganalysis is the extension of steganalysis problem in which the detector examines multiple objects sharing the same sender and pools all the available pieces of evidence. Batch steganography and pool steganalysis are dual problems introduced in (Ker, 2006) and highlighted as significant open problems in steganography in (Ker et al., 2013).

In the subject of batch steganography, early works were focused on non-adaptive message spreading techniques (Ker, 2007; Ker, 2008a; Ker and Pevny, 2012). Later on, in (Zhao et al., 2016), a content-adaptive batch steganography method has been introduced where the suitability of the images in the batch are taken into consideration for spreading payload. The results were further improved using better spreading strategy in (Cogranne et al., 2017). In all these methods, all the available covers were grouped into one batch, and the effect of smaller batch sizes has never been studied. Also, no closed-form expression has been derived for steganalysis of batch steganography.

For the pool steganalysis problem, the case of having multiple senders has been studied in (Ker and Pevnỳ, 2011; Ker and Pevny, 2012). Sequential steganalysis scenario has been discussed in (Cogranne, 2015). Later, researchers have focused on finding out if a single sender is guilty or not. Assuming that the detector is omniscience, it has been shown that the average pooling function is close to optimal in (Pevnỳ and Nikolaev, 2015). Cogranne et al. have shown that knowing the steganographer's strategy improves pool steganalysis results (Cogranne et al., 2017). However, Zakaria et al. proposed a pooling

method that performs close to an omniscience pool steganalysis without having the knowledge of the sender's strategy (Zakaria et al., 2019). In all the mentioned works, there is no statistical analysis for modeling pool steganalysis of steganography with the state-of-the-art payload spreading strategies in real images.

1.2 Main contributions

In this section, we provide a summary of various studies presented in this thesis. We do not intend to propose any method for computing cost of embedding in spatial or DCT domain, or estimating pixel residual variances, but rather to develop an embedding model which can be leveraged for spreading the hidden message. Note that the separation between the computation of embedding cost or residual variance and the embedding model suggests that the proposed methods can potentially be used for steganography in other cover mediums such as video and audio data, extending the applicability of our studies to other steganography scenarios.

1.2.1 Image Steganography for Raw Images

We develop a statistical framework for image steganography in spatial domain in which the cover and the stego messages are modeled as multivariate Gaussian random variables. We propose a novel quantized Gaussian embedding model by maximizing the detection error of the most common optimal detectors within the adopted statistical model. Afterward, the closed-form detection error is derived within the adopted model for spatial image steganography. Furthermore, we extend the formulation to cost-based steganography, resulting in a universal embedding scheme that improves the empirical results of current cost-based and statistical model-based approaches. This methodology and its presented

solution remain the same for any m-ary embedding scenario, because of assuming a continuous hidden message.

1.2.2 Image Steganography for Compressed Images

We extend the statistical framework developed for raw image steganography to compressed image steganography. Similarly, the cover and the hidden message are modeled by multivariate Gaussian distribution. Based on this statistical model, we propose a novel quantized Gaussian JPEG steganography model, which is able to accomplish embedding using any spatial or DCT embedding cost as well as residual variances. Employing the proposed model improves the security of previous works and outperforms the state-of-the-art JPEG steganography algorithms.

1.2.3 Batch Steganography

Batch steganography is the extensions of steganography where the hidden message is spread in multiple objects. To address this problem, we extend the closed-form detection error derived within our statistical model for image steganography to batch steganography. Using the closed-form expression, we introduce a novel batch steganography method, **Ada**ptive **B**atch size **I**mage **M**erging steganographer (*AdaBIM*), and mathematically prove it outperforms the state-of-the-art batch steganography method and further verify its superiority by experiments.

1.2.4 Pool Steganalysis

The pool steganalysis problem arises when the detector knows a pool of objects share the same source, and therefore, it jointly analyzes the objects. To tackle the pool steganalysis problem, we extend the closed-form expression of single image steganalysis detection error to pool steganalysis for an omniscience optimal warden. We employ the derived expression to approximate the empirical results of pool

steganalysis computed by an ensemble classifier steganalyzer. The approximation is based on the empirical detection error of single image steganalysis, and it approximates the detection error for any pool size greater than one. Although the approximation is derived based on the adopted statistical model, it is precise for all the payloads, embedding domains, embedding methods, and steganalysis features as long as the pooling strategy is optimal. This validates our analytical model. In addition to approximation of the error, we employ the proposed model to make predictions about the behavior of pool steganalysis error variance. Our model shows that the variance increases as the pool size increases in small payloads. We observe the same behavior in empirical results, which re-validates our analytical model. Small payloads are more useful comparing to high payloads, which are easily detectable. Therefore, we conclude that although pooling makes the detector more reliable as it decreases the detection error, it makes the detector less reliable in the sense that it increases the variance. In other words, pooling makes the steganalyzer less stable.

1.3 Organization

The rest of this thesis is organized as follows. A detailed summary of previous works in steganography and steganalysis is presented in Chapter 2. In Chapter 3, we elaborate on our published works (Sharifzadeh et al., 2019a; Sharifzadeh et al., 2019b) for developing an embedding model for image steganography in spatial domain and leveraging the model for batch steganography. In Chapter 4, we discuss the extension of the model to DCT domain resulting in an embedding model for JPEG steganography and a unified framework for estimation of pool steganalysis error and its variance. Chapter 5 concludes this thesis and discusses possible future directions.

CHAPTER 2

OVERVIEW OF STEGANOGRAPHY AND STEGANALYSIS

2.1 Steganography

Since the definition of steganography problem as the prisoners' problem, there have been lots of theoretical and empirical studies in this topic (Simmons, 1984). One of the earliest theoretical studies is Cachin's work from an information-theoretic point of view, where he proposed a model for steganography (Cachin, 1998). He interpreted the problem of detecting the existence of a secret message in a medium as a hypothesis testing problem. Then, he defined perfectly secure and ε-secure steganography using relative entropy and KullbackLeibler divergence between the cover and stego messages. This problem was investigated further in a study by Moulin et al., where an upper-bound was derived for steganography by incorporating the trade-off between the achievable information rate and the allowed distortion levels for the steganographer and the steganalyzer (Moulin and O'Sullivan, 2003). They provided explicit formulas for hiding capacity in various cases based on different assumptions regarding the probability distribution of the cover medium and the availability of side information. In a later study by Moulin et al., considering a unified framework for data hiding problem, capacity formulas and random-coding exponents were derived for coding part of such problems resulting in asymptotic upper bounds on the achievable probability of channel coding error (Moulin and Wang, 2007). In another study, a coding method was proposed for perfectly secure steganography, which achieves the transmission rate upper bound derived in the literature while guaranteeing not to alter the underlying distribution of cover

(Wang and Moulin, 2008). Similarly, in (Ryabko and Ryabko, 2009), a perfectly secure cover generating steganography system was proposed, which asymptotically reaches the transmission rate upper bound. All these theoretical works assumed that the steganographer has a perfect knowledge of the cover probability distribution, and they concluded that the transmission rate is proportional to the cover size. However, in practice, steganographers can only approximate the cover distribution, and this results in hidden message size being a sub-linear function of cover size. This has been explored in many studies under various conditions, and they all concluded in a sub-linear relation, and some of them concluded in the well-known square root law of steganography (Anderson, 1996; Ker, 2004; Böhme, 2005; Ker, 2006; Ker, 2007; Ker, 2008b; Ker et al., 2008; Filler et al., 2009). The square root law has different variants under different conditions and they all state that the steganographic capacity of a cover is proportional to the square root of cover size.

Now, we go through a summary of previous works in practical steganography for hiding data in images, which is the focus of this thesis. Image steganography methods can be divided into two groups, i.e., spatial and frequency domain. Spatial image steganography methods alter the intensity of pixels of an image; however, frequency-domain methods embed in a transform domain, e.g., by changing the coefficients of Discrete Cosine Transform (DCT) of a JPEG image. In this thesis, we focus on spatial and JPEG steganography. In both types of image steganography, non-adaptive methods have low security because of treating all the cover elements similarly and not taking into account their correlation. After the development of STC, lots of studies have been done on adaptive steganography, which tries to model the image and assign a suitability measure to each cover element and then embed the hidden message

accordingly. In the rest of this section, we explain recently proposed methods for spatial and JPEG steganography.

The state-of-the-art spatial image steganography algorithms fall into two main categories: cost-based and statistical or model-based methods. On the one hand, cost-based algorithms heuristically define the cost of embedding in each pixel. These costs should be lower in noisy or textured regions where changes due to embedding hidden message are less detectable. However, they should be higher in smooth regions where even small perturbations are easily detectable due to cover elements being highly correlated. To achieve this goal, Wavelet Obtained Weights (WOW) algorithm (Holub and Fridrich, 2012) utilizes a bank of directional high-pass filters to find suitable areas for embedding in which the image has high frequency of change in every direction. Similarly, the S-UNIWARD algorithm assigns a cost to each cover pixel by calculating the summation of changes in directional filter bank decomposition coefficients of the cover caused by changing that pixel (Holub et al., 2014). As a result, in smooth regions where pixels are easy to predict in every direction, embedding costs are high and less hidden message is embedded. However, in noisy regions that are hard to model, more embedding takes place. In another work, Li et al. have proposed a method called HILL for calculating costs, which is faster and more secure comparing to S-UNIWARD (Li et al., 2014). HILL finds noisy regions of a cover image using a high-pass filter, and then, smooths the estimated costs by two low-pass filters.

On the other hand, statistical or model-based spatial steganography methods analytically model the cover instead of using pure heuristics for computing embedding costs, and then, they hide more data in regions that are noisier according to the model. The first statistical model-based steganography is HUGO (Pevnỳ et al., 2010b), which defines distortion as a weighted sum of differences between SPAM feature

vectors of a cover image and its stego version (Pevny et al., 2010a). The main downside of embedding while preserving an empirical feature space such as SPAM is that since the method is over-trained to that feature, it results in low security if the warden utilizes a more complete feature space (Kodovsky et al., 2011). Another approach is proposed in (Fridrich and Kodovskỳ, 2013), which models the cover image pixels by independent normally distributed variables, where the variances are computed using a proposed variance estimator. Then, the message is embedded in a ternary scheme in each pixel while minimizing the KL divergence between the cover and the stego message. Using a similar framework but with a generalized Gaussian statistical model for cover images, a better variance estimator, embedding quinary message in each pixel, and minimizing the detection error of an optimal hypothesis testing detector, better results were achieved in (Sedighi et al., 2015). Building upon the result of these two works, MiPOD (Sedighi et al., 2016) was proposed outperforming state-of-the-art cost-based image steganography methods. In MiPOD, the cover is modeled as independent Gaussian random variables, and the stego message is the result of embedding a ternary message in each cover pixel. The embedding is done in a way to minimize the power of a hypothesis testing detector. In contrast to the methods utilizing the KL divergence, this method does not require the assumption of small payload.

Similar to spatial image steganography, steganography methods in JPEG domain include two main categories: cost-based and statistical or model-based approaches. A summary of the state-of-the-art cost-based JPEG steganography methods is as follows. Similar to S-UNIWARD, J-UNIWARD uses directional filter banks to calculates costs, but it computes the cost of changing the scaled DCT coefficient instead of changing a pixel in spatial domain (Holub et al., 2014). Then, in contrast to the conventional JPEG steganography approaches, which embed only in non-zero AC DCT coefficients, it alters all the

coefficients according to the calculated costs. J-UNIWARD suffers from high computational complexity because of using wavelet domain for determining costs. In another work, Guo et al. proposed a faster approach by assigning costs to DCT coefficients in a way that results in embedding changes being uniformly spread among different coefficients magnitudes (Guo et al., 2014). As a result, the average changes of the first and the second-order statistics of DCT coefficients is reduced, and UED achieves acceptable performance. Later, based on UED, UERD was proposed, which improved the results of UED algorithm by taking into account the energy of a block of DCT coefficients and its neighbors for determining the cost of embedding in its coefficients (Guo et al., 2015). UERD embeds more in blocks with higher energy as these blocks belong to noisier regions of image, and therefore, it achieves better security comparing to UED. In blocks located between noisy and smooth regions of images, UERD does not work well, because the high energy of a noisy region results in lower embedding costs not only for its own coefficients but also for coefficients in its neighboring blocks. This issue has been addressed in a later study where a new distortion function called IUERD was proposed. By incorporating the correlation of neighboring DCT blocks more efficiently, IUERD achieves considerably better security compared to its predecessors, and its performance is comparable to J-UNIWARD, which is computationally more expensive. In JPEG steganography, there has been only one statistical-based method called J-MiPOD, which is based on MiPOD, and it employs the same statistical cover model as MiPOD (Denemark and Fridrich, 2017). In the mentioned study, in addition to proposing J-MiPOD, they have also introduced algorithms for steganography with side information or pre-cover for both spatial and JPEG domains (SI-MiPOD and SI-J-MiPOD).

The extension of image steganography in a single cover is batch steganography in which the embedder spreads the hidden message among multiple cover objects. Batch steganography for non-adaptive message spreading techniques was studied in (Ker, 2007; Ker, 2008a; Ker and Pevny, 2012), showing that the message should be distributed evenly or concentrated in the fewest possible number of cover mediums depending on the payload. However, for content-adaptive methods, Zhao et al. showed that choosing a more suitable sub-batch of images to carry the whole message significantly improves security comparing to randomly choosing a sub-batch (Zhao et al., 2016). Further studies improved the performance by spreading the payload among all the images of a batch using three message spreading techniques, distortion limited sender (DiLS), detectability limited sender (DeLS), and image merging sender (IMS) (Cogranne et al., 2017). Assuming only one batch containing all the images of a dataset, DiLS and DeLS spread the payload among them in a way to have the same distortion and KL divergence, respectively, according to an adopted cover model. However, IMS, the best performing technique, merges all the images, then the embedding algorithm distributes the payload among them. In other words, IMS treats all the pixels in a batch as though they belonged to one image.

2.2 Steganalysis

The converse problem of steganography is steganalysis, in which a detector tries to distinguish between a cover object and a stego one. Steganalysis can be used as a performance measure for designing steganography algorithms, and for comparing different steganography methods. Theoretical approaches for steganalysis include, but is not limited to, KullbackLeibler divergence and hypothesis testing using likelihood ratio. These theoretical methods require the exact knowledge of the underlying distribution of the cover and the stego objects. However, such knowledge is not accessible for real-world cover objects

such as images. Therefore, image steganalysis is done empirically using machine learning techniques by training a classifier on steganalysis features extracted from a database of images, including both cover and stego objects. This approach has a few disadvantages, which are subjects of many studies in the field of steganalysis research. Theoretical performance analysis is impossible due to the usage of heuristically defined features that are needed for training classifiers. Another problem called cover source mismatch arises because of the differences between the database and cover source. And last but not least, these steganalysis methods are time-consuming, and they become more computationally expensive as the feature dimension increases.

Now, we go through a summary of steganalysis features proposed for detecting spatial and JPEG steganography in the literature. There is a group of steganalysis features called rich models that are built by concatenating a large number of sub-models where each sub-model computes noise residuals using different denoising filters. In all the denoising filters, the prediction of a cover element is made using only neighbors of the element, not the element itself. Therefore, in analyzing a stego message, the embedding only affects the residual of the prediction, not the prediction, given that the hidden message elements are independent. In steganalysis using rich models, the classifier is responsible for understanding the dependencies between all the residuals calculated by sub-models. The first rich model for spatial steganalysis was introduced in (Pevny et al., 2010a), which uses first-order and second-order Markov chains to model the differences between adjacent pixels. Although this method, SPAM, was designed to detect spatial steganography, it detected JPEG steganography as well. SPAM was later extended to two of the well-known rich models for spatial steganalysis, i.e., SRM (Fridrich and Kodovsky, 2012) and PSRM (Holub and Fridrich, 2013), by adding more sub-models. They both use the same 45 lin-

ear and non-linear pixel estimators to predict each pixel and approximate the noise component of that pixel by subtracting its value from its estimation. However, after approximation of noise residuals, they utilize different statistical representation of noise residuals. SRM features are computed based on four-dimensional co-occurrences, but PSRM is based on histograms of residual projections on to multiple random directions. PSRM features outperform SRM features, but they are computationally more expensive. Further improvement was done in (Denemark et al., 2014) where maxSRM is proposed. This steganalysis feature set makes use of selection channel and utilizes an approximation of the probability of embedding changes. Throughout this thesis, we use maxSRMd2, which is a variation of maxSRM, and it is proposed in the same paper for steganalysis of spatial steganography (Denemark et al., 2014).

Rich models are also extended to steganalysis for JPEG steganography in (Kodovskỳ and Fridrich, 2012), where JRM features are proposed based on prediction of DCT coefficients from their frequency and spatial neighborhoods. JRM is effective for detecting non-adaptive JPEG steganography methods; however, it is far less successful in detection of content-adaptive methods. Steganalysis feature sets that are successful in detection of adaptive JPEG steganography methods are called phase aware features as they split the histogram of the noise residuals by their JPEG phase, i.e., location of the DCT coefficient in the 8×8 block (Holub and Fridrich, 2014; Song et al., 2015; Holub and Fridrich, 2015). The effectiveness of splitting by JPEG phase comes from the fact that the impact of altering DCT coefficients on pixels in a decompressed JPEG image depends on the JPEG phase. All three mentioned methods calculate the histogram of the residual, but they utilize different filter banks. DCTR determines residuals using a filter bank of 64 kernels corresponding to 64 JPEG phases, while GFR uses 256 Gabor filters,

and PHARM uses 900 kernels. For steganalysis of JPEG steganography, we use DCTR and GFR feature set as they achieve state-of-the-art performance.

All the performance measures reported in this paper are computed using an ensemble of random forest classifiers optimized for high dimensional features (Kodovsky et al., 2012). This classifier is trained on the features extracted from a database of images containing cover and stego images. Then, the classifier is tested on a testing set, that has no overlap with the training set, and the average probability of detection error, defined as the average of false alarm and missed detection based on equal priors, is reported as a performance measure. For a fair comparison of different steganography approaches, we employ the standard BOSSbase 1.01 database with $10k$ gray-scale 512×512 pixels images (Bas et al., 2011).

The extension of single image steganalysis is pool steganalysis, where the detector knows that a pool of object share the same source and, therefore, examines them together. The multiple sender scenario has been discussed, and various methods were proposed to rank the senders according to their possibility of being a stego message sender (Ker and Pevnỳ, 2011; Ker and Pevny, 2012). Further work has been done to answer a more general question of whether a source is guilty or not with a different assumption about the knowledge of the detector and the payload spreading strategy. In the case of the detector being omniscience, the average pooling function is shown to be close to optimal for various payload spreading strategies in (Pevnỳ and Nikolaev, 2015). In a later study, sequential steganalysis, a variation of pool steganalysis problem, is discussed (Cogranne, 2015). Cogranne et al. have shown that knowledge of the payload spreading strategy improves pool steganalysis (Cogranne et al., 2017). However, Zakaria et al.

proposed a pooling method that performs close to an omniscience pool steganalysis without having the knowledge of payload spreading strategy (Zakaria et al., 2019).

CHAPTER 3

ADAPTIVE BATCH SIZE IMAGE MERGING STEGANOGRAPHY AND QUANTIZED GAUSSIAN IMAGE STEGANOGRAPHY

3.1 Introduction

Steganography problem is formulated by the prisoner's problem where Alice and Bob want to communicate through a cover medium without raising any suspicion from Wendy, the warden (Simmons, 1984). In this paper, we focus on the most popular and studied cover medium, digital images. Non-adaptive image steganography approaches (Cheddad et al., 2010; Johnson and Jajodia, 1998) are easily detectable as they neglect pixel to pixel dependencies (Fridrich et al., 2001). Therefore, in order to achieve a better security, hidden message should be embedded in textured or noisy areas rather than smooth regions. This has led to a group of content-adaptive spatial image steganography methods, which we call cost-based methods. In these methods, message embedding is done while minimizing the caused distortion, and it is formulated as a source coding problem with a fidelity criterion (Shannon, 1959). These methods include two main steps, first is calculating the cost of embedding in each pixel using a heuristically defined distortion function, and second is embedding the message according to the costs. The second step is solved for a general distortion function using syndrome trellis codes and Gibbs measure (Filler et al., 2011; Filler and Fridrich, 2010). Examples of such steganography methods are **S**patial **UNI**versal **WA**velet **R**elative **D**istortion (SUNIWARD) (Holub et al., 2014) and

HIgh-pass, **L**ow-pass, and **L**ow-pass (HILL) (Li et al., 2014). Although these methods achieve superior results, there is no theoretical relation between statistical security measures and these derived distortion functions (Böhme, 2010). Thus, the security of these methods can be measured only empirically. This issue has been resolved in the other category of steganography methods, which we call statistical-based methods. They rely on a cover model and aim to minimize statistical distortion while embedding.

The first successful example of such a steganography method is **H**ighly **U**ndetectable ste**GO** (HUGO), which tries to preserve SPAM feature vector (Pevny et al., 2010a) of the cover while embedding (Pevnỳ et al., 2010b). HUGO has low security against steganalysis with more complete feature space since it is overfit to SPAM features (Kodovsky et al., 2011). To avoid this drawback, embedding can be done while minimizing statistical detectability instead of preserving an empirical feature space. This has been addressed in a revolutionary work by Fridrich et al., where a general Gaussian model was developed for the cover image, and embedding was done by minimizing its statistical distortion modeled as Kullback-Leibler (KL) divergence (Fridrich and Kodovskỳ, 2013). The results were improved using a generalized Gaussian model and measuring statistical distortion as the performance of a likelihood ratio testing detector (Sedighi et al., 2015). By assuming Gaussian cover model and utilizing a better pixel variance estimator, security of (Sedighi et al., 2015) was enhanced in **Mi**nimizing the **P**ower of **O**ptimal **D**etector (MiPOD) (Sedighi et al., 2016).

In all the mentioned statistical model-based methods, as a result of a non-constrained message probability distribution, embedding probabilities are calculated using numerically solvable equations. Therefore, their performances are not expressed as closed-form functions of payload. Having a closed-form

detection error plays a critical role in understanding image steganography and also batch steganography in which the payload is spread across multiple objects.

Batch steganography and pool steganalysis are the extensions of steganography and steganalysis where the message is spread in multiple objects, and the detector jointly analyzes objects. These two concepts were introduced in (Ker, 2006) and highlighted as important open problems in (Ker et al., 2013). Non-adaptive message spreading batch steganography was studied in (Ker, 2007; Ker, 2008a; Ker and Pevny, 2012). Batch steganography for content-adaptive methods was introduced in (Zhao et al., 2016), where a more suitable sub-bath of images is chosen for embedding. The results were further improved by spreading the payload among all the images of a batch in (Cogranne et al., 2017). In all the proposed methods, the batch size is assumed to be infinity. In other words, the whole dataset is grouped into one batch. To the best knowledge of the authors, smaller batch sizes have never been studied in the literature.

In this chapter, our contribution is threefold:

1. For the first time, we model the hidden message as continuous Gaussian random variables and propose a novel Gaussian embedding technique by minimizing the detection error of the three most common optimal hypothesis detectors simultaneously. Subsequently, the closed-form detection error as a function of payload is derived for such an embedder. The explained formulation is also extended to the distortion minimization framework. As a result, the proposed embedding model can be applied not only utilizing residual variances estimated by any variance estimator used in model-based approaches such as MiPOD (Sedighi et al., 2016) and (Sedighi et al., 2015) but also using embedding costs calculated by any cost-based image steganography methods, such

as HILL (Li et al., 2014) and SUNIWARD (Holub et al., 2014). In all the cases, the proposed method results in better security compared to the original embedding schemes.

2. Employing continuous hidden message in the formulation allows us to do $(2q+1)$-ary embedding for any q by only changing the quantization levels. Therefore, we effortlessly investigate the effect of maximum pixel change (q) on the security of image steganography within the adopted model. We conclude that the higher the q is, the better the security is, which is contrary to the common belief of executing small changes or altering only the least significant bit of pixels.

3. We obtain the closed-form detection error for image merging batch steganography with batch size M. Consequently, we prove that using larger batch size results in higher detection errors in small payloads. However, for large enough payloads, using smaller batch sizes is more secure. Based on this, we introduce a novel **Ada**ptive **B**atch size **I**mage **M**erging steganographer (*AdaBIM*) that merges images in batches with size M, where M depends on the payload. It outperforms the state-of-the-art batch steganography method based on empirical evaluations.

This chapter is organized as follows. The statistical model for the cover and stego images are presented in Sec. 3.2. Using the statistical model, a framework is developed for a hypothesis testing detector in Sec. 3.3.1. Three optimal decision strategies for such a detector are investigated in Sec. 3.3.2. Based on all these strategies, a novel Gaussian embedding model is proposed in Sec. 3.4.1. In Sec. 3.4.2, the impact of batch size is studied, and a new batching strategy, *AdaBIM*, is proposed and proven to be superior. Then, the results are further extended to the distortion steganography framework in Sec. 3.4.3, which makes the Gaussian embedding model applicable to cost-based methods. The experimental results are provided in Sec. 3.5.

3.2 Statistical Models

In this section, the statistical models for the cover and the stego messages are described. Cover image pixels are modeled by independent Gaussian random variables. Subsequently, the distribution of the stego image pixels is derived by embedding a Gaussian message in each cover pixel.

The motivation of using a continuous random variable to model the discrete message arises because of the difficulty in solving this problem in the discrete space. We, therefore, propose to work in a continuous framework in which both the cover and the message are modeled by continuous random variables. Once the problem is solved in the continuous space, we discretize the derived solution to the original discrete model to obtain the desired results. We note that the discrete model could potentially be solved directly to provide the same (and possibly even superior) results. However, a direct closed-form solution for the discrete model is currently unknown and remains an open problem. Furthermore, we would like to have a unified probability framework where the cover and message distributions are consistent and remain unchanged once the message has been added to the cover in a spatial steganography scenario; i.e., we are limited to stable distributions, also known as Levy alpha-stable distributions, that are closed under linear transformation. Our interest is further focused on a random variable model among the stable distributions that is symmetric. It is known that a symmetric alpha-stable distribution can be viewed as a transform of zero-mean Gaussian random variables whose variance is drawn from a stable distribution (see, e.g., Section 3.2.2. in (Lee, 2010)). We, therefore, assume a Gaussian cover model as well as a Gaussian message model, which as a result of the central limit theorem, has the added advantage of robustness to channel and noise degradation as well as hostile attacks (see, e.g., Section 1.2.1 in (Lee, 2010)).

3.2.1 Cover Model

Cover images are shown by $\mathbf{c} = [c_1, \ldots, c_n] \in \mathscr{P} = \{0, \ldots, 255\}^n$, where $\mathscr{P}$ is the set of all vector representation of 8-bit gray-scale images of size $n_1 \times n_2 = n$. Each c_i is modeled as an independent Gaussian variable, $\mathscr{N}(\mu_i, \omega_i^2)$, quantized to $\mathscr{P}$. Suppose $\hat{\mu}_i$ is an unbiased estimation of μ_i based on the rest of the image. Thus, the residual of the estimation, defined as $x_i = c_i - \hat{\mu}_i$, has a Gaussian distribution, $\mathscr{N}(0, \sigma_i^2)$, where σ_i^2 is greater than ω_i^2 as it includes both the pixel's variance (ω_i^2) and the estimation error. Assuming $\sigma_i \gg \Delta$, where Δ is the quantization step size equal to 1, the probability distribution of the i^{th} cover pixel residual is

$$p_{x_i}(k) \propto \frac{1}{\sigma_i \sqrt{2\pi}} \exp\left(\frac{-k^2}{2\sigma_i^2}\right) \tag{3.1}$$

Refer to (Sedighi et al., 2016) for more information regarding this model. This statistical model is violated in practice in smooth or saturated regions because of assuming unbounded pixels and $\sigma_i \gg \Delta$. However, our proposed method will avoid embedding in those regions anyway which is covered thoroughly in Sec. 3.4.1.

3.2.2 Stego Model

Unlike the previous works which only considered discrete hidden message elements, we model them, m_i, as Gaussian random variables with variance β_i distributed according to

$$p_{m_i}(k) = \frac{1}{\beta_i \sqrt{2\pi}} \exp\left(\frac{-k^2}{2\beta_i^2}\right) \tag{3.2}$$

The stego image is the summation of the cover image with the stego message elements, i.e. $\mathbf{s} = \mathbf{c} + \mathbf{m}$. Hence, the i^{th} stego pixel residual is $y_i = x_i + m_i$, and based on (3.1) and (3.2), its probability distribution is derived as

$$p_{y_i}(k) \propto \frac{1}{\sqrt{2\pi(\sigma_i^2 + \beta_i^2)}} \exp\left(\frac{-k^2}{2(\sigma_i^2 + \beta_i^2)}\right) \tag{3.3}$$

with the assumption of unbounded quantization levels and $\sqrt{\sigma_i^2 + \beta_i^2} \gg \Delta$. The next section is devoted to find the proper β_is for achieving the best security for a payload limited sender.

3.3 Hypothesis Testing

The problem of steganography in a single image with a fixed payload can be formulated as constraint maximization of detection error of the warden (Sedighi et al., 2015; Sedighi et al., 2016) given by

$$\begin{cases} \underset{(\beta_1,\ldots,\beta_n)}{\arg\max}\, \mathrm{P_E}(\beta_1,\ldots,\beta_n) \\ \sum_{i=1}^{n} \mathrm{H}(p_{m_i}) = np \end{cases} \tag{3.4}$$

where $\mathrm{P_E}$ is the detection error derived in the following section, $\mathrm{H}(p_{m_i})$ is the entropy of a random variable with probability distribution p_{m_i} in natural unit of information (nats) and p is the relative payload in nats per pixel.

3.3.1 Likelihood Ratio Test Framework

To derive the detection error of the steganalyzer which is a function of the message variances, i.e. $\mathrm{P_E}(\beta_1,\ldots,\beta_n)$, we assume that the steganalyzer utilizes a likelihood ratio test (LRT) to do a binary

hypothesis testing between $\mathscr{H}_0$ and $\mathscr{H}_1$, representing the cases of receiving a cover or a stego image respectively. We assume the worst-case scenario of an omniscience steganalyzer who knows all the β_is and σ_is. Let us assume that $\mathbf{r} = [r_1, \ldots, r_n]$ are the residuals of the received image's pixels, and they are statistically independent. As a consequence, the likelihood ratio for the whole image can be written as $\prod_{i=1}^{n} \Lambda_i$ in which Λ_i, the likelihood ratio for the i^{th} pixel, can be written based on (3.1) and (3.3) as follows

$$\Lambda_i = \frac{p_{y_i}(r_i)}{p_{x_i}(r_i)} = \sqrt{\frac{\sigma_i^2}{\sigma_i^2 + \beta_i^2}} \exp\left(\frac{-r_i^2}{2} \frac{-\beta_i^2}{\sigma_i^2(\sigma_i^2 + \beta_i^2)}\right) \tag{3.5}$$

As a result, the natural logarithm of the likelihood ratio is

$$\ln \Lambda_i = \ln \sqrt{\frac{\sigma_i^2}{\sigma_i^2 + \beta_i^2}} + \frac{\beta_i^2}{2\sigma_i^2(\sigma_i^2 + \beta_i^2)} r_i^2 \tag{3.6}$$

where r_i has a normal distribution. Hence, r_i^2 multiplied by a constant term results in a Gamma distribution. Therefore, the natural logarithm of the likelihood ratio, $\ln \Lambda_i$, is a constant term plus a random variable with $\Gamma(k_i, \theta_i)$ distribution, where k_i and θ_i are the shape and scale parameters respectively. Parameter θ_i depends on the variance of r_i, in other words, whether r_i is distributed according to (3.1) or (3.3). In order to derive k_i and θ_i for both hypotheses, we employ Taylor series expansion of $\ln(1+x)$ where $x = \beta_i^2/\sigma_i^2$, assuming $x < 1$

$$\ln\left(\frac{\sigma_i^2}{\sigma_i^2 + \beta_i^2}\right) = -\ln\left(1 + \frac{\beta_i^2}{\sigma_i^2}\right) \approx -\frac{\beta_i^2}{\sigma_i^2} + \frac{1}{2}\left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \tag{3.7}$$

If $x = -\beta_i^2/(\sigma_i^2+\beta_i^2)$, the approximation is

$$\ln\left(\frac{\sigma_i^2}{\sigma_i^2+\beta_i^2}\right) \approx -\frac{\beta_i^2}{\sigma_i^2+\beta_i^2} - \frac{1}{2}\left(\frac{\beta_i^2}{\sigma_i^2+\beta_i^2}\right)^2 \tag{3.8}$$

which can be further simplified using Taylor series of $\frac{x}{1+x}$

$$\frac{\beta_i^2}{\sigma_i^2+\beta_i^2} \approx \frac{\beta_i^2}{\sigma_i^2} \tag{3.9}$$

Given $\mathscr{H}_0$, the Gamma distribution parameters are $k = 0.5$ and $\theta_i = \beta_i^2/(\sigma_i^2+\beta_i^2)$. The resulted mean and variance of the natural logarithm of the likelihood ratio are:

$$\begin{cases} \underset{r_i|\sigma_i,\beta_i}{\mathrm{E}}^{\mathscr{H}_0}[\ln\Lambda_i] = \ln\left(\sqrt{\frac{\sigma_i^2}{\sigma_i^2+\beta_i^2}}\right) + k\theta_i \approx \frac{-1}{4}\left(\frac{\beta_i^2}{\sigma_i^2+\beta_i^2}\right)^2 \approx \frac{-1}{4}\left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \\ \underset{r_i|\sigma_i,\beta_i}{\mathrm{Var}}^{\mathscr{H}_0}[\ln\Lambda_i] = k\theta_i^2 \approx \frac{1}{2}\left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \end{cases} \tag{3.10}$$

where the approximations are based on (3.8) and (3.9). However, for $\mathscr{H}_1$, $k = 0.5$, $\theta_i = \beta_i^2/\sigma_i^2$ and the mean and variance are

$$\begin{cases} \underset{r_i|\sigma_i,\beta_i}{\mathrm{E}}^{\mathscr{H}_1}[\ln\Lambda_i] = \ln\left(\sqrt{\frac{\sigma_i^2}{\sigma_i^2+\beta_i^2}}\right) + k\theta_i \approx \frac{1}{4}\left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \\ \underset{r_i|\sigma_i,\beta_i}{\mathrm{Var}}^{\mathscr{H}_1}[\ln\Lambda_i] = k\theta_i^2 = \frac{1}{2}\left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \end{cases} \tag{3.11}$$

where the approximation is based on (3.7). For large enough number of pixels (n), the following theorem can be used to approximate the probability distribution of $\sum_{i=1}^{n}\ln(\Lambda_i)$.

Theorem 1. *Asymptotic Sum of Gamma Random Variables*

Suppose $X_1, \cdots, X_n$ are all independently distributed by Gamma with shape k, but with scaling parameters $\theta_1, \cdots \theta_n$ respectively. If all θ's are bounded, the probability distribution of the following summation, where $a_1, \ldots, a_n$ are some constants, converges to normal distribution as shown below.

$$\sum_{i=1}^{n}(X_i + a_i) \xrightarrow{d} \mathcal{N}\Big(\sum_{i=1}^{n}(k\theta_i + a_i), k\sum_{i=1}^{n}\theta_i^2\Big) \tag{3.12}$$

See Appendix A: Asymptotic Sum of Gamma Random Variables for the proof. Thus, the probability distribution of $\sum_{i=1}^{n}\ln(\Lambda_i)$, for large enough n, can be approximated with the following distributions, based on (3.10) and (3.11):

$$\begin{cases} \mathcal{N}(\frac{-1}{4}\alpha,\ \frac{1}{2}\alpha) & \text{if } \mathscr{H}_0 \text{ is true} \\ \mathcal{N}(\frac{+1}{4}\alpha,\ \frac{1}{2}\alpha) & \text{if } \mathscr{H}_1 \text{ is true} \end{cases} \tag{3.13}$$

where α is as follows

$$\alpha = \sum_{i=1}^{n}\left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \tag{3.14}$$

The result shown in (3.13), is also consistent with the shift hypothesis, which states embedding only affects the mean of the detector's output (Ker, 2006). Here is the logarithm of the LRT

$$\sum_{i=1}^{n}\ln(\Lambda_i) \underset{\mathscr{H}_0}{\overset{\mathscr{H}_1}{\gtrless}} \gamma \tag{3.15}$$

where γ is the decision threshold. In the next section, we will discuss three different optimal decision criteria for deriving the decision boundary, γ, and consequently the detection error of the warden $\mathrm{P_E}(\beta_1,\ldots,\beta_n)$.

3.3.2 Optimal Decision Strategies

To derive the detection error of steganalyzer, $\mathrm{P_E}$, we employ the most common optimality criteria for hypothesis testing, Bayes, minimax, and Neyman-Pearson. All these strategies utilize a likelihood ratio test (LRT), but with different decision boundaries. In this section, we show that they all result in the same simplification of $\mathrm{P_E}(\beta_1,\ldots,\beta_n)$.

3.3.2.1 Bayes Criterion

Let's denote the prior probabilities of $\mathscr{H}_0$, and $\mathscr{H}_1$ with P_0, and P_1 respectively. The event and the cost associated with the decision $\mathscr{H}_i$ given that the true hypothesis is $\mathscr{H}_j$ are shown with D_{ij} and C_{ij} respectively. The risk function is defined as

$$R = \sum_{j=0}^{1}\sum_{j=0}^{1} P_i C_{ji}\; p(D_{ji}) \tag{3.16}$$

The Bayes decision boundary, γ_{Bayes}, which minimizes the risk defined in (3.16), is given by

$$\gamma_{\text{Bayes}} = \ln\left(\frac{P_0(C_{10}-C_{00})}{P_1(C_{01}-C_{11})}\right) \tag{3.17}$$

Consequently, the expected value of the detection error (summation of false alarm and missed detection) is given by

$$\mathrm{P}_{\mathrm{E}_{\mathrm{Bayes}}} = \phi(\frac{\gamma - \frac{\alpha}{4}}{\sqrt{\frac{\alpha}{2}}})P_1 + \phi(\frac{-\gamma - \frac{\alpha}{4}}{\sqrt{\frac{\alpha}{2}}})P_0 \tag{3.18}$$

where ϕ is the cumulative distribution of standard normal distribution, i.e. $\phi(x) = (2\pi)^{-0.5}\int_{-\infty}^{x} \mathrm{e}^{-x^2/2}dx$. If $P_0 = P_1$, which is frequently used for evaluating the security in practical steganalysis, derivative of (3.18) with respect to α is negative. This shows that regardless of the γ's value in (3.17), $\mathrm{P}_{\mathrm{E}_{\mathrm{Bayes}}}$ is a monotonic decreasing function of α in case of equal priors. As a result, a steganographer can minimize α instead of maximizing the $\mathrm{P}_{\mathrm{E}_{\mathrm{Bayes}}}$.

3.3.2.2 Minimax Criterion

In a minimax criterion, the decision rule is the same as the Bayes' rule but for the least favorable priors. The least favorable prior probability of $\mathscr{H}_1$, P_1^L, is defined as the prior probability that maximizes the risk function in (3.16). In case of having differentiable R, it is proven that P_1^L can be 0, 1, or the solution of $R_0 = R_1$. The first two cases will result in $\gamma = \pm\infty$. Therefore, we will consider the third case, which is called an equalizer rule. To find the threshold of the equalizer rule, we need to solve the following equation

$$C_{11}\Big(1-\phi\Big(\frac{\gamma-\alpha/4}{\sqrt{\alpha/2}}\Big)\Big)+C_{01}\phi\Big(\frac{\gamma-\alpha/4}{\sqrt{\alpha/2}}\Big) = C_{00}\Big(1-\phi\Big(\frac{-\gamma-\alpha/4}{\sqrt{\alpha/2}}\Big)\Big)+C_{10}\phi\Big(\frac{-\gamma-\alpha/4}{\sqrt{\alpha/2}}\Big) \tag{3.19}$$

By assuming symmetric costs, $C_{00} = C_{11}$ and $C_{01} = C_{10}$, $\gamma = 0$ is the solution that has the minimum expected risk over all possible prior distributions, and its error is given by

$$\mathrm{P}_{\mathrm{E}_{\mathrm{minimax}}} = \phi(\frac{-\alpha/4}{\sqrt{\alpha/2}}) = \phi(-\sqrt{\frac{\alpha}{8}}) \tag{3.20}$$

which is a monotonically decreasing function of α.

3.3.2.3 Neyman-Pearson Criterion

In Bayesian formulation, the overall expected cost is minimized to find the optimal strategy. In minimax criterion, the case where the prior probabilities are unknown is discussed and the optimal decision is found based on the cost of each decision and the calculated least favorable priors. However, in practice, there might not be any cost defined for each decision. Therefore, we utilize a Neyman-Pearson formulation to find the optimal decision and its corresponding error. In this framework, the detector maximizes the probability of detection, $p(D_{11})$, while keeping the probability of false alarm bounded, $p(D_{10}) \leq l$, where l is the significance level of the test. According to the Neyman-Pearson Lemma, an optimal decision rule exists for any $p(D_{10}) = l$. As a consequence, the decision threshold for such an optimal decision rule can be calculated as

$$\gamma_{\text{Neyman-Pearson}} = -\sqrt{\frac{\alpha}{2}}\phi^{-1}(l) - \frac{\alpha}{4} \tag{3.21}$$

This results in the following total probability of error:

$$\mathrm{P}_{\mathrm{E}_{\text{Neyman-Pearson}}} = \phi\left(-\phi^{-1}(l) - \sqrt{\frac{\alpha}{2}}\right) P_1 + lP_0 \tag{3.22}$$

The same as the two previously discussed criteria, this criterion also results in an error, which is a monotonically decreasing function of α. Based on this behavior, the problem formulation in (3.4) can be simplified and will be discussed in the next section. From now on, for simplicity's sake, we employ

the minimax error calculated in this section for the warden's detection error, $\mathrm{P_E}(\beta_1,\ldots,\beta_n)$, as it does not depend on any variable other than α.

3.4 Gaussian Embedding Model

In this section, a novel image steganography method is introduced based on maximizing the detection error of three optimal detectors shown in the previous section. First, the methodology is derived for embedding in a single image, and then it is extended to batch steganography. Subsequently, based on the theoretical findings for batch steganography, a new algorithm, *AdaBIM*, is proposed. Last but not least, the formulation is extended to the distortion image steganography framework, which makes the algorithm applicable in case of having the cost of embedding in each pixel instead of the residual variance.

3.4.1 Single Image Steganography

The detection error, $\mathrm{P_E}(\beta_1,\ldots,\beta_n)$, is shown to be a monotonic decreasing function of α. Thus, in the proposed Gaussian embedding scenario, the problem of optimal embedding for a fixed payload, shown in (3.4), can be written as

$$\begin{cases} \underset{(\beta_1,\ldots,\beta_n)}{\arg\min}\, \alpha \equiv \underset{(\beta_1,\ldots,\beta_n)}{\arg\min} \sum_{i=1}^{n} \left(\frac{\beta_i^2}{\sigma_i^2}\right)^2 \\ \sum_{i=1}^{n} \mathrm{H}(p_{m_i}) = np \end{cases} \tag{3.23}$$

where p is the relative payload per pixel in nats. Shannon entropy of the hidden message elements (m_i), a Gaussian random variable with variance β_i^2, can be written as:

$$\mathrm{H}(p_{m_i}) = \frac{1}{2}\ln(2\pi \mathrm{e}\beta_i^2) \tag{3.24}$$

The solution of (3.23) using Lagrangian multipliers is the solution of the following equation:

$$\frac{\partial}{\partial \beta_i}\left(\sum_{j=1}^{n}\left(\frac{\beta_j^2}{\sigma_j^2}\right)^2 + \lambda\left(np - \frac{1}{2}\sum_{j=1}^{n}\ln(2\pi \mathrm{e}\beta_j^2)\right)\right) = 0 \tag{3.25}$$

for $i = 1,\ldots,n$, where λ is the Lagrangian multiplier that is calculated using the payload constraint in (3.23), and thus will be shown as a function of the payload, p. The solution of (3.23) is as follows

$$\beta_i^* = \frac{\sqrt[4]{\lambda(p)}}{\sqrt{2}}\sigma_i \quad \text{for } i = 1,\ldots,n \tag{3.26}$$

To achieve optimal security, the message's variance, β_i^2, should be proportional to the pixel's residual variance, σ_i^2. In other words, in a noisy or textured region where residual variances are high, embedding variances are high as well. On the other hand, if a pixel's residual variance is zero, which means it belongs to a smooth region, no embedding takes place. Now that the distribution of the continuous hidden message is determined, the actual message is computed by quantizing the Gaussian distributed message to $\mathscr{Q} = \{-q,\ldots,-1,0,1,\ldots,+q\}$, for any natural number q.

Here is the explanation of the proposed algorithm steps. First, in order to calculate the message variances, β_i, the pixel's residual variance, σ_i, is calculated using any variance estimator such as the

methods proposed in (Sedighi et al., 2015; Sedighi et al., 2016). Then, by assuming a $(2q+1)$-ary embedding scenario where the message is a Gaussian random variable with variance β_i quantized to $\mathscr{Q} = \{-q, \ldots, +q\}$, the following system of equations with $n+1$ equations and variables, $\beta_1, \ldots, \beta_n$ and λ, is solved using Newton-Raphson method.

$$\begin{cases} \beta_i^* = \frac{\sqrt[4]{\lambda(p)}}{\sqrt{2}}\sigma_i \quad \text{for } i = 1, \ldots, n \\ -\sum_{i=1}^{n}\sum_{k=-q}^{q}(p_{m_i}(k)\ln p_{m_i}(k)) = np \end{cases} \tag{3.27}$$

where p is the relative payload in nats per pixel and p_{m_i}, the probability distribution of m_i, is given by

$$p_{m_i}(k) = \frac{\phi(\frac{k+0.5}{\beta_i}) - \phi(\frac{k-0.5}{\beta_i})}{\phi(\frac{q+0.5}{\beta_i}) - \phi(\frac{-q-0.5}{\beta_i})}, \ \ \forall k \in \{-q, \ldots, +q\} \tag{3.28}$$

which is a quantized truncated Gaussian. In other words, $p_{m_i}(k)$ is the probability of changing the i^{th} pixel by k. For implementing the proposed embedding technique by syndrome-trellis codes (Filler et al., 2011), we need to find the embedding costs for all the pixels. Embedding cost, $\rho_i(k)$, is defined as the amount of distortion added to image by changing the i^{th} pixel by k. These costs are calculated by solving the following system of equations, having Gibbs form (Filler and Fridrich, 2010) for all pixels.

$$p_{m_i}(k) = \mathrm{e}^{-\rho_i(k)} / \sum_{d=-q}^{q} \mathrm{e}^{-\rho_i(d)} \qquad \forall i \in \{1, \ldots, n\}, \ \ \forall k \in \{-q, \ldots, q\} \tag{3.29}$$

There are $n \times q$ equations and variables by assuming symmetric costs, and $\rho_i(0) = 0, \ \forall i \in \{1, \ldots, n\}$. Note that finding the costs using Eq. (3.29) guarantees that by increasing the computational com-

plexity, the coding loss can become arbitrarily small. To avoid rapid increase of complexity and any loss of performance for q values higher than 1, the actual embedding can be done using multi-layered STCs schemes which employ a layered-construction to decompose the non-binary case into several binary cases (Filler et al., 2011). Refer to the mentioned work for more information regarding the time complexity and the coding loss of such a coding scheme. However, in this study, the same as all the other conceptual studies, the coding process is disregarded, and the embedding process is simulated by altering the image according to the probabilities shown in (3.28). The pseudo-code of the proposed embedding model is shown in Fig. 1.

The steganalyzer detection error for such an embedder can be computed based on (3.20) and (3.26). In order to get that, the closed-form expression of the Lagrangian multiplier, $\lambda(p)$, is needed. By substituting (3.26) in the payload constraint of (3.23) and utilizing (3.24), the Lagrangian multiplier is given by

$$\lambda(p) = \frac{\mathrm{e}^{4p}}{\left(\pi \mathrm{e} \sqrt[n]{\prod_{i=1}^{n} \sigma_i^2}\right)^2} \tag{3.30}$$

which is a monotonically increasing function of payload as expected. As a result, all the message variances, β, are monotonically increasing functions of the payload as well. Note that based on the assumption of all residual variances being much greater than 1, $\sigma \gg 1$, for very small payloads, $p \ll 1$, lambda is very small, $\lambda \ll 1$. In addition, for large payloads, $p \to \infty$, lambda also approaches infinity. In the following section, based on these asymptotic behaviors, we compare the security of different batch sizes in various payloads.

Figure 1: Pseudo-code for Gaussian Embedding Model.

Input: $\mathbf{c}$ = Cover Image, p = Payload, q, Hidden Message

Output: $\mathbf{s}$ = Stego Image

1: Compute all the pixel residual variances σ_i or embedding costs ρ_i for each cover pixel c_i.

2: **if** using residual variances, σ_i, for embedding **then**

3: Solve (3.27) using Newton-Raphson method to find λ.

4: Calculate all β_i values by (3.26).

5: **else if** using embedding costs, ρ_i, for embedding **then**

6: Substitute the first equation in (3.27) with (3.41), then solve it using Newton-Raphson method to find λ.

7: Calculate all β_i values by (3.41).

8: **end if**

9: Determine all $p_{m_i}(k)$ values for all k and i by (3.28).

10: Encode the hidden message according to the computed change probabilities, p_{m_i}, to get $\mathbf{m} = [m_1, \ldots, m_n]$.

11: Generate the stego image by $\mathbf{s} = \mathbf{c} + \mathbf{m}$.

Based on (3.26) and the definition of α in (3.14), α is given by

$$\alpha^* = \sum_{i=1}^{n} \left(\frac{\beta_i^{*2}}{\sigma_i^2} \right)^2 = \frac{n\lambda(p)}{4} \tag{3.31}$$

which results in the following error in detection for the whole image using (3.20) and (3.30),

$$\mathrm{P_E} = \phi\left(-\sqrt{\frac{n\lambda(p)}{32}} \right) = \phi\left(-\sqrt{\frac{n}{32}} \cdot \frac{\mathrm{e}^{2p}}{\pi \mathrm{e} \sqrt[n]{\prod_{i=1}^{n} \sigma_i^2}} \right) \tag{3.32}$$

It can be concluded that the geometric mean of residual variances, $\sqrt[n]{\prod_{i=1}^{n} \sigma_i^2}$, is a suitability measure of the image for steganography since for a fixed payload, the greater it is, the higher the detection error is. In addition, an image with higher residual variances (having noisier regions) has a higher suitability measure as expected.

To calculate the average detection error for N images, we assume all the images have the same number of pixels, n, for simplicity. Thereby, the closed-form expression for average detection error of the Gaussian embedding scheme is

$$\mathrm{P_E}(M = 1, N, p) = \frac{1}{N} \sum_{l=1}^{N} \phi\left(-\sqrt{\frac{n\lambda_l(p)}{32}} \right) \tag{3.33}$$

where $\lambda_l(p)$ is the Lagrangian multiplier shown in (3.30) for the l^{th} image and M is the batch size which is 1 as no batching took place. In the next section, we discuss greater batch sizes.

3.4.2 Adaptive Batch Size Image Steganography

The problem of optimizing the distribution of a fixed size message among pixels of a single image is discussed in the previous section, and the closed-form expression for detection error is derived. In this section, the results are extended to batch steganography in which the message is spread in multiple images. The state-of-the-art batch steganography method, image merging sender (IMS), batches all the images of a dataset into one group (Cogranne et al., 2017). In this section, we investigate the case when images are batched in groups of size M. Therefore, there are N/M batches of images in a dataset with N images. Without loss of generality, we assume the l^{th} batch contains images with indexes $(l-1)M,\dots,lM-1$. We use IMS for spreading $n\cdot M\cdot p$ nats among M images in each batch (Cogranne et al., 2017; Sharifzadeh et al., 2017). This means that all the images in each batch are merged together and treated as one image. Thus, formulation is the same as (3.23) except that the number of pixels is $n\cdot M$ instead of n. Therefore, the solution is similar to the solution of (3.23) shown in (3.26) and it is given by

$$\beta_{ij}^{*}=\frac{\sqrt[4]{\lambda_{l}^{(M)}(p)}}{\sqrt{2}}\sigma_{ij}\quad\forall i\in\{1,\dots,n\} \tag{3.34}$$

and $\forall j\in\{(l-1)M,\dots,lM-1\}$, where σ_{ij} is the variance of the i^{th} pixel of j^{th} image, and $\lambda_{l}^{(M)}$ is the Lagrangian multiplier for the l^{th} batch derived similar to (3.30) as

$$\lambda_{l}^{(M)}(p)=\frac{\mathrm{e}^{4p}}{\left(\pi\mathrm{e}\sqrt[M]{\prod_{j=(l-1)M}^{lM-1}\sqrt[n]{\prod_{i=1}^{n}\sigma_{ij}^{2}}}\right)^{2}} \tag{3.35}$$

Based on (3.35) and (3.24), payload of the j^{th} image is

$$p_j = np + \frac{n}{2}\ln\left(\frac{\sqrt[n]{\prod_{i=1}^{n}\sigma_{ij}^2}}{\sqrt[M]{\prod_{k=(l-1)M}^{lM-1}\sqrt[n]{\prod_{i=1}^{n}\sigma_{ik}^2}}}\right) \tag{3.36}$$

which shows that in an image with suitability measure, geometric mean of residual variances, greater than the suitability measure of the whole batch, more information than the average payload, $n \cdot p$ nats per image, is embedded. Similarly, the payload of an image with suitability measure smaller than the batch's is smaller than the average payload. This results in all the images in the same batch having equal detection error.

Based on (3.33) and (3.35), the average detection error for the whole database for the proposed embedding can be written as:

$$\mathrm{P_E}(M,N,p) = \frac{1}{N}\sum_{l=1}^{N/M}\sum_{j=(l-1)M}^{lM-1}\phi\left(-\sqrt{n\lambda_l^{(M)}(p)/32}\right) = \frac{M}{N}\sum_{l=1}^{N/M}\phi\left(-\sqrt{n\lambda_l^{(M)}(p)/32}\right) \tag{3.37}$$

In other words, equation (3.37) is the security measure of the algorithm for batch size M and payload p. The following theorem is needed to compare the security of various M's.

Theorem 2. *Effect of Batch Size on Security*

Given any powers of two, M and N, where 2M is less or equal than N, the following statements are true.

(i) $P_E(M,N,p) < P_E(2M,N,p) \qquad p \ll 1$

(ii) $P_E(M,N,p) > P_E(2M,N,p) \qquad p \to \infty$

See Appendix B: Effect of Batch Size on Security for the proof. Based on this theorem, for payloads much smaller than 1, sorted batch sizes according to their detection error in ascending order are $1,2,4,8,...,N$. However, for large enough payloads, this ranking is totally flipped, and $M = 1$ has the highest detection error. Theorem 2 is consistent with the experiments, not only for the Gaussian version of HILL, MiPOD, and SUNIWARD but also their original versions. This flip happens in payloads between 0.75 and 1.5 bits per pixel, depending on the embedding algorithm.

Based on theorem 2, we propose employing different batch sizes in different payloads. This results in a novel **Ada**ptive **B**atch size **I**mage **M**erging steganographer (*AdaBIM*). In *AdaBIM*, the batch size is N for payloads close to zero, and then it gradually decreases as the payload increases until it reaches 1. This is done based on empirical results. Based on (3.36), it is observed that *AdaBIM* spreads the payload non-uniformly among all the images according to their suitability measure (more payload in images with more textured regions) for payloads near zero. However, for large payloads where the batch size is 1, the payload is spread uniformly among images.

The state-of-the-art batch steganography method (IMS) uses $M = N$ for all the payloads. Therefore, *AdaBIM* performs as well as IMS in payloads near zero. However, as the payload increases, we proved *AdaBIM* outperforms IMS. We also demonstrate this by comparing their empirical performances against the state-of-the-art steganalysis method in Sec. 3.5.

3.4.3 Extension to Cost Based Methods

Cost based steganography methods calculate embedding cost instead of residual variances (Li et al., 2014; Pevnỳ et al., 2010b; Holub et al., 2014). In these methods, the steganographer tries to minimize the expected value of a distortion function, $\mathrm{D}(\mathbf{s},\mathbf{c})$, where $\mathbf{s}$ and $\mathbf{c}$ are the stego and cover images re-

spectively. To adapt our framework to be applicable for these methods and boost their performance, we define the distortion to be the expected value of the absolute difference between the pixel intensities the same as prior arts. As a result, the steganography problem for a payload limited sender can be written as:

$$\begin{cases} \underset{(\beta_1,\ldots,\beta_n)}{\arg\min}\, \mathrm{E}[\mathrm{D}(\mathbf{s},\mathbf{c})] = \underset{(\beta_1,\ldots,\beta_n)}{\arg\min} \sum_{i=1}^{n} \mathrm{E}_{m_i|\beta_i}\left[\rho_i|s_i\text{-}c_i|\right] \\ \sum_{i=1}^{n} \mathrm{H}(p_{m_i}) = np \end{cases} \tag{3.38}$$

where ρ_i is the cost of embedding ± 1 in the i^{th} pixel which can be calculated by any of the mentioned algorithms (Li et al., 2014; Pevnỳ et al., 2010b; Holub et al., 2014). Assuming the same Gaussian embedding scenario where $m_i \sim \mathcal{N}(0,\beta_i^2)$, the expected value of the distortion is

$$\mathrm{E}_{m_i|\beta_i}\left[\rho_i|s_i - c_i|\right] = \mathrm{E}_{m_i|\beta_i}\left[\rho_i|m_i|\right] = \rho_i\beta_i\sqrt{\frac{2}{\pi}} \tag{3.39}$$

Using Lagrangian multipliers approach, the problem is translated to

$$\frac{\partial}{\partial\beta_i}\left(\sum_{j=1}^{n}\left(\rho_j\beta_j\sqrt{\frac{2}{\pi}}\right) + \lambda\left(np - \frac{1}{2}\sum_{j=1}^{n}\ln(2\pi\mathrm{e}\beta_j^2)\right)\right) = 0 \tag{3.40}$$

The solution to (3.40) is

$$\beta_i^* = \frac{\lambda(p)}{\rho_i}\sqrt{\frac{\pi}{2}} \tag{3.41}$$

where λ is the Lagrangian multiplier, which can be calculated using the payload constraint in (3.38). The rest of the embedding approach is the same as it is explained in Sec. 3.4.1. The proposed Gaussian steganography method can be applied in both cases of having pixel residual variances and embed-

ding costs. This makes our approach a universal technique for improving all the recent spatial image steganography methods.

3.5 Experiments and Discussions

Throughout this chapter, BOSSbase 1.01 database with $10k$ gray-scale 512×512 pixels images (Bas et al., 2011) is used. To show the performance of each method, the average detection error, defined as the average of false positive and negative rates, is reported. It is evaluated by an ensemble classifier steganalyzer (Kodovsky et al., 2012) with a 10-fold cross-validation, trained on maxSRMd2 feature vectors with $34,671$ elements (Denemark et al., 2014). 4096 and 4096 images are chosen randomly as training/validation, and testing set respectively since throughout this paper we assumed the size of dataset to be a power of 2 and 4096 is the largest power of 2 less than $5k$ (half of the images in the dataset).

Three state-of-the-art content-adaptive image steganography methods: HILL (Li et al., 2014), MiPOD (Sedighi et al., 2016), and SUNIWARD (Holub et al., 2014) are used for evaluations with settings that are shown in the original papers to achieve the best security. HILL algorithm is used with a 3×3 Ker-Bohme high-pass filter and a 3×3 and a 15×15 averaging filters as low-pass filters (Li et al., 2014). MiPOD method utilizes a two dimensional Wiener filter with width, $w = 2$, and medium blocks, which means $p = 9$ and $l = 9$ (Sedighi et al., 2016). SUNIWARD algorithm is used with $\sigma = 1$ (Denemark et al., 2014). For the 7-ary version of HILL, and SUNIWARD, the cost of adding $\pm d$ to the i^{th} pixel is the distortion introduced by changing only the i^{th} pixel by $\pm d$ according to distortion function of the corresponding algorithm. For 7-ary version of MiPOD, the probability of changes is computed by employing

TABLE I: DETECTION ERROR COMPUTED BY STEGANALYSIS USING maxSRMd2 FEATURES IN DIFFERENT PAYLOADS RANGING FROM 0 TO 1 BPP FOR THE PROPOSED GAUSSIAN VERSION OF THE HILL ALGORITHM WITH DIFFERENT q VALUES IN A $(2q{+}1)$-*ary* EMBEDDING SCENARIO. COPYRIGHT

q	payload = **0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.75**	**1**
1	.499±.0025	.488±.0018	.464±.0031	.412±.0033	.351±.0026	.296±.0035	.240±.0029	.132±.0025	.064±.0028
2	.498±.0023	.488±.0018	.467±.0020	.415±.0025	.359±.0028	.303±.0034	.253±.0029	.150±.0035	.084±.0028
3	.499±.0033	.489±.0019	.469±.0025	.417±.0027	.361±.0035	.306±.0040	.256±.0044	.154±.0027	.091±.0034
4	.500±.0030	.489±.0018	.469±.0040	.418±.0027	.361±.0024	.307±.0036	.257±.0050	.155±.0024	.092±.0030
5	.500±.0017	.491±.0027	.469±.0033	.417±.0021	.364±.0036	.309±.0034	.256±.0023	.157±.0033	.094±.0032
6	.500±.0026	.490±.0027	.468±.0029	.418±.0023	.363±.0037	.309±.0034	.258±.0036	.158±.0032	.094±.0028

a 3×3 Fisher information matrix following the same framework used in (Sedighi et al., 2015) for 5-ary embedding.

Embedding in saturated pixels are shown to drop the performance of steganography methods (Sedighi and Fridrich, 2016), therefore in all of the experiments, we avoid embedding in saturated pixels as well as the pixels that will be saturated after embedding. For example, in a 7-ary embedding scheme, all the pixels having the following intensities are avoided: $0, 1, 2, 253, 254, 255$.

In all the batch steganography experiments, the largest batch size tested is 128. The batch sizes greater than 128 are not tried due to computational limitations. We believe this batch size is enough to show sufficient proof for the claimed statements.

3.5.1 Determining Maximum Pixel Change (q)

The Gaussian embedding technique proposed in Sec. 3.4.1 has a controlling parameter q which represents the maximum changes of the cover pixels during embedding. To find the optimal q, we have evaluated the performance of HILL's Gaussian version derived in (3.41) with different settings, $q = 1, \ldots, 6$, for various payloads between 0 and 1 bits per pixel (bpp). The results are presented in Table I. It is observed that for the Gaussian embedding model, the larger the q is, the higher the security is. For example, comparing G-HILL with $q = 1$ and $q = 3$ shows that the former performs significantly better for payloads higher or equal than 0.1 bpp. Similar conclusion can be drawn from Table II for G-MiPOD and G-SUNIWARD. However, the complexity of the coding algorithm will increase if q increases (Filler et al., 2011). Furthermore, the results in Table I also suggest that q values greater than 3 do not result in considerably better security comparing to q equal to 3. Thus, we choose $q = 3$ for the rest of the experiments resulting in septenary (7-ary) embedding scenarios unless mentioned otherwise.

3.5.2 Comparison of Gaussian Embedding with Prior Arts

In this section, we compare the security of three stat-of-the-art image steganography methods, HILL (Li et al., 2014), MiPOD (Sedighi et al., 2016), and SUNIWARD (Holub et al., 2014), with their proposed Gaussian versions. We conduct experiments on all the methods with both ternary ($q = 1$) and septenary ($q = 3$) embedding for various payloads between 0 and 1 bpp. The results are presented in Table II. For the proposed Gaussian versions of these algorithms, we use a prefix of G, e.g. G-HILL. G-HILL and G-SUNIWARD use the embedding cost calculated by HILL and SUNIWARD respectively and they compute the message variances by (3.41). G-MiPOD uses the variance estimator of MiPOD to calculate pixel residual variances and computes the message variances by (3.26). It is observed that

TABLE II: DETECTION ERROR COMPUTED BY STEGANALYSIS USING MAXSRMD2 FEATURES IN PAYLOADS RANGING FROM 0 TO 1 BPP FOR THREE IMAGE STEGANOGRAPHY METHODS AND THEIR PROPOSED GAUSSIAN VERSIONS WITH DIFFERENT Q VALUES IN A $(2Q{+}1)$-*ARY* EMBEDDING SCENARIO. COPYRIGHT ©2019 IEEE.

Embedding	q	payload = **0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.75**	**1**
G-HILL	3	**.499±.0033**	**.489±.0019**	**.469±.0025**	**.417±.0027**	**.361±.0035**	**.306±.0040**	**.256±.0044**	**.154±.0027**	**.091±.0034**
HILL	3	.496±.0014	.486±.0026	.461±.0032	.411±.0023	.353±.0036	.298±.0029	.243±.0030	.142±.0024	.082±.0023
G-HILL	1	.499±.0025	.488±.0018	.464±.0031	.412±.0033	.351±.0026	.296±.0035	.240±.0029	.132±.0025	.064±.0028
HILL	1	.499±.0030	.488±.0019	.464±.0023	.409±.0032	.346±.0029	.292±.0034	.234±.0023	.130±.0030	.062±.0024
G-MiPOD	3	.498±.0023	.483±.0017	.461±.0024	.407±.0023	.351±.0027	.295±.0036	.241±.0034	.145±.0026	.083±.0020
MiPOD	3	.497±.0028	.480±.0019	.453±.0024	.402±.0019	.347±.0030	.289±.0029	.241±.0019	.151±.0044	.092±.0020
G-MiPOD	1	.497±.0030	.482±.0024	.457±.0014	.401±.0023	.346±.0033	.287±.0032	.233±.0024	.124±.0032	.062±.0024
MiPOD	1	.498±.0026	.479±.0017	.451±.0030	.397±.0017	.339±.0018	.279±.0034	.229±.0026	.131±.0031	.066±.0024
G-SUNIWARD	3	.500±.0026	.484±.0036	.456±.0027	.392±.0038	.324±.0025	.263±.0036	.214±.0032	.123±.0030	.067±.0024
SUNIWARD	3	.500±.0025	.482±.0030	.448±.0024	.381±.0019	.313±.0040	.256±.0042	.205±.0033	.120±.0030	.068±.0023
G-SUNIWARD	1	.499±.0015	.485±.0011	.453±.0023	.386±.0025	.319±.0028	.256±.0027	.208±.0026	.109±.0023	.051±.0022
SUNIWARD	1	.499±.0031	.483±.0017	.444±.0023	.373±.0026	.298±.0033	.239±.0032	.187±.0036	.098±.0025	.042±.0018

the statistically significant improvement of the Gaussian embedding scheme, assuming a significance level of 0.05, with $q = 1$ and $q = 3$ emerges in the range of 0.05-0.2 bpp and 0.05-0.1 bpp respectively depending on the algorithm. However, the advantages of the Gaussian embedding model become less significant for HILL algorithm with $q = 1$ in very high payloads of 0.75-1 bpp. MiPOD outperforms G-MiPOD in payloads of 0.75-1 bpp, regardless of the q value. For SUNIWARD with $q = 3$ and payload of 1 bpp, the improvement is not significant. The most secure embedding is G-HILL with $q = 3$ in all the payloads.

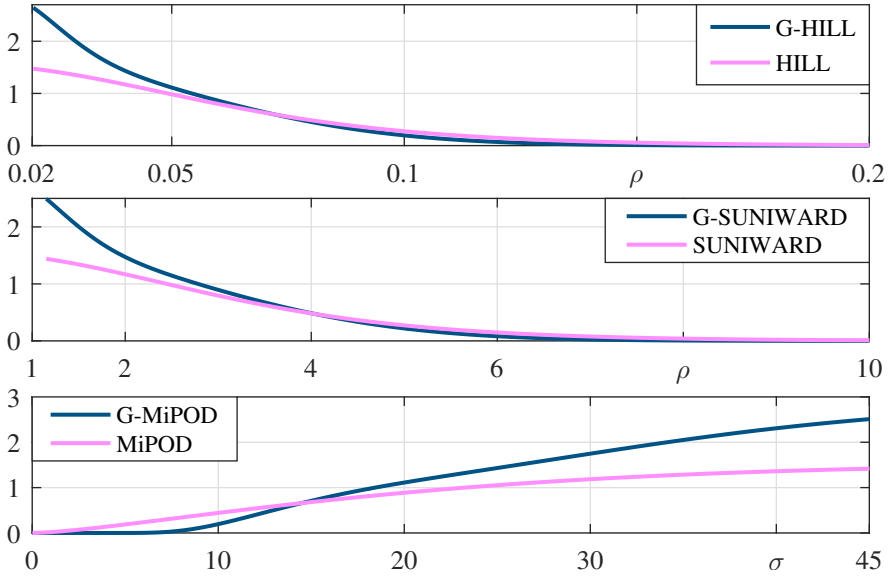

Figure 2: Bits of information embedded in pixels of a single image (1.pgm) versus pixels embedding cost or residual variance for the proposed Gaussian versions and original versions of HILL (top), SUNIWARD (middle), and MiPOD (bottom), when embedding a payload of 0.3 bpp.

We believe that the improvement is due to the fact that the proposed Gaussian method embeds more bits in textured areas (pixels with low embedding costs or equivalently high residual variances) and fewer bits in smooth areas (pixels with high embedding costs or equivalently low residual variances). To confirm that, in Fig. 2, we have plotted the number of bits embedded in each pixel versus pixel's embedding cost computed by HILL and SUNIWARD, and also pixel's residual variances computed by MiPOD for a payload of 0.3 bpp in "1.pgm". It is observed that the proposed Gaussian embedding scheme embeds fewer bits in smooth regions and more in noisy regions comparing to the original methodologies.

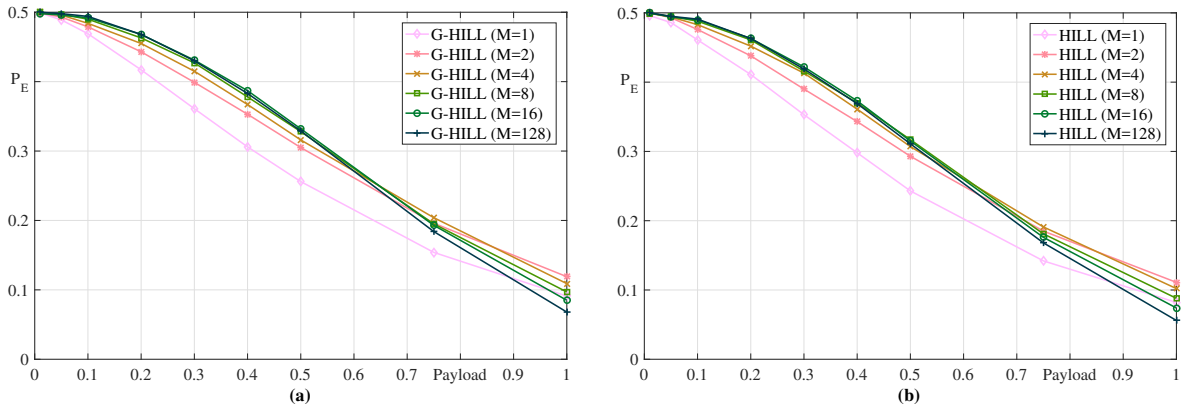

Figure 3: Detection error computed by steganalysis using maxSRMd2 features in different payloads ranging from 0 to 1 bpp for (a) G-HILL (b) HILL algorithms with different batch sizes ($M = 1, 2, 4, 8, 16, 128$). It can be seen that the best performing batch size decreases as the payload increases.

3.5.3 Batch Steganography

In theorem 2, two statements are proven for the effect of batch size on detection error of batch steganography. To examine this theorem in practice, we evaluate the performance of G-HILL and HILL with various batch sizes $(1, 2, 4, 8, 16, 128)$ for different payloads between 0 and 1 bpp. The results, depicted in Fig. 3, indicate that the performance improves by increasing the batch size for payloads from 0 to 0.2 bpp. This behavior is consistent with theorem 2, stating that the detection error is higher for larger batch sizes if the payload is much lower than 1 nat per pixel (equivalently 1.44 bpp). Theorem 2 also states that when payload approaches infinity, the detection error is lower for larger batch sizes. In Fig. 3, this behavior starts to emerge for payloads greater than 0.3 bpp, and in payload of 1 bpp, the greatest batch size (128) has the lowest security comparing to smaller batch sizes. By comparing

TABLE III: DETECTION ERROR OF BATCH STEGANOGRAPHY USING THREE STEGANOGRAPHY METHODS AND THEIR PROPOSED GAUSSIAN VERSION WITH TWO DIFFERENT BATCHING STRATEGIES, IMS WITH BATCH SIZE 128 AND THE PROPOSED *ADABIM* WITH ADAPTIVE BATCH SIZE, COMPUTED BY STEGANALYSIS USING MAXSRMD2 FEATURES IN DIFFERENT PAYLOADS RANGING FROM 0 TO 1 BPP.

Embedding	Batching	payload = **0.01**	**0.05**	**0.1**	**0.2**	**0.3**	**0.4**	**0.5**	**0.75**	**1**
G-HILL	*AdaBIM*	.500±.0022	.498±.0022	.494±.0021	.468±.0029	.431±.0037	.387±.0022	.332±.0033	.204±.0028	.119±.0036
	IMS	.500±.0022	.498±.0022	.494±.0021	.468±.0029	.430±.0014	.383±.0020	.329±.0031	.184±.0035	.068±.0018
HILL	*AdaBIM*	.500±.0024	.495±.0017	.491±.0026	.463±.0025	.422±.0025	.373±.0017	.317±.0027	.191±.0027	.111±.0023
	IMS	.500±.0024	.495±.0017	.491±.0026	.463±.0025	.419±.0027	.369±.0017	.311±.0017	.168±.0027	.056±.0023
G-MiPOD	*AdaBIM*	.499±.0022	.497±.0037	.488±.0020	.453±.0024	.396±.0020	.327±.0023	.256±.0022	.133±.0022	.072±.0024
	IMS	.499±.0022	.497±.0037	.488±.0020	.453±.0024	.392±.0028	.322±.0016	.243±.0024	.070±.0024	.023±.0017
MiPOD	*AdaBIM*	.499±.0032	.497±.0025	.485±.0032	.446±.0020	.383±.0017	.306±.0017	.232±.0035	.129±.0043	.070±.0033
	IMS	.499±.0032	.497±.0025	.485±.0032	.446±.0020	.379±.0011	.299±.0032	.214±.0021	.065±.0019	.018±.0012
G-SUNIWARD	*AdaBIM*	.499±.0024	.497±.0028	.491±.0027	.464±.0025	.429±.0023	.382±.0048	.336±.0036	.221±.0027	.139±.0022
	IMS	.499±.0024	.497±.0028	.491±.0027	.464±.0025	.429±.0023	.380±.0033	.332±.0049	.206±.0028	.101±.0019
SUNIWARD	*AdaBIM*	.500±.0027	.494±.0034	.487±.0022	.457±.0021	.418±.0023	.370±.0026	.324±.0030	.205±.0027	.135±.0013
	IMS	.500±.0027	.494±.0034	.487±.0022	.457±.0021	.417±.0029	.367±.0022	.317±.0021	.180±.0020	.067±.0010

the performances shown in Table II and III, similar behavior is observed for G-MiPOD, MiPOD, G-SUNIWARD, and SUNIWARD algorithms for batch sizes equal to 1 and 128. The beauty of theorem 2 is the fact that it is formulated based on the proposed Gaussian embedding scheme; however, it also holds for the original algorithms as well (HILL, MiPOD, and SUNIWARD).

By taking advantage of this phenomenon, *AdaBIM* is proposed that has significantly higher performance, with a p-value less than or equal to 0.05, for the majority of the payloads between 0 and 1 bpp,

TABLE IV: AVERAGE TIME IN SECONDS SPENT TO EMBED A CODED MESSAGE IN AN IMAGE USING THREE DIFFERENT STEGANOGRAPHY METHODS AND THEIR PROPOSED GAUSSIAN VERSIONS IN DIFFERENT $(2Q+1)$-ARY EMBEDDING SCENARIOS WITH VARIOUS BATCH SIZES (M).

M	q	G-HILL	HILL	G-MiPOD	MiPOD	G-SUNIWARD	SUNIWARD
1	1	0.160	0.037	0.197	0.263	0.219	0.058
	3	0.372	0.074	0.425	0.799	0.457	0.107
128	1	0.097	0.022	0.111	0.206	0.129	0.051
	3	0.252	0.065	0.335	0.665	0.416	0.093

compared to IMS. See Table III. Security improvement in *AdaBIM* rises in the range of 0.2-0.4 bpp depending on the steganography algorithm. Authors believe that advantages of *AdaBIM* could emerge in even lower payloads if IMS batch size (M) is equal to the total number of images in the database as it is defined in its original paper, instead of $M = 128$. However, due to computational limitations, we could not utilize higher M.

The performance improvement of *AdaBIM* comparing to IMS, is due to the fact that in *AdaBIM*, the batch size gradually decreases as the payload increases. In low payloads, the highest batch size ($M = 128$) has the highest security. However, as the payload increases, the optimum M decreases until very high payloads (near 1), where the optimal option is $M = 2$. Note that, the optimal batch size in each payload varies for different methods. Thus, it needs to be calculated separately for each algorithm. It is needless to say that the larger the number of experimented M is, the more precise the optimal M is.

This time-consuming step needs to be done once, and the calculated optimal batch sizes can be used in practice. In other words, finding optimal batch sizes for each payload and algorithm can be seen as a training step whose results can be used in future practices without further calculations.

In this study, we do not impose any assumptions about the warden's knowledge of image sources, and thus, we do not perform any pooled steganalysis experiments. However, we plan to investigate the pooled steganalysis problem in the future, where the derived closed-form expression of the detection error could be utilized to improve the performance when the warden is assumed to know the image sources.

3.5.4 Computational Time

In this section, we compare the amount of time that each embedding algorithm and its Gaussian version spend to embed a $(2q+1)$-ary message in one image. In addition, we also compare their computation time in the batch steganography scenario for $M = 128$. See Table IV. Each time is reported in seconds and calculated by taking the average of time spent per image when embedding payloads ranging from 0.01 to 1 bpp in the whole database. It is observed that all the proposed approaches (G-HILL, G-MiPOD, and G-SUNIWARD) are faster than MiPOD, the state-of-the-art model-based method. G-HILL and G-SUNIWARD are 3 to 5 times slower than HILL and SUNIWARD, respectively, depending on q and M values; however, given the superior security of the Gaussian versions, their computation time is still reasonable. In general, embedding a 7-ary message is 2 to 3 times slower than 3-ary message. For all the embedding methods, the batch steganography scenario with $M = 128$ is faster than $M = 1$ for similar q, which is expected since MATLAB performs vectorization faster than "for" loops.

CHAPTER 4

QUANTIZED GAUSSIAN JPEG STEGANOGRAPHY AND POOL STEGANALYSIS

4.1 Introduction

Steganography is the art of embedding a hidden message in a cover medium without getting detected by the warden (Simmons, 1984). The most common medium for steganography is digital image data due to having high redundancy, which results in high capacity for embedding. In early works in digital image steganography both in spatial and compressed domains, non-adaptive methods were proposed, and they treated all the pixels or DCT coefficients in the same manner. Examples of such methods in spatial domain are (Cheddad et al., 2010; Johnson and Jajodia, 1998) and in JPEG steganography are Jsteg (Upham, 1993), F5 (Westfeld, 2001), and nsF5 (Fridrich et al., 2007). As a result of not taking pixel to pixel or coefficient to coefficient dependencies into consideration, all non-adaptive approaches have low security (Fridrich et al., 2001; Fridrich et al., 2007). Thus, for attaining a higher security performance, adaptive methods have been developed.

Content adaptive steganography methods embed more in textured regions rather than smooth regions of an image to minimize the produced distortion. Distortion minimization embedding is formulated to source coding with a fidelity criterion (Shannon, 1959), and it is solved for a general case by syndrome trellis codes (Filler and Fridrich, 2010; Filler et al., 2011). This coding scheme employs a distortion measure or embedding cost for each cover element and executes embedding accordingly, e.g., a higher embedding rate in low-cost elements. Many methods are available for computing the embedding costs

for image steganography for both spatial and JPEG domains. HILL (Li et al., 2014) and SUNIWARD (Holub et al., 2014) are well-known examples of spatial domain steganography. For JPEG domain steganography, UED (Guo et al., 2014), UERD (Guo et al., 2015), IUERD (Pan et al., 2016), and JUNIWARD (Holub et al., 2014) are among the most frequently used approaches. Even though some of these methods such as HILL, SUNIWARD, and JUNIWARD have the highest security, they are all based on heuristically defined distortions, and therefore, there is no theoretical/statistical measure for their performances. This issue has been addressed in another type of image steganography, called statistical or model-based.

Statistical or model-based image steganography methods mathematically model the cover image and perform embedding while minimizing a distance measure between the cover and the stego image. Examples of such approaches in spatial domain are HUGO (Pevnỳ et al., 2010b), MG (Fridrich and Kodovskỳ, 2013), MVGG (Sedighi et al., 2015), and MiPOD (Sedighi et al., 2016). Denmark et al. introduced the only statistical-based method in JPEG domain called J-MiPOD based on MiPOD statistical model and also proposed algorithms for steganography with pre-cover for both spatial and JPEG domains (SI-MiPOD and SI-J-MiPOD) (Denemark and Fridrich, 2017). In all of the mentioned statistical-based approaches, the optimization problem, defined as minimizing distance between cover and stego images while embedding, results in numerically solvable equations. Thus, there are no closed-form expressions for the embedding probabilities and detection error. Having such an expression, especially for the detection error, would be beneficial in understanding and estimating image steganography behavior as well as batch steganography and pool steganalysis.

In our previous work, we developed a statistical framework for spatial steganography, which resulted in closed-form expressions for embedding probabilities and detection error while achieving state-of-the-art empirical performance (Sharifzadeh et al., 2019b). In this work, we extend our model to JPEG domain and propose a statistical framework for JPEG steganography, which results in closed-form expressions for detection error and embedding probabilities. Our proposed framework can employ any embedding costs defined in the spatial or JPEG domain, and also any residual variance estimator for JPEG steganography. In addition, it can be utilized to model single image and pool steganalysis.

Pool steganalysis is the extension of the steganalysis problem in which the warden knows multiple objects share the same source and, therefore, pools evidence from all of the objects to achieve a higher detection performance. This problem was introduced by Ker, and it is the dual of batch steganography problem in which the steganographer embeds a payload in multiple cover objects (Ker, 2006). Both problems are major research problems in steganography (Ker et al., 2013). Previous studies have proposed methods for ranking multiple sources according to their "guiltiness" (Ker and Pevnỳ, 2011; Ker and Pevny, 2012). However, a more general question remains; which source is guilty? This question was studied under the assumption of an omniscience detector, and it was shown that for finding the guilty source, the average pooling strategy performance is close to optimal for a vast range of hidden message distribution strategies (Pevnỳ and Nikolaev, 2015). In another study, the problem of sequential steganalysis is discussed, and a method is proposed for finding the first stego message in a sequence of objects (Cogranne, 2015). Cogranne et al. formulated the problem in spatial domain and demonstrated that knowledge of the steganographer's strategy increases the performance of pool steganalysis (Cogranne et al., 2017). In contrast to these studies, Zakaria et al. assumed that steganalyzer does not

know the payload spreading strategy and proposed a pooling method that performs close to an omniscience steganalyzer for all the state-of-the-art payload spreading strategies (Zakaria et al., 2019). In all the mentioned works, there is no statistical analysis for modeling pool steganalysis of steganography with state-of-the-art payload spreading strategies in real images.

In this study, we derive the detection error for single image steganalysis mathematically based on the adopted statistical model. We show that the detection error formula is valid for embedding in spatial domain or any linear transformation domain. This allows us to derive a unified closed-form formulation for the optimal pool steganalysis strategy and its error for steganography in any domain. Here, we assume steganalyzer is omniscience, and payload is spread among all of the images uniformly or using the state-of-the-art batch steganography method (Sharifzadeh et al., 2019b). To show the relevance of the results, we employ the derived closed-form expression for pool steganalysis error to approximate the empirical detection error of JPEG steganography, and it's variance for various pool sizes. We demonstrate that our proposed approximation is precise, considering the error of empirical steganalysis. As a result, one can approximate the pool steganalysis results instead of running time-consuming and cumbersome experiments.

In this work, our contribution is threefold:

1. We develop a statistical model for JPEG cover and stego images. Based on that, we extend our previous embedding model for spatial steganography to JPEG steganography and derive the closed-form detection error for such an embedder against an optimal hypothesis detector (Sharifzadeh et al., 2019b). The embedding model is generalized in the sense that it is able to utilize any

embedding cost or variance estimator defined in spatial domain or JPEG domain, and it results in superior security comparing to the state-of-the-art approaches.

2. We extend the closed-form expression of single image steganalysis detection error to pool steganalysis for an omniscience optimal warden. We employ the derived expression to approximate empirical results of pool steganalysis computed by an ensemble classifier steganalyzer based on the empirical detection error of single image steganalysis (Kodovsky et al., 2012). Although the approximation is derived based on our proposed embedding model, it is precise for all the payloads, embedding domains, embedding methods, and steganalysis features as long as the pooling strategy is optimal. It also holds for single image steganography and batch steganography using the state-of-the-art batching strategy, i.e., *AdaBIM* (Sharifzadeh et al., 2019b).

3. We approximate the variance of such a pool steganalyzer and show that it increases as the pool size increases in small payloads employing the proposed detection error approximation. Small payloads are more interesting as they are more applicable than high payloads, which are easily detectable. Therefore, we conclude that although pooling makes the detector more reliable as it decreases detection error, it makes the detector less reliable in the sense that it increases the variance. In other words, pooling makes the steganalyzer less stable. We observed the same behavior in empirical results as well, which confirms the correctness of the approximation.

This chapter is organized as follows. The statistical models for cover and stego message are presented in Sec. 4.2. Based on the proposed Gaussian model, a framework for quantized Gaussian JPEG steganography is introduced in Sec. 4.3. The results are then extended to pool steganalysis in Sec. 4.4. In Sec. 4.5, we provide the empirical results.

4.2 Statistical Models

In this section, we describe the statistical model for the cover image in spatial domain, and subsequently, we derive the probability distribution of DCT coefficients of cover. Also, we derive the statistical model of the stego image in DCT domain by embedding a Gaussian message in each coefficient.

4.2.1 Cover Model

We show an 8-bit gray-scale image in spatial domain by $\mathbf{P} = [P_1, \ldots, P_{n'}]$, where n' is the number of blocks, and P_b is the b^{th} block of 8×8 pixels, $P_b = [p_{bij}]_{8\times 8}$. Note that total number of pixels shown by n is $n = n' \times 64$. All the pixels, p_{bij}, are quatized to $\{0, 1, \ldots, 255\}$. Lets assume $\hat{p}_{bij}$ is an unbiased estimation of the pixel based on its neighbors. We model the estimation errors, defined as $e_{bij} = p_{bij} - \hat{p}_{bij}$, as independent Gaussian random variables, $\mathcal{N}(0, \omega_{bij}^2)$. This model is based on the assumption of fine quantization which is given by $\omega_{bij} \gg 1$, since the quantization step is 1. For a detailed explanation of this model, refer to (Sedighi et al., 2016). Suppose the scaled DCT coefficients of the cover image are similarly shown as $\mathbf{F} = [F_1, \ldots, F_{n'}]$, where $F_b = [f_{bkl}]_{8\times 8}$ and each coefficient, f_{bkl}, is given by

$$f_{bkl} = \frac{1}{q_{kl}} \sum_{i,j=0}^{7} \mathrm{w}(k,l,i,j) p_{bij} \quad \forall k,l \in \{0,1,\ldots,7\} \tag{4.1}$$

where q_{kl} is the kl^{th} element of JPEG quantization matrix and $\mathrm{w}(k,l,i,j)$ is defined as

$$\mathrm{w}(k,l,i,j) = \frac{\mathrm{c}(k)\mathrm{c}(l)}{4} \cos\frac{\pi k(2i+1)}{16} \cos\frac{\pi l(2j+1)}{16} \tag{4.2}$$

where $\mathrm{c}(x)$ is given by

$$\mathrm{c}(x) = \begin{cases} 1/\sqrt{2} & \text{if } x = 0 \\ 1 & o.w. \end{cases} \tag{4.3}$$

By using Eq. (4.1) and the estimation in spatial domain, $\hat{p}_{bij}$, we can estimate the scaled DCT coefficients as well. The estimation, $\hat{f}_{bij}$, is

$$\hat{f}_{bkl} = \frac{1}{q_{kl}} \sum_{i,j=0}^{7} \mathrm{w}(k,l,i,j)\hat{p}_{bij} \tag{4.4}$$

and the residual of the estimation, $x_{bkl} = f_{bkl} - \hat{f}_{bkl}$, is

$$x_{bkl} = \frac{1}{q_{kl}} \sum_{i,j=0}^{7} \mathrm{w}(k,l,i,j)e_{bij} \tag{4.5}$$

which is a linear combination of zero mean Gaussian random variables. Therefore, the distribution of scaled DCT coefficient residual is

$$p_{x_{bkl}}(k) = \frac{1}{\sigma_{bkl}\sqrt{2\pi}} \exp\left(\frac{-k^2}{2\sigma_{bkl}^2}\right) \tag{4.6}$$

where σ_{bkl}, based on all e_{bij} being independent, is given by

$$\sigma_{bkl}^2 = \frac{1}{q_{kl}^2} \sum_{i,j=0}^{7} \mathrm{w}^2(k,l,i,j)\omega_{bij}^2 \tag{4.7}$$

where ω_{bij}^2 is the residual variance of ij^{th} pixel of the b^{th} block in the raw image. The conclusion of DCT residuals having Gaussian distribution, shown in Eq. (4.6), is drawn based on the fact that DCT

is a linear transformation. Thus, the conclusion is valid for any linear transformation of image. Note that Eq. (4.6) is the probability distribution of scaled DCT coefficient residual or estimation error not the coefficient's distribution. It is well known in the literature that the probability distribution of the scaled DCT coefficient of an image is Laplacian (Joshi and Fischer, 1995; Lam and Goodman, 2000). The Gaussian distribution of the residuals or in other words noise in the JPEG domain can alternatively be derived based on the previous studies on DCT coefficients of JPEG images. By analysing JPEG errors, it has been shown that the summation of all the quantization, rounding, and truncation errors has a Gaussian distribution (Luo et al., 2010). In a later work on uncovering JPEG compression history, Li et al. have shown that distribution of the error in JPEG domain depends on the number of compression cycles and quantization matrix elements and it has a Gaussian distribution or a quantized-Gaussian distribution (Li et al., 2015).

Now, we prove that given the independence of the estimation errors in spatial domain, the errors are independent in DCT domain as well. Based on Eq. (4.5), and $\mathrm{E}[e_{bij}e_{bi'j'}] = 0$ for two distinct pixels, the covariance of the errors in the same block is

$$\mathrm{E}[x_{bkl}x_{bk'l'}] = \frac{1}{q_{kl}q_{k'l'}} \sum_{i,j=0}^{7} \mathrm{w}(k,l,i,j)\mathrm{w}(k',l',i,j)\omega_{bij}^2 \tag{4.8}$$

We can assume that ω_{bij}^2 is constant in each block, which is reasonable as in real image ω_{bij}^2 is highly correlated with the energy of the b^{th} block, and it has small variation in each block of 8×8 pixels. At the end of this paragraph, we show that this assumption results in a diagonal covariance matrix, which elements are shown in Eq. (4.8). But in general, the covariance matrix is not necessarily diagonal. It can

be diagonalized/whitened using eigen-decomposition because it is a real symmetric matrix. Suppose the eigen-decomposition of error covariance matrix of b^{th} block is $\mathrm{U}_b\Gamma_b\mathrm{U}_b^T$. Then, the hidden message can be computed using the method, which is explained in Sec. 4.3 based on the whitened error covariance, Γ_b. Then the computed message is multiplied by U_b, quantized and embedded into DCT coefficients. This method is explained thoroughly in Sec. 4.5.3 where we show that it results in slightly better performance only in high payloads comparing to skipping the whitening step. It also drastically increases the time complexity, which is discussed in Sec. 4.5.4. Note that dependant hidden message elements cannot be embedded in dependent cover elements by syndrome trellis codes in practice because of violating the additive distortion assumption of such coding method, although there have been some studies on using STC for non-additive distortion coding for steganography in special cases such as (Zhang et al., 2016). As a result, for the rest of this study, we assume that ω_{bij}^2 is constant in each block, unless mentioned otherwise. Therefore, we can move the ω_{bij}^2 out of the summation in Eq. (4.8). Given that $\sum_{i,j=0}^{7}\mathrm{w}(k,l,i,j)\mathrm{w}(k',l',i,j) \approx 0$ unless $k=k'$ and $l=l'$, the covariance of the errors are

$$\mathrm{E}[x_{bkl}x_{bk'l'}] = \begin{cases} 1 & \text{if } k=k' \text{ and } l=l' \\ 0 & o.w. \end{cases} \tag{4.9}$$

Thus all x_{bkl} are independent zero-mean Gaussian random variables with variances shown in Eq. (4.7). Note that the residual variances, ω_{bij}^2, can be calculated using any variance estimator such as the ones proposed in (Sedighi et al., 2015; Sedighi et al., 2016). In the following two subsections, we discuss the cases where the cost of embedding in spatial domain and DCT domain is given.

4.2.1.1 Embedding Cost in Spatial Domain

For the proposed Gaussian embedding model, any embedding cost in spatial domain, e.g. costs defined in (Li et al., 2014; Holub et al., 2014), can also be used as a proxy to calculate ω_{bij}^2. As we have shown in our previous work, $\omega_{bij}^2 \approx 1/\eta_{bij}^2$ where η_{bij} is the cost of changing the ij^{th} pixel of the b^{th} block by 1 in the raw image. Therefore, based on Eq. (4.7), the DCT residual variances are derived as follows in case of having spatial domain embedding costs, i.e. η_{bij}, instead of residual variances, ω_{bij}^2.

$$\sigma_{bkl}^2 = \frac{1}{q_{kl}^2} \sum_{i,j=0}^{7} \mathrm{w}^2(k,l,i,j) \frac{1}{\eta_{bij}^2} \tag{4.10}$$

4.2.1.2 Embedding Cost in DCT domain

In case of having the cost of embedding in each DCT coefficient as η_{bij}, which is the cost of changing the scaled ij^{th} DCT coefficient of the b^{th} block, the DCT residual variance is given by

$$\sigma_{bkl}^2 = \frac{1}{\eta_{bij}^2} \tag{4.11}$$

based on our previous work where we showed the reciprocal of the squared embedding cost can be used as a proxy for calculating residual variance (Sharifzadeh et al., 2019b). In Eq. (4.11), η_{bij} can be computed by any of the methods proposed in (Guo et al., 2014; Guo et al., 2015).

As a result of Equations (4.7), (4.10), and (4.11), the embedding model is universal, and it works with embedding costs or residual variances calculated in the spatial domain, or the embedding costs calculated in the DCT domain.

This statistical cover model is violated in practice in smooth or saturated blocks because of assuming unbounded DCT coefficients and $\sigma_{bkl} \gg 1$. However, our proposed method avoids embedding in those regions, which is covered thoroughly in Sec. 4.3.

4.2.2 Stego Model

We show hidden message by $\mathbf{M} = [M_1, \ldots, M_n]$, where M_b is the b^{th} block of 8×8 message elements, $M_b = [m_{bij}]_{8\times 8}$. In contrast to all the previous works in which hidden message elements are modeled as discrete random variables, we model them, m_{bij}, as Gaussian random variables with variances β_{bij} distributed according to

$$p_{m_{bij}}(k) = \frac{1}{\beta_{bij}\sqrt{2\pi}} \exp\left(\frac{-k^2}{2\beta_{bij}^2}\right) \tag{4.12}$$

The scaled DCT coefficients of the stego image is the summation of the cover coefficients with hidden message elements, i.e. $\mathbf{S} = \mathbf{F} + \mathbf{M}$. Hence, the kl^{th} scaled DCT coefficient residual of the b^{th} block is $y_{bkl} = x_{bkl} + m_{bkl}$, and based on Eq. (4.6) and Eq. (4.12), its probability distribution is derived as

$$p_{y_{bkl}}(k) \propto \frac{1}{\sqrt{2\pi(\sigma_{bkl}^2 + \beta_{bkl}^2)}} \exp\left(\frac{-k^2}{2(\sigma_{bkl}^2 + \beta_{bkl}^2)}\right) \tag{4.13}$$

in which we assume unbounded quantization levels and $\sqrt{\sigma_{bkl}^2 + \beta_{bkl}^2} \gg 1$. In the next section, we find $\mathbf{B} = [B_1, \ldots, B_n]$, where B_b is the b^{th} block of 8×8 message elements variances, $B_b = [\beta_{bij}]_{8\times 8}$, that maximizes the security for a payload limited sender.

4.3 Methodology

In this section, we discuss the problem of JPEG steganography in a single image which is formulated into the following constrained optimization.

$$\begin{cases} \arg\max_{\mathbf{B}} \mathrm{P_E}(\mathbf{B}) \\ \sum_{b=1}^{n'} \sum_{i,j=0}^{7} \mathrm{H}(p_{m_{bij}}) = \nu p \end{cases} \tag{4.14}$$

Where ν is the number of non-zero AC DCT coefficients, $\mathrm{P_E}$ is the detection error of the steganalyzer derived in the following section, $\mathrm{H}(p_{m_{bij}})$ is the entropy of a random variable with probability distribution $p_{m_{bij}}$ in natural unit of information (nats) and p is the relative payload in nats per non-zero AC coefficients.

Assume the worst-case scenario in which the steganalyzer is omniscience and knows all the cover and hidden message probability distributions, i.e., $p_{x_{bij}}$ and $p_{m_{bij}}$. To compute the detection error of this steganalyzer, i.e. $\mathrm{P_E}(\mathbf{B})$, suppose that it employs a likelihood ratio test (LRT) to decide whether the received image is a cover or it conveys a hidden message, shown by null hypothesis ($\mathscr{H}_0$) and alternative hypothesis ($\mathscr{H}_1$) respectively.

Suppose $\mathbf{R} = [R_1, \ldots, R_n]$ are the residuals of received image's DCT coefficients where R_b is the b^{th} block of 8×8 residuals, i.e. $R_b = [r_{bij}]_{8\times 8}$, and they are statistically independent. Therefore, the

likelihood ratio for all the DCT coefficients can be simplified as $\prod_{b=1}^{n'}\prod_{i,j=0}^{7}\Lambda_{bij}$ where Λ_{bij} is the likelihood ratio for the ij^{th} residual of b^{th} block. Given Eq. (4.6) and Eq. (4.13), Λ_{bij} is

$$\Lambda_{bij}=\frac{p_{y_{bij}}(r_{bij})}{p_{x_{bij}}(r_{bij})}=\sqrt{\frac{\sigma_{bij}^2}{\sigma_{bij}^2+\beta_{bij}^2}}\exp\left(\frac{-r_{bij}^2}{2}\frac{-\beta_{bij}^2}{\sigma_{bij}^2(\sigma_{bij}^2+\beta_{bij}^2)}\right) \tag{4.15}$$

Thus the natural logarithm of Λ_{bij} is given by

$$\ln\Lambda_{bij}=\ln\sqrt{\frac{\sigma_{bij}^2}{\sigma_{bij}^2+\beta_{bij}^2}}+\frac{\beta_{bij}^2}{2\sigma_{bij}^2(\sigma_{bij}^2+\beta_{bij}^2)}r_{bij}^2 \tag{4.16}$$

In Eq. (4.16), r_{bij} is Gaussian random variable. Thus, $\ln\Lambda_{bij}$ is a constant term plus a Gamma distributed term with shape (k_{bij}) and scale (θ_{bij}) parameters, i.e. $\Gamma(k_{bij},\theta_{bij})$. In both cases of $\mathscr{H}_0$ and $\mathscr{H}_1$, the shape parameter is equal to 0.5, i.e. $k_{bij}=0.5$. However, the scale parameter, θ_{bij}, depends on the variance of r_{bij}, and it is given by

$$\theta_{bij}=\begin{cases}\beta_{bij}^2/(\sigma_{bij}^2+\beta_{bij}^2) & \text{if } \mathscr{H}_0 \text{ is true.}\\ {\beta_{bij}}^2/\sigma_{bij}^2 & \text{if } \mathscr{H}_1 \text{ is true.}\end{cases} \tag{4.17}$$

Based on our previous work (Sharifzadeh et al., 2019b), for large enough number of DCT coefficients (or pixels), the following approximation for probability distribution of $\sum_{b=1}^{n'}\sum_{i,j=0}^{7}\ln\Lambda_{bij}$ holds.

$$\sum_{b=1}^{n'}\sum_{i,j=0}^{7}\ln(\Lambda_{bij})\xrightarrow{\text{d}}\begin{cases}\mathscr{N}(\frac{-1}{4}\alpha,\ \frac{1}{2}\alpha) & \text{if } \mathscr{H}_0 \text{ is true}\\ \mathscr{N}(\frac{+1}{4}\alpha,\ \frac{1}{2}\alpha) & \text{if } \mathscr{H}_1 \text{ is true}\end{cases} \tag{4.18}$$

$$\alpha = \sum_{b=1}^{n'} \sum_{i,j=0}^{7} \left(\frac{\beta_{bij}^2}{\sigma_{bij}^2} \right)^2 \tag{4.19}$$

Eq. (4.18) shows that embedding hidden message in scaled DCT coefficients changes variance of detectors output, however the mean stays the same. This behaviour is similar to the one explained by shift hypothesis for embedding in spatial domain (Ker, 2006).

A steganalyzer utilizing a LRT compares the likelihood ratio with a decision threshold to figure out if there is hidden message in an image or not. The natural logarithm of the LRT is given by

$$\sum_{b=1}^{n'} \sum_{i,j=0}^{7} \ln(\Lambda_{bij}) \underset{\mathscr{H}_0}{\overset{\mathscr{H}_1}{\gtrless}} \text{decision threshold} \tag{4.20}$$

It has been previously shown that for the given LRT, using minimax, one of the most common optimal decision criteria, the decision threshold equal to 0 results in the lowest expected risk over all possible priors (Sharifzadeh et al., 2019b). As a result, based on Eq. (4.18) and Eq. (4.19), the detection error for the optimal detector is given by

$$\hat{\mathrm{P}}_{\mathrm{E}} = \phi\left(\frac{-\alpha/4}{\sqrt{\alpha/2}}\right) = \phi\left(-\sqrt{\frac{\alpha}{8}}\right) \tag{4.21}$$

where ϕ is the cumulative density function of standard normal distribution. $\hat{\mathrm{P}}_{\mathrm{E}}$ shown in Eq. (4.21) is monotonically decreasing as α increases. Thus, to achieve a more secure steganography method, we can minimize α instead of maximizing the error of the steganalyzer. The same conclusion can be made

employing other common optimal decision rules such as Bayes and Neyman–Pearson. Consequently, the problem shown in Eq. (4.14) can be simplified as

$$\begin{cases} \arg\min_{\mathbf{B}} \alpha \equiv \arg\min_{\mathbf{B}} \sum_{b=1}^{n'} \sum_{i,j=0}^{7} \left(\frac{\beta_{bij}^2}{\sigma_{bij}^2}\right)^2 \\ \sum_{i=1}^{n'} \sum_{i,j=0}^{7} \mathrm{H}(p_{m_{bij}}) = \nu p \end{cases} \tag{4.22}$$

The solution of Eq. (4.22) using Lagrangian multiplier method is given by

$$\beta_{bij}^{*} = \frac{\sqrt[4]{\lambda(p)}}{\sqrt{2}} \sigma_{bij} \tag{4.23}$$

where λ is the Lagrangian multiplier determined by the payload constraint in Eq. (4.22) as a function of the relative payload, p, and it is derived as follows

$$\lambda(p) = \frac{\mathrm{e}^{4p}}{\left(\sqrt[\nu]{\prod_{b=1}^{n'} \prod_{i,j=0}^{7} \pi \mathrm{e} \sigma_{bij}^2}\right)^2} \tag{4.24}$$

Therefore

$$\alpha = \sum_{b=1}^{n'} \sum_{i,j=0}^{7} \left(\frac{\beta_{bij}^2}{\sigma_{bij}^2}\right)^2 = 64n' \frac{\lambda(p)}{4} = \frac{n\lambda(p)}{4} \tag{4.25}$$

$$\hat{\mathrm{P}}_{\mathrm{E}} = \phi\left(-\sqrt{\frac{\alpha}{8}}\right) = \phi\left(-\sqrt{\frac{n\lambda(p)}{32}}\right) \tag{4.26}$$

where $n = n' \times 64$ is the total number of pixels or DCT coefficients. The closed-form expression for detection error of steganalysis shown in Eq. (4.26) is derived based on the Gaussian distribution of cover elements residuals and hidden message elements. In addition, the Gaussian distribution is drawn from

the fact that DCT is a linear transformation. As a result, the closed-form expression for detection error of steganalysis shown in Eq. (4.26) is valid for embedding using the proposed adopted model in raw image, or any linear transformation of image such as DCT. Based on this generalized error formulation, in the next section, we develop an statistical model for pool steganalysis which is valid for steganography in raw image data or any linear transformation of image data.

Eq. (4.23) shows that the message variance is proportional to the DCT coefficients residual variance. As a result, we embed more nats by adding a Gaussian with higher variance in noisy coefficients comparing to coefficients with small variance.

Now that the problem is solved in the continuous domain, we translate the problem, shown in Eq. (4.22), to discrete domain by quantizing hidden message to $\mathscr{Q} = \{-q, \ldots, -1, 0, 1, \ldots, +q\}$, as follows

$$\begin{cases} \beta^*_{bij} = \frac{\sqrt[4]{\lambda(p)}}{\sqrt{2}} \sigma_{bij} \quad \forall b, i, j \\ -\sum_{b=1}^{n'} \sum_{i,j=0}^{7} \sum_{k=-q}^{q} (p_{m_{bij}}(k) \ln p_{m_{bij}}(k)) = \nu p \end{cases} \tag{4.27}$$

$$p_{m_{bij}}(k) = \frac{\phi(\frac{k+0.5}{\beta_{bij}}) - \phi(\frac{k-0.5}{\beta_{bij}})}{\phi(\frac{q+0.5}{\beta_{bij}}) - \phi(\frac{-q-0.5}{\beta_{bij}})} \tag{4.28}$$

Eq. (4.28) is a truncated Gaussian random variable indicating the probability of changing the ij^{th} coefficient of b^{th} block by k. We utilize the Newton-Raphson method to find the Lagrangian multiplier, $\lambda(p)$, which determines all hidden message variances, i.e. β_{bij}, and distributions, i.e. $p_{m_{bij}}$. To be able to take advantage of practical embedding methods such as syndrome-trellis codes (STCs) (Filler et al., 2011) for real world implementation of the proposed embedding model, the cost of changing each coefficient is required. We show cost of changing the ij^{th} coefficient of b^{th} block by k by $\rho_{bij}(k)$. Assuming

symmetric costs, i.e. $\rho_{bij}(k) = \rho_{bij}(-k)$, there are $64 \times n \times q$ variables and equations having Gibbs form given by

$$p_{m_{bij}}(k) = \mathrm{e}^{-\rho_{bij}(k)} / \sum_{d=-q}^{q} \mathrm{e}^{-\rho_{bij}(d)}, \tag{4.29}$$

$\forall b \in \{1,\ldots,n\},\ \forall i,j \in \{0,\ldots,7\},\ \forall k \in \{1,\ldots,q\}$. Computing these costs, allows us to utilize STCs for the actual embedding when $q = 1$ and multi-layered STCs for $q > 1$ (Filler et al., 2011). However in this manuscript, similar to conceptual studies in steganography, we disregard the coding process and change the coefficients according to the change rates shown in Eq. (4.28). A summary of our proposed method is shown in Fig. 4.

4.4 Pool Steganalysis

In the previous section, we have derived the closed-form solution for JPEG steganography against optimal single image steganalysis and its error. In this section, we discuss the case where the steganalyzer also knows the source of a pool of images. Then, the detection error is derived for an arbitrarily sized pool of images, in which the images are all stego or cover. The notation is the same as before except that we show the image number using superscript in parenthesis, e.g., $\lambda^{(i)}$ is the Lagrangian multiplier for the i^{th} image. In addition, we show the detection error for pool size l by $\hat{\mathrm{P}}_{\mathrm{E}}(l)$ when it is theoretically estimated and by $\mathrm{P}_{\mathrm{E}}(l)$ when it is empirically computed. The following theorem explains how to derive $\hat{\mathrm{P}}_{\mathrm{E}}(l)$ and what would be its error's behavior.

Theorem 3. *Statistical Model for Pool Steganalysis Detector's Error and Variance*

Suppose that l images are sent from the same source and in the case of being stego images, they carry the same amount of hidden message or embedding has been done using the state-of-the-art batch

Figure 4: Pseudo-code of the JPEG Gaussian Embedding Model

Input: $\mathbf{F}$ = Cover Image Scaled DCT Coefficients, p = Payload, q, Hidden Message

Output: $\mathbf{S}$ = Stego Image Scaled DCT Coefficients

if using residual variances in spatial domain, ω_{bij}^2, for embedding **then**

    derive residual variances in DCT domain by Eq. (4.7).

else if using embedding costs in spatial domain, η_{bij}, for embedding **then**

    derive residual variances in DCT domain by Eq. (4.10).

else if using embedding costs in DCT domain, η_{bij}, for embedding **then**

    derive residual variances in DCT domain by Eq. (4.11).

end if

Find λ by solving the system of equations shown in Eq. (4.27) using Newton–Raphson method.

Calculate all β_{bij} values for all $b,\ i$, and j by Eq. (4.23).

Determine all $p_{m_{bij}}(k)$ values for all $b,\ i,\ j$, and k by Eq. (4.28).

Encode hidden message according to the determined change rates, $p_{m_{bij}}$, to get $\mathbf{M} = [M_1, \ldots, M_n]$.

Compute the stego image scaled DCT coefficients by $\mathbf{S} = \mathbf{F} + \mathbf{M}$.

steganography method (Sharifzadeh et al., 2019b). An omniscience optimal detector should examine the images together and decide based on the summation of all the images detection statistics. The error of such optimal detector can be approximated by

$$\hat{P}_E(l) \approx \phi\left(\phi^{-1}\left(\hat{P}_E(1)\right)\sqrt{l}\right) \tag{4.30}$$

The standard deviation of $\hat{P}_E(l)$, i.e. $\hat{\sigma}_l$, as a function of the standard deviation of $\hat{P}_E(1)$, i.e. $\hat{\sigma}_1$, is given by

$$\hat{\sigma}_l \approx \sqrt{l}\exp\left(-\frac{1}{2}\left(\phi^{-1}\left(\hat{P}_E(1)\right)\right)^2(l-1)\right)\hat{\sigma}_1 \tag{4.31}$$

which is an increasing function of the pool size (l) until $l = l_0$ and a decreasing function afterwards, where l_0 is written as

$$l_0 = \left(\phi^{-1}\left(\hat{P}_E(1)\right)\right)^{-2} \tag{4.32}$$

See Appendix C: Statistical Model for Pool Steganalysis Detector's Error and Variance for the proof. Given that Theorem 3 is true for steganography in raw image data or any linear transformation of image data, its true for JPEG steganography as well. The beauty of this approximation is that utilizing it, one can run only one experiment employing an ensemble classifier steganalyzer (Kodovsky et al., 2012) to find $\hat{\mathrm{P}}_{\mathrm{E}}(1)$, and plug the result in Eq. (4.30) to find $\hat{\mathrm{P}}_{\mathrm{E}}(l)$ for any l. In Sec. 4.5.5, we show that although this approximation is based on the Gaussian embedding model and optimal pool steganalysis, it works for any embedding method, as long as the same steganalyzer is employed for all the pool sizes using the explained optimal pooling strategy.

Another conclusion that can be drawn from Theorem 3 is that although pool steganalysis gives better results comparing to single image steganalysis, it suffers from an increasing variance as the pool size increases for some payloads. To the best of authors' knowledge, this phenomenon has never been discussed nor been formulized in the literature. The variance increases until pool size reaches l_0, shown in Eq. 4.32 and Fig. 8, and it decreases afterwards. In Sec. 4.5.5, we observe that this statistical model and its results are aligned with the empirical results.

4.5 Experiments and Discussion

Throughout this paper, we use the BOSSbase 1.01 database containing $10k$ gray-scale 512×512 pixels images (Bas et al., 2011). All the images are compressed to JPEG with two quality factors, 75 and 95. Performance evaluations are done using an ensemble of classifiers with 10-fold cross-validation trained on steganalysis features extracted from $5k$ images chosen randomly as training/validation set and tested on features extracted from the rest $5k$ images (Kodovsky et al., 2012). We utilize two different state-of-the-art JPEG steganalysis feature vectors DCTR (Holub and Fridrich, 2014) and GFR (Song et al., 2015) with 8000 and 17000 elements, respectively. Performances are reported by the classifier average detection error defined as the mean of false alarm and missed detection rates in payloads ranging from 0.05 to 1, i.e. $p \in \{0.05, 0.1, 0.2, 0.3, 0.4, 0.5, 0.75, 1\}$, bits per non zero AC coefficient (bpnzac).

To find out if a performance improvement is statistically significant, we employ the significance level of 0.05. For all the performance evaluations in this article, sample sizes are 10, and the standard deviations of samples are in the range of 0.001 to 0.005. In the worst-case scenario of comparing two performances, both having a standard deviation of 0.005, if the difference between them is greater than 0.0047, it is statistically significant.

For all the experiments, we employ five different JPEG steganography methods. The first two are the two state-of-the-art JPEG steganography methods, i.e., UERD (Guo et al., 2015) and JUNIWARD (Holub et al., 2014), with their optimal parameters for achieving best security. The next two methods are based on the mentioned methods, UERD and JUNIWARD, but utilizing our proposed quantized Gaussian embedding, we show them by G-UERD and G-JUNIWARD respectively. In addition to these four methods, we also experiment G-JHILL, which employs the proposed quantized Gaussian embedding model using spatial domain embedding cost computed by the HILL algorithm, as shown in Sec 4.2.1.1. HILL algorithm is used with a 3×3 Ker-Bohme high-pass filter and a 3×3 and a 15×15 averaging low-pass filters (Li et al., 2014).

4.5.1 Determining Maximum DCT Coefficient Change (q)

The parameter q of the proposed quantized Gaussian embedding model summarized in Fig. 4 controls the maximum amount that DCT coefficients will be changed during embedding. In other words, our embedding model is a $(2q+1)$-ary embedding. To determine optimal q value for achieving highest security, we evaluate all the JPEG steganography methods with different q values, i.e. $q \in \{1,2,3\}$. The results are presented in Table V. It can be seen that for our proposed Gaussian embedding model, reported in the top three sections of the table, i.e. G-UERD G-JUNI G-JHILL, higher q values results in higher performance, however the improvement is not significant for lower payloads. The security is significantly improved only for JPEG quality factor of 95 and in higher payloads ($p \geq 0.5$) which are less important comparing to lower payloads due to high detection probability. Note that using higher q values results in a more complex encoding algorithm (Filler et al., 2011). As a result, for the rest of the

TABLE V: Detection error of steganalysis using GFR features for various payloads (p) and various embedding algorithms with different q values resulting in a $(2q+1)$-*ary* embedding scenario.

	JPEG Quality Factor = 75									JPEG Quality Factor = 95								
Algorithm	q	p = .05	0.1	0.2	0.3	0.4	0.5	0.75	1	q	p = .05	0.1	0.2	0.3	0.4	0.5	0.75	1
G-UERD	1	0.4600	0.4037	0.2837	0.1814	0.1065	0.0603	0.0133	0.0048	1	0.4876	0.4663	0.4127	0.3483	0.2802	0.2130	0.0797	0.0218
	2	0.4612	0.4074	0.2838	0.1789	0.1075	0.0637	0.0135	0.0043	2	0.4877	0.4648	0.4111	0.3483	0.2793	0.2141	0.0903	0.0328
	3	0.4581	0.4082	0.2840	0.1803	0.1082	0.0606	0.0136	0.0046	3	0.4864	0.4654	0.4126	0.3474	0.2807	0.2135	0.0967	0.0362
G-JUNI	1	0.4637	0.4085	0.2870	0.1885	0.1081	0.0596	0.0115	0.0034	1	0.4914	0.4767	0.4335	0.3782	0.3141	0.2446	0.0990	0.0292
	2	0.4614	0.4062	0.2880	0.1826	0.1094	0.0606	0.0125	0.0033	2	0.4925	0.4757	0.4341	0.3764	0.3149	0.2545	0.1190	0.0457
	3	0.4595	0.4063	0.2895	0.1831	0.1097	0.0615	0.0131	0.0044	3	0.4938	0.4747	0.4354	0.3758	0.3161	0.2567	0.1242	0.0520
G-JHILL	1	0.4650	0.4134	0.2986	0.1893	0.1139	0.0631	0.0131	0.0048	1	0.4943	0.4794	0.4437	0.3945	0.3354	0.2727	0.1336	0.0439
	2	0.4640	0.4138	0.2968	0.1908	0.1160	0.0689	0.0156	0.0047	2	0.4939	0.4805	0.4436	0.3943	0.3384	0.2775	0.1519	0.0705
	3	0.4637	0.4141	0.2971	0.1896	0.1174	0.0686	0.0163	0.0061	3	0.4944	0.4802	0.4435	0.3941	0.3395	0.2798	0.1554	0.0787
UERD	1	0.4560	0.3942	0.2729	0.1874	0.1179	0.0665	0.0169	0.0064	1	0.4857	0.4655	0.4121	0.3466	0.2788	0.2114	0.0845	0.0216
	2	0.4491	0.3807	0.2464	0.1629	0.0974	0.0547	0.0117	0.0044	2	0.4855	0.4654	0.4083	0.3384	0.2701	0.2053	0.0846	0.0293
	3	0.4480	0.3785	0.2422	0.1579	0.0913	0.0510	0.0109	0.0037	3	0.4883	0.4605	0.4011	0.3279	0.2593	0.1920	0.0782	0.0305
JUNI	1	0.4623	0.4056	0.2813	0.1852	0.1052	0.0582	0.0108	0.0018	1	0.4948	0.4796	0.4324	0.3749	0.3089	0.2357	0.0852	0.0153
	2	0.4602	0.4000	0.2700	0.1799	0.1029	0.0555	0.0104	0.0027	2	0.4951	0.4796	0.4304	0.3754	0.3031	0.2335	0.0949	0.0313
	3	0.4585	0.3987	0.2645	0.1753	0.0991	0.0500	0.0093	0.0023	3	0.4925	0.4796	0.4316	0.3740	0.2978	0.2271	0.0925	0.0341

experiments, we only consider $q = 1$ which has similar performance comparing to $q = 2$ and $q = 3$ for most of the payloads and requires a less complex encoder.

We have also shown the results of different $(2q + 1)$-ary embedding scenarios for non-Gaussian embedding algorithms, UERD, and JUNIWARD, in the bottom two sections of the Table V. It can be concluded that higher q values result in lower security for almost all the payloads.

TABLE VI: Detection error of steganalysis using GFR features for various payloads (p) and various embedding algorithms.

	JPEG Quality Factor = 75								JPEG Quality Factor = 95							
Algorithm	p = .05	0.1	0.2	0.3	0.4	0.5	0.75	1	p = .05	0.1	0.2	0.3	0.4	0.5	0.75	1
UERD	0.4560	0.3942	0.2729	0.1874	0.1179	0.0665	0.0169	0.0064	0.4880	0.4655	0.4121	0.3466	0.2788	0.2114	0.0845	0.0216
G-UERD	0.4600	0.4037	0.2837	0.1814	0.1065	0.0603	0.0133	0.0048	0.4876	0.4663	0.4127	0.3483	0.2802	0.2130	0.0797	0.0218
JUNI	0.4623	0.4056	0.2813	0.1852	0.1052	0.0582	0.0108	0.0018	0.4948	0.4796	0.4324	0.3749	0.3089	0.2357	0.0852	0.0153
G-JUNI	0.4637	0.4085	0.2870	0.1885	0.1081	0.0596	0.0115	0.0034	0.4914	0.4767	0.4335	0.3782	0.3141	0.2446	0.0990	0.0292
G-JHILL	0.4650	0.4134	0.2986	0.1893	0.1139	0.0631	0.0131	0.0048	0.4943	0.4794	0.4437	0.3945	0.3354	0.2727	0.1336	0.0439

TABLE VII: Detection error of steganalysis using DCTR features for various payloads (p) and various embedding algorithms.

	JPEG Quality Factor = 75								JPEG Quality Factor = 95							
Algorithm	p = .05	0.1	0.2	0.3	0.4	0.5	0.75	1	p = .05	0.1	0.2	0.3	0.4	0.5	0.75	1
UERD	0.4698	0.4211	0.3257	0.2417	0.1654	0.1039	0.0240	0.0056	0.4958	0.4852	0.4509	0.4001	0.3313	0.2615	0.0981	0.0228
G-UERD	0.4750	0.4350	0.3379	0.2422	0.1614	0.0982	0.0274	0.0062	0.4948	0.4869	0.4497	0.4022	0.3400	0.2706	0.1084	0.0301
JUNI	0.4801	0.4494	0.3560	0.2570	0.1715	0.1040	0.0187	0.0023	0.4960	0.4866	0.4613	0.4158	0.3602	0.2923	0.1030	0.0128
G-JUNI	0.4814	0.4543	0.3637	0.2647	0.1780	0.1076	0.0196	0.0035	0.4954	0.4891	0.4625	0.4216	0.3722	0.3103	0.1307	0.0335
G-JHILL	0.4819	0.4549	0.3646	0.2678	0.1810	0.1114	0.0191	0.0042	0.4982	0.4892	0.4610	0.4259	0.3731	0.3167	0.1444	0.0409

These observations suggest that the proposed quantized Gaussian embedding model is more accurate compared to the widely used Gibbs form (Filler and Fridrich, 2010) for calculating embedding probabilities.

4.5.2 Comparison of Quantized Gaussian Embedding with Prior Arts

In this section, we compare the security of the proposed steganography method with the state-of-the-art JPEG steganography methods against steganalysis using DCTR and GFR features. We conclude that using the proposed embedding model results in performance improvement for all the algorithms in most of the payloads. We also show that the proposed G-JHILL method outperforms all the previously developed methods in all the payloads.

We compare the detection error of UERD, G-UERD, JUNIWARD, G-JUNIWARD, and G-JHILL using GFR features in Table VI. For the UERD algorithm, the proposed Gaussian version (G-UERD) outperforms UERD significantly in payloads less than 0.3 bpnzac for images with JPEG quality 75 and its detection probabilities at these payloads are similar to the one for JUNIWARD which is a more time-consuming algorithm. For JPEG quality of 95, G-UERD has a statistically similar performance comparing to UERD. For JUNIWARD, the proposed Gaussian version (G-JUNIWARD) performs better than or similar to the original JUNIWARD algorithm, and the improvement is statistically significant for JPEG quality of 95 and payload greater than 0.3 bpnzac. The proposed G-JHILL outperforms all the mentioned algorithms in all the payloads and JPEG quality factors (or performs similarly to the most secure one). For images with JPEG quality factor of 75, the gap between the performance of G-JHILL and the most secure algorithm amongst the other methods (G-JUNIWARD for $p \leq 0.3$ and UERD for $p > 0.3$) is statistically significant at 0.1 and 0.2 bpnzac. For images with JPEG quality factor of 95, the gap is significant at 0.2, 0.3, 0.4, 0.5, 0.75, and 1 bpnzac.

In addition to running experiments using GFR features, we utilize DCTR features as well, and the results are reported in Table VII. Similar behaviors as the ones seen using GFR can be seen there; however, the performance gaps are greater compared to Table VI.

We believe that the proposed quantized Gaussian embedding model improves performance due to the fact that it embeds more bits in low cost or high variance DCT coefficients and less bits in high cost or low variance ones comparing to the Gibbs measure used by all the spatial and JPEG steganography methods.

4.5.3 Whitening

In this section, we conduct experiments on G-JHILL algorithm to check the empirical results of applying whitening explained in Sec. 4.2.1. For applying whitenning to all the blocks, there are two extra steps that are added to the algorithm explained in Fig. 4. First, instead of the residual variances computed in "if" clause in lines 1 through 7, we use variances of the whitened residuals using the eigen-decomposition. In other words, in each block, we first decompose each block residual covariance matrix by eigen-decomposition to $\mathrm{U}_b\Gamma_b\mathrm{U}_b^T$, where U_b is the orthogonal 64×64 matrix of eigenvectors and Γ_b is the diagonal matrix of eigenvalues. Then the diagonal elements of Γ_b are used instead of residual variances, i.e. σ_{bij}^2. The second extra step is that in each Newton-Raphson iteration for solving Eq. (4.27) after computing $B_b^* = [\beta_{bij}^*]_{8\times 8}$, the hidden message elements are transformed back by $\mathrm{U}_b \cdot \mathrm{vec}(B_b^*)$ where vec is vectorization function. This process increases the time complexity of the embedding method, but it increases the performance. In Table VIII, the performances of G-JHILL algorithm is reported for both cases of using and not using whitening. It can be seen that there is no statistically significant change in the detection error for payloads up to 0.5 bpnzac. However, in 0.75 an 1 bpnzac,

the G-JHILL version that employs whitening performs significantly better. In the next section, we discuss the amount of increase in computation time for using whitening.

TABLE VIII: Detection error of steganalysis using GFR features in various payloads (p), and different JPEG quality factors (Q.F.) for G-JHILL with and without whitening (Wh.).

Q.F.	Wh.	p=.05	0.1	0.2	0.3	0.4	0.5	0.75	1
75	No	.4650	.4134	.2986	.1893	.1139	.0631	.0131	.0048
	Yes	.4639	.4145	.2991	.1906	.1156	.0670	.0152	.0081
95	No	.4943	.4794	.4437	.3945	.3354	.2727	.1336	.0439
	Yes	.4949	.4786	.4443	.3918	.3384	.2749	.1460	.0642

4.5.4 Computational Time

In this section, we compare the computation time needed for all of the steganography algorithms studied in this paper. The computation times are reported in seconds per image in Table IX for two JPEG quality factors, i.e. 75 and 95, and two payloads, i.e. 0.1 and 0.2 bpnzac. It is observed that the proposed Gaussian embedding versions of UERD and JUNIWARD are 2 to 5 times slower than the original algorithms, which is still reasonable given their higher performance. G-JHILL (Wh.) is the G-JHILL algorithm with whitening which is 2 to 3 times slower than G-JHILL. It can be seen that higher

TABLE IX: Average computational time in seconds for embedding a coded hidden message with size of p bpnzac in a JPEG image with quality factor Q.F..

Q.F.	p	UERD	G-UERD	JUNI	G-JUNI	G-JHIL	G-JHILL (Wh.)
75	0.1	.2978	1.141	2.326	4.863	4.373	12.03
	0.2	.2705	1.626	2.669	5.086	4.877	12.32
95	0.1	.2587	1.344	2.451	5.003	4.598	12.16
	0.2	.3054	1.472	2.631	5.200	4.866	12.25

payload increases the embedding time but the JPEG quality factor does not affect the computation time significantly.

4.5.5 Pool Steganalysis Detection Error

In this section, we conduct experiments regarding Sec. 4.4 and Theorem 3, where we have shown that instead of running cumbersome pool steganalysis experiments, one can estimate the detection error for pool sizes greater than 1 based on Eq. 4.30 and empirically computed detection error for pool size equal to 1. We use various pool sizes, i.e. $l \in \{1,3,\dots,99\}$, for both empirical and estimated results. The pooling strategy here is using the summation of detection statistics of all images in a pool. This strategy is shown to be optimal in Theorem 3 in case of embedding the same payload in all images or using the state-of-the-art batch steganographer (Sharifzadeh et al., 2019b).

Results for using the G-UERD embedding algorithm are shown in Fig. 5. In each plot, the pink lines are the empirical results, and their error bars show the detectors error standard deviation. The solid blue lines with "$*$" markers are the results computed by the proposed estimation. The results for JPEG quality

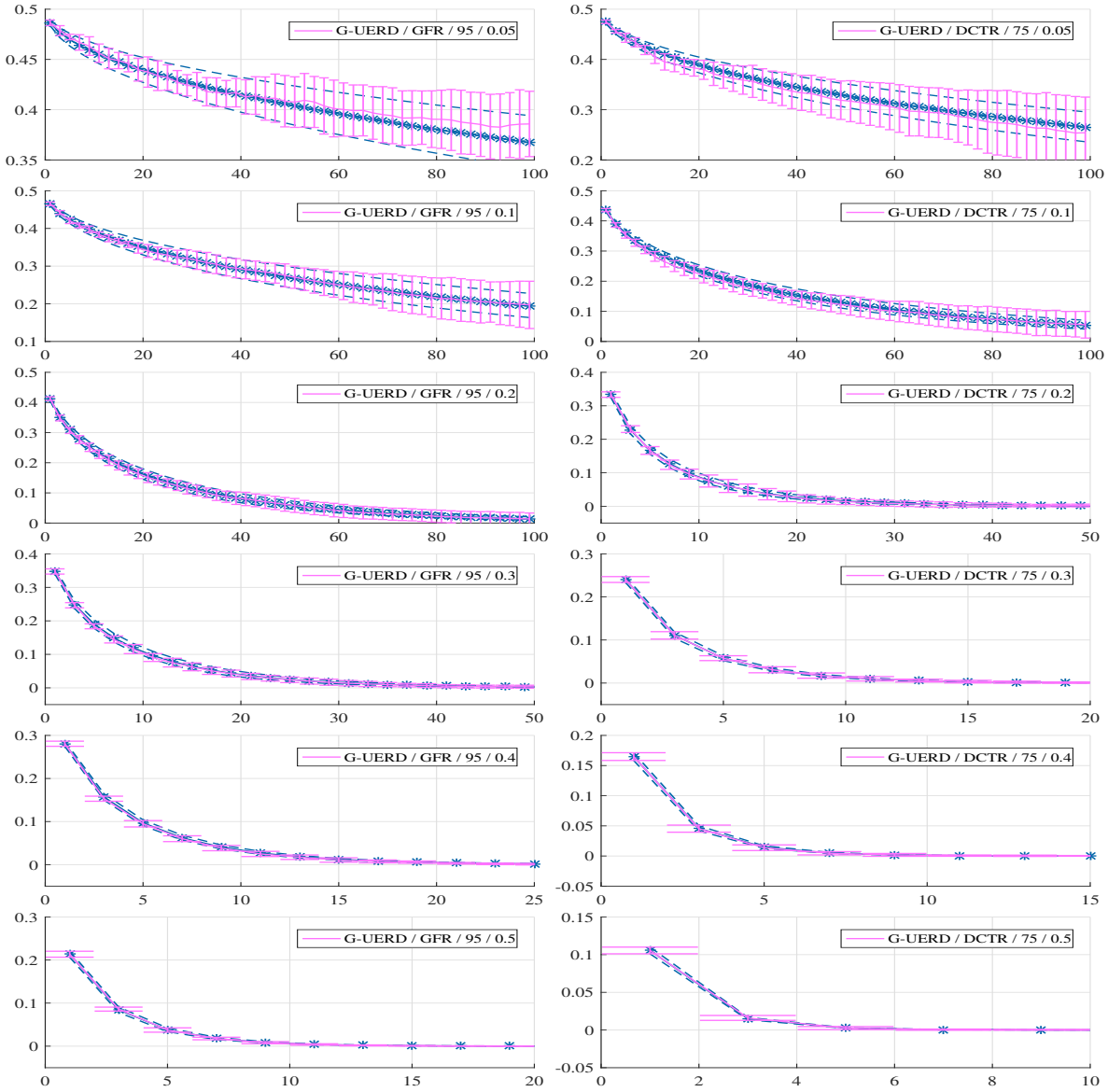

Figure 5: Empirical pool steganalysis detection error (pink lines), and the estimated one and its standard deviation calculated by Eq. 4.30 and Eq. 4.31 respectively (solid blue lines with "$*$" markers and dashed blue lines respectively), versus pool size for G-UERD algorithm, two steganalysis features (GFR and DCTR), different JPEG quality factors (75 and 95) and payloads $(0.05, 0.1, 0.2, 0.3, 0.4, 0.5)$. Plot legends are read as "Embedding method / Steganalysis feature / JPEG quality / Payload".

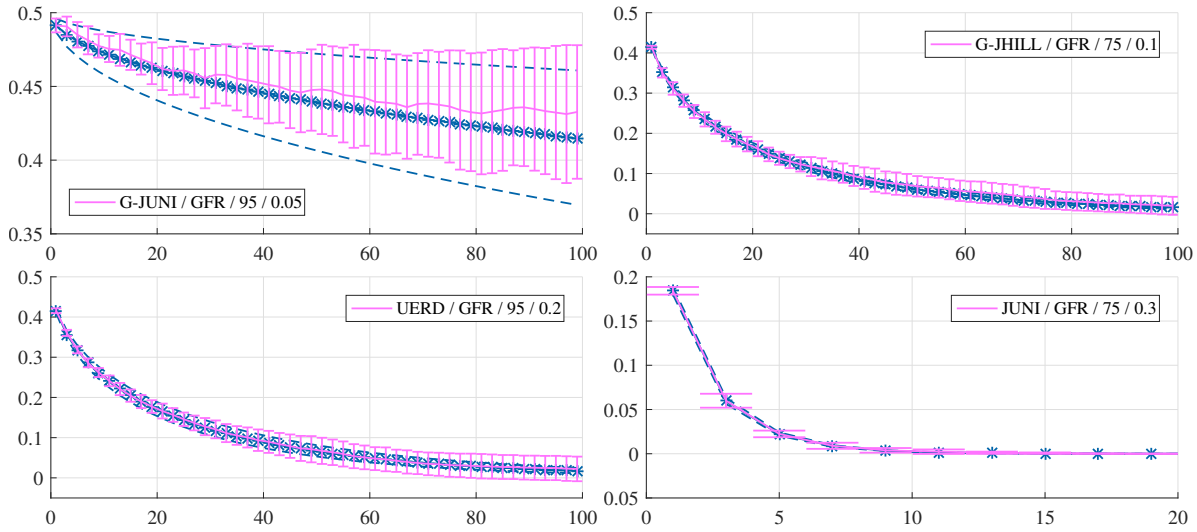

Figure 6: Empirical pool steganalysis detection error (pink lines), and the estimated one and its standard deviation calculated by Eq. 4.30 and Eq. 4.31 respectively (solid blue lines with "$*$" markers and dashed blue lines respectively), versus pool size for various algorithm, GFR as steganalysis feature, different JPEG quality factors (75 and 95) and payloads $(0.05, 0.1, 0.2, 0.3)$. Plot legends are read as "Embedding method / Steganalysis feature / JPEG quality / Payload".

factor of 95, using the GFR feature, and payloads of $0.05, 0.1, 0.2, 0.3, 0.4, 0.5$ bpnzac are provided in the left column in which it can be seen that our estimation is precise. To show that the proposed estimation is precise for other quality factors and other steganalysis features as well, we show similar plots for JPEG quality factor of 75, using DCTR feature, and payloads of $0.05, 0.1, 0.2, 0.3, 0.4, 0.5$ bpnzac on the right column. Based on Fig. 5, the proposed estimation is valid in all the payloads, JPEG quality factors, and steganalysis features for the G-UERD algorithm. To show that it is valid for all embedding methods regardless of them using the proposed Gaussian embedding model or not, we provide similar plots for G-JUNIWARD, G-JHILL, UERD, and JUNIWARD in Fig. 6. In this Figure, we have tried to cover all experimented embedding algorithms, JPEG quality factors with different payloads by the fewest possible number of plots due to space and computation limitations.

4.5.6 Pool Steganalysis Detection Error Variance

In this section, we discuss the behavior of the variance of the pool steganalysis detector. In Sec. 4.4 and Theorem 3, we have shown that although pooling improves detection error, it increase the variance of the detector for some payloads depending on the value of single image steganalysis detection error. In other words, according to Theorem 3, the variance of the detection error is an increasing function of pool size for pool sizes smaller than l_0, defined in Eq. (4.32), and it is a decreasing function for greater pool sizes.

To examine this finding, in all the plots in Fig. 5 and Fig. 6, in addition to the empirical and estimated pool steganalysis, results shown by pink error bars and solid blue lines with "$*$" markers respectively, we show the estimated standard deviation shown in Eq. (4.31) with dashed blue lines. In other words, in all the mentioned plots, the upper and the lower dashed blue lines are $\hat{\mathrm{P}}_{\mathrm{E}}(l) + \hat{\sigma}_l$

and $\hat{\mathrm{P}}_{\mathrm{E}}(l) - \hat{\sigma}_l$ respectively. It can be observed that the pink error bar sizes have similar behaviors as the distances between dashed blue lines. In other words, as the pool size increases, when dashed blue lines are diverging, the error bars sizes increase, and when dashed blue lines are converging, the error bars sizes decrease. The turning point of the explained behavior depends on the value of $\mathrm{P}_{\mathrm{E}}(1)$ and it decreases as $\mathrm{P}_{\mathrm{E}}(1)$ for empirical results. This is similar to the behavior of the estimated turning point l_0 shown in Eq. (4.32) which validates Theorem 3.

Here, we go through a few examples from the plots. In the left column of Fig. 5, in the top plot where $l_0 \approx 1035$, it can be observed that the size of the error bars of the pink line is increasing until $l = 99$. For the second plot from the top, where $l_0 \approx 139$, the error bars expand as well until $l = 99$. In contrast to the last two examples, in the third plot from the top, where $l_0 \approx 20.5$, the error bars are becoming larger until around $l = 41$, and then they start growing smaller in size. Similarly, for the fourth plot from the top where $l_0 \approx 6.6$, the size of the pink error bar is increasing as l increases until around $l = 9$ where it starts to decrease. For the second plot from the bottom in the left column of Fig. 5, where $l_0 \approx 2.9$, error bars start to shrink after approximately $l = 5$. And for the last plot where $l_0 \approx 1.5$, the error bar size is a decreasing function of l.

As a result of the mentioned behavior, which we also mathematically proved in Theorem 3, pool steganalysis suffers from instability, i.e., high variance, for small payloads when single image steganalysis detection error is near 0.5. The instability is a serious disadvantage for pool steganalysis, especially in low payloads, and large pool sizes as the standard deviation can grow from a small number such as 0.004 in pool size equal to 1 to a huge number such as 0.04 in pool size equal to 99.

CHAPTER 5

CONCLUSION

5.1 Summary of contributions

In the third chapter, a statistical framework is developed for raw image steganography problem in which the cover and the stego messages are modeled by independent Gaussian random variables. Subsequently, a novel Gaussian embedding model is proposed by simultaneously minimizing the detection error of three optimal hypothesis testing detectors. The proposed Gaussian embedding model can work with both pixel embedding costs and residual variances, which makes it a universal embedding technique applicable to all the state-of-the-art image steganography methods, and it improves their security significantly. Additionally, the closed-form detection error as a function of payload is derived within the adopted model for image steganography, and it is extended to batch steganography as well. The availability of the closed-form detection error allowed us to investigate the effect of batch size on the security of batch steganography. As a result, a new batching strategy, *AdaBIM*, is introduced, which is shown to outperform the state-of-the-art both mathematically and empirically.

In the fourth chapter, we extend the statistical framework proposed in the third chapter to JPEG steganography in which we employ a Gaussian model for the cover coefficients and also the hidden message elements. Based on that, we propose a quantized Gaussian embedding model that is able to work with any embedding cost or residual variance computed in spatial or DCT domain. We show that

using this embedding model improves the performance of the existing JPEG steganography algorithms in most of the payloads, and also achieves superior performance for all the payloads using cost calculated by HILL. Subsequently, the proposed statistical model allows us to derive the closed-form expression of an optimal omniscience single image steganalyzer error and extended it to pool steganalysis. We use the closed-form expression of pool steganalysis error to approximate the empirical results for pool steganalysis accurately. The main benefit of this approximation is that it is accurate if the pooling method is optimal regardless of payload, steganalysis feature, and embedding method and domain. In addition to approximating the error, we correctly predict the error variance empirical behavior with respect to pool size, and therefore, reveal a deficiency of pool steganalysis.

5.2 Future directions

In future, we plan to investigate skewed statistical models such as generalized Gaussian distribution for cover and stego image pixels. This may lead to asymmetric embedding steganography method that can embed in saturated pixels and also outliers in smooth regions. Additionally, the derived closed-form expression of the detection error could be utilized to solve the pooled steganalysis problem as well as the batch steganography problem directly without utilizing the image merging sender. Another possible direction for future is investigating side-informed steganography as an immediate extension of the fourth chapter. In addition, the derived closed-form expressions can be used for calculation of embedding costs and residual variances. Furthermore, the proposed statistical model can be employed for video steganography if frame to frame dependencies are taken into account in computation of the residual variances.

Appendices

Appendix A: Asymptotic Sum of Gamma Random Variables

Suppose $X_1, \cdots, X_n$ are all independently distributed by Gamma with shape k, but with scaling parameters $\theta_1, \cdots \theta_n$ respectively. If all θ's are bounded, the probability distribution of the following summation, where $a_1, \ldots, a_n$ are some constants, converges to normal distribution as shown below.

$$\sum_{i=1}^{n}(X_i + a_i) \xrightarrow{\mathrm{d}} \mathcal{N}\Big(\sum_{i=1}^{n}(k\theta_i + a_i), k\sum_{i=1}^{n}\theta_i^2\Big) \tag{A.1}$$

Proof of Theorem 1. Let Y be the sum of all X_is and Z be the normalized Y, i.e.

$$Z = \frac{Y - \mathrm{E}[Y]}{\mathrm{Var}[Y]} = \frac{\sum_{\ell=1}^{n}(X_\ell - k\theta_\ell)}{\sqrt{k\sum_{i=1}^{n}\theta_i^2}} \tag{A.2}$$

Based on (Mathai, 1982), probability distribution of Z converges to standard normal distribution, $\mathcal{N}(0,1)$, when $n \to \infty$ if the following conditions are met.

1. $0 < k\sum_{i=1}^{n}(\frac{\theta_i}{\sqrt{n}})^2 < \infty$
2. $\lim_{n\to\infty} k\sum_{i=1}^{n}(\frac{\theta_i}{\sqrt{n}})^r = 0 \quad \text{for } r \geq 3$

These conditions are met as long the θ's are bounded. To show this, suppose that $0 < \theta_{\min} \leq \theta_i \leq \theta_{\max} < \infty$ for all i's. Then it can be easily shown that

$$kn\frac{\theta_{min}^r}{\sqrt{n^r}} < k\sum_{i=1}^{n}(\frac{\theta_i}{\sqrt{n}})^r < kn\frac{\theta_{max}^r}{\sqrt{n^r}} \tag{A.3}$$

Appendix A (Continued)

If $n \to \infty$, and $r = 2$, the lower and upper bounds are $k\theta_{min}^2$ and $k\theta_{max}^2$ respectively and they are both bounded. If $n \to \infty$, and $r \geq 3$, they both tend to zero.

Therefore, for large enough n, probability distribution of Y can be approximated with normal distribution, i.e.

$$Y \sim \mathcal{N}\Big(k\sum_{i=1}^{n}\theta_i,\ k\sum_{i=1}^{n}\theta_i^2\Big) \tag{A.4}$$

Now we are one step away from the complete proof of the theorem. Suppose $Y' = \sum_{i=1}^{n}(X_i + a_i)$ which is just a constant, $\sum_{i=1}^{n} a_i$, plus Y. As a result

$$Y' \sim \mathcal{N}\Big(\sum_{i=1}^{n}(k\theta_i + a_i),\ k\sum_{i=1}^{n}\theta_i^2\Big) \tag{A.5}$$

which proves the theorem. ■

Appendix B: Effect of Batch Size on Security

In this section, the following lemma is proven first. Then, the result is extended to compare the detection error in case of using different batch sizes and prove Theorem 2.

Lemma. *Given any $x, \alpha \geq 0$, the following statements for normal cumulative distribution function, ϕ, are true:*

(i) $\frac{1}{2}\phi(-x)+\frac{1}{2}\phi(-\alpha x) \leq \phi(-\sqrt{\alpha}x) \qquad x, \alpha x \ll 1$

(ii) $\frac{1}{2}\phi(-x)+\frac{1}{2}\phi(-\alpha x) \geq \phi(-\sqrt{\alpha}x) \qquad x, \alpha x \to \infty$

Proof (i). The first part of the lemma states that when x tends to zero, the following inequality holds;

$$\frac{1}{2}\phi(-x)+\frac{1}{2}\phi(-\alpha x) \leq \phi(-\sqrt{\alpha}x) \qquad x, \alpha x \ll 1 \tag{B.1}$$

To prove (B.1), we approximate ϕ with the first two terms of its Taylor series expansion given by

$$\phi(z) = \frac{1}{2} + \frac{1}{\sqrt{2\pi}} \sum_{n=0}^{\infty} \frac{(-1)^n z^{2n+1}}{(2n+1)2^n n!} \xrightarrow{z \ll 1} \frac{1}{2} + \frac{z}{\sqrt{2\pi}} \tag{B.2}$$

Applying approximation shown in (B.2) to (B.1) results in

$$\frac{1}{2} + \frac{(-x)}{2\sqrt{2\pi}} + \frac{(-\alpha x)}{2\sqrt{2\pi}} \leq \frac{1}{2} + \frac{(-\sqrt{\alpha}x)}{\sqrt{2\pi}} \tag{B.3}$$

Appendix B (Continued)

Since x is positive, this in turn means

$$-\alpha - 1 \leq -2\sqrt{\alpha} \tag{B.4}$$

which is true due to the fact that $0 \leq (\sqrt{\alpha} - 1)^2$ ■

Proof (ii). The second part of the lemma states that when x approaches infinity, the following inequality holds;

$$\frac{1}{2}\phi(-x) + \frac{1}{2}\phi(-\alpha x) \geq \phi(-\sqrt{\alpha}x) \qquad x, \alpha x \to \infty \tag{B.5}$$

To prove (B.5), we approximate ϕ with its asymptotic expansion. To derive this expansion, the asymptotic expansion of the error function, erf, is utilized which is given by

$$\text{erf}(z) = \text{-1+}\frac{\text{e}^{-z^2}}{\sqrt{\pi}} \sum_{n=0}^{\infty} \frac{\text{(-1)}^n (2n\text{-1})!!}{2^n \ \text{(-}z\text{)}^{n+1}} \quad \text{as } z \to -\infty \tag{B.6}$$

where !! is the double factorial, i.e. $n!! = n \cdot (n-2) \cdots 1$. Given that $\text{erf}(z) = 2\phi(\sqrt{2}z) - 1$ and using the first two terms of the asymptotic series in (B.6), it can be shown that

$$\phi(-z) \xrightarrow{z \to \infty} \frac{\text{e}^{\frac{-z^2}{2}}}{\sqrt{2\pi}z} \tag{B.7}$$

By applying (B.7), to (B.5), we get the following inequality,

$$\frac{\text{e}^{\frac{-x^2}{2}}}{2\sqrt{2\pi}x} + \frac{\text{e}^{\frac{-(\alpha x)^2}{2}}}{2\sqrt{2\pi}\alpha x} \geq \frac{\text{e}^{\frac{-(\sqrt{\alpha}x)^2}{2}}}{\sqrt{2\pi\alpha}x} \tag{B.8}$$

Appendix B (Continued)

Since x and αx are positive as they approach infinity, inequality (B.8) can be simplified as

$$\alpha^{\frac{1}{2}} \mathrm{e}^{\frac{-(1-\alpha)x^2}{2}} + \alpha^{\frac{-1}{2}} \mathrm{e}^{\frac{-(\alpha^2-\alpha)x^2}{2}} \geq 2 \tag{B.9}$$

This is true for every positive α, since for every α other than one, one of the terms on the left hand side of this equation goes to infinity as x approaches infinity, and for α equal to one, the left hand side is exactly two and equality happens. ■

Proof of Theorem 2. This theorem compares the error of detection for batch size of M and $2M$ for embedding p nats per pixel in a database of N images. Suppose M, and N are powers of two and $2M \leq N$. Without loss of generality, assume that the l^{th} batch includes these images: $(l-1)B, \ldots, lB-1$, where B is the batch size. As a result, the l^{th} batch when $B = 2M$, contains images of batches $2l-1$ and $2l$ of the case when $B = M$. Based on (3.33), the average detection error for these images $((l-1)2M, \ldots, 2lM-1)$, when $B = M$, is

$$\frac{1}{2}\phi\left(-\sqrt{n\lambda_{2l-1}^{(M)}(p)/32}\right) + \frac{1}{2}\phi\left(-\sqrt{n\lambda_{2l}^{(M)}(p)/32}\right) \tag{B.10}$$

and when $B = 2M$, is

$$\phi\Big(-\sqrt{n\lambda_{l}^{(2M)}(p)/32}\Big) \tag{B.11}$$

Appendix B (Continued)

where $\lambda_l^{(M)}(p)$ is the Lagrangian multiplier for the l^{th} batch when the batch size and the payload are M, and p respectively. For using the lemma to compare these two average detection errors, (B.10) and (B.11), let us define x and α as

$$x = \sqrt{\frac{n\lambda_{2l-1}^{(M)}(p)}{32}} \tag{B.12}$$

$$\alpha = \frac{\sqrt{\frac{n\lambda_{2l}^{(M)}(p)}{32}}}{\sqrt{\frac{n\lambda_{2l-1}^{(M)}(p)}{32}}} = \frac{\sqrt[M]{\prod_{j=(2l-1)M-1}^{(2l-1)M} \sqrt[n]{\prod_{i=1}^{n} \sigma_{ij}^2}}}{\sqrt[M]{\prod_{j=(2l-1)M}^{2lM-1} \sqrt[n]{\prod_{i=1}^{n} \sigma_{ij}^2}}} \tag{B.13}$$

where the simplification is done based on (3.35). Note that, α, defined in (B.13), is constant for all the payloads, p, regardless of the value of x and l. Employing (3.35), it can be shown that

$$\lambda_l^{(2M)}(p) = \sqrt{\lambda_{2l}^{(M)}(p)\lambda_{2l-1}^{(M)}(p)} \tag{B.14}$$

which results in

$$\sqrt{n\lambda_l^{(2M)}(p)/32} = \sqrt{\alpha}x \tag{B.15}$$

Based on the variable definitions in (B.12), (B.13), and (B.15), and the first part of the lemma, (B.1), it can be shown that if $x, \alpha x \ll 1$, (B.10) is less or equal than (B.11). Therefore, the summation of (B.10) over all batches, $l \in \{1, \dots \frac{N}{M}\}$, is less or equal than the summation of (B.11). In addition, as it was shown in Sec. 3.4, for payloads much smaller than one, Lagrangian multiplier λ for all batches

Appendix B (Continued)

and consequently x and αx are much smaller than one. Therefore, based on (3.37), and the mentioned inequality, it is concluded that

$$\mathrm{P_E}(M,N,p) < \mathrm{P_E}(2M,N,p) \qquad p \ll 1 \tag{B.16}$$

Following similar steps but using the second part of the lemma, (B.5), it can be shown that if x and αx, shown in (B.12) and (B.13), approach infinity, the summation of (B.10) over all batches, $l \in \{1,\dots \frac{N}{M}\}$, is greater or equal than the summation of (B.11). In addition, as it was shown in Sec. 3.4, for payloads approaching infinity, Lagrangian multiplier λ for all batches and consequently x and αx approach infinity. Thus, the following inequality holds

$$\mathrm{P_E}(M,N,p) > \mathrm{P_E}(2M,N,p) \qquad p \to \infty \tag{B.17}$$

which proves the second part of Theorem 2. ■

Appendix C: Statistical Model for Pool Steganalysis Detector's Error and Variance

In this section, we discuss the pool steganalysis problem for steganography in raw image or any linear transformation of image. The discussion is based on the Gaussian statistical model which is valid for any linear transformation of image. The model for spatial domain steganography is presented in (Sharifzadeh et al., 2019a) and for JPEG steganography is shown in Sec. 4.2.1 and 4.2.2. Within the adopted statistical model, the detection error of an optimal single image steganalysis is given by

$$\phi\Big(-\sqrt{\frac{n}{32}\lambda(p)}\Big) \tag{C.1}$$

where $\lambda(p)$ is the Lagrangian multiplier for relative payload p. Now, we discuss the case in which the detector knows that l images are sent by the same source. We prove that in such cases, an optimal pool steganalyzer should examine the images together. To show this, we compare the detection error for both cases of inspecting l images together and separately. Inspecting images together results in a similar detection error with summation of Lagrangian multipliers for all of the l images because the logarithm of the likelihood ratio is equal to summation of logarithm of likelihood ratios for l images. Given that the steganographer is embedding in each image separately, the Lagrangian multiplier values are different for every image, i.e. $\lambda^{(a)}(p)$ is the Lagrangian multiplier for the a^{th} image. The detection error for such a detector is as follows

$$\phi\Big(-\sqrt{\frac{n}{32}\sum_{a=1}^{l}\lambda^{(a)}(p)}\Big) \tag{C.2}$$

Appendix C (Continued)

This shows that the optimal detector developed here uses pooling strategy of summing detection statistics of all the images in the pool.

If l images, known to have the same source, are inspected separately, the average detection error is given by

$$\frac{1}{l}\sum_{a=1}^{l}\phi\Big(-\sqrt{\frac{n\lambda^{(a)}(p)}{32}}\Big) \tag{C.3}$$

Eq. (C.3) is greater or equal than the formula below based on Jensen's inequality and the fact that $\phi(-\sqrt{x})$ is a convex function of x if $x>0$.

$$\phi\Big(-\sqrt{\frac{n}{32\,l}\sum_{a=1}^{l}\lambda^{(a)}(p)}\Big) \tag{C.4}$$

Eq. (C.4) is greater than detection error shown in Eq. (C.2) based on the fact that $\phi(-\sqrt{x})$ is a decreasing function of x if $x>0$. This proves that steganalyzer should inspect all the images from the same source together to achieve a lower detection error. However, this approach will result in a detector with higher variance which is covered later in this section. Now that we have derived the optimal pool steganalysis strategy and its detection error, we show that instead of running time consuming pool steganalysis experiments, one can utilizes Eq. (C.2) to approximate the results.

Assume that a database of N images (JPEG or raw) is used for embedding a relative payload of p nats (p nats per non zero AC DCT coefficients for JPEG images and p nats per pixel for raw images)

Appendix C (Continued)

using the proposed Gaussian embedding model. The average detection error of an optimal single image steganalyzer for the whole database is given by

$$\hat{\mathrm{P}}_{\mathrm{E}}(1) = \frac{1}{N}\sum_{a=1}^{N}\phi\Big(-\sqrt{\frac{n}{32}\lambda^{(a)}(p)}\Big) \tag{C.5}$$

which can be approximated as shown below by assuming that all $\lambda^{(a)}(p)$ values are the same and equal to a value $\lambda(p)$

$$\hat{\mathrm{P}}_{\mathrm{E}}(1) \approx \phi\Big(-\sqrt{\frac{n}{32}\lambda(p)}\Big) \tag{C.6}$$

The mentioned assumption is true for the state-of-the-art batch steganography method which embeds in each image according to its steganographic capacity and uses an image merging sender which results in equal values of λ (Sharifzadeh et al., 2019b; Cogranne et al., 2017). The assumption is an approximation for a steganographer that embeds the same payload in all the images but it still results in a precise estimation as shown in Sec. 4.5.5.

If the images are received in pools of l images, the detection error of an optimal pool steganalyzer is given by

$$\hat{\mathrm{P}}_{\mathrm{E}}(l) = \frac{l}{N}\sum_{t=0}^{N/l-1}\phi\Big(-\sqrt{\frac{n}{32}\sum_{a=t\times l+1}^{(t+1)\times l}\lambda^{(a)}(p)}\Big) \tag{C.7}$$

which can also be approximated as shown below using the same assumption of equal Lagrangian multipliers

$$\hat{\mathrm{P}}_{\mathrm{E}}(l) \approx \phi\Big(-\sqrt{\frac{n}{32}l\lambda(p)}\Big) \tag{C.8}$$

Appendix C (Continued)

Therefore, based on Eq. (C.6) and Eq. (C.8), an approximation of $\hat{\mathrm{P}}_{\mathrm{E}}(l)$ based on the value of $\hat{\mathrm{P}}_{\mathrm{E}}(1)$ is given by

$$\hat{\mathrm{P}}_{\mathrm{E}}(l) \approx \phi\left(\phi^{-1}\left(\hat{\mathrm{P}}_{\mathrm{E}}(1)\right)\sqrt{l}\right) \tag{C.9}$$

where ϕ^{-1} is the inverse function of cumulative standard normal distribution, ϕ.

In the rest of this section, we discuss the error of this approximation if $\hat{\mathrm{P}}_{\mathrm{E}}(1)$ has an error with standard deviation of $\hat{\sigma}_1$. We show the standard deviation of error of $\hat{\mathrm{P}}_{\mathrm{E}}(l)$ with $\hat{\sigma}_l$. Suppose that all the errors are small, i.e. $\forall l\ \hat{\sigma}_l \ll 1$. Therefore, our approximation shown in Eq. (C.9) has error with standard deviation, i.e. $\hat{\sigma}_l$, given by

$$2\hat{\sigma}_l \approx \phi\left(\phi^{-1}\left(\hat{\mathrm{P}}_{\mathrm{E}}(1)+\sigma_1\right)\sqrt{l}\right) - \phi\left(\phi^{-1}\left(\hat{\mathrm{P}}_{\mathrm{E}}(1)-\sigma_1\right)\sqrt{l}\right) \tag{C.10}$$

This can be further simplified using the following Taylor series expansion

$$\phi\left(\phi^{-1}(x\pm\delta x)\sqrt{l}\right) \approx \phi\left(\phi^{-1}(x)\sqrt{l}\right) \pm \frac{\partial\phi\left(\phi^{-1}(x)\sqrt{l}\right)}{\partial x}\delta x \tag{C.11}$$

By plugging in this Taylor series in Eq. (C.10), our approximation error can be calculated as

$$\hat{\sigma}_l \approx \left.\frac{\partial\phi\left(\phi^{-1}(x)\sqrt{l}\right)}{\partial x}\right|_{x=\hat{\mathrm{P}}_{\mathrm{E}}(1)} \hat{\sigma}_1 = \gamma\hat{\sigma}_1 \tag{C.12}$$

$$\gamma \doteq \sqrt{l}\exp\left(-\frac{1}{2}\left(\phi^{-1}\left(\hat{\mathrm{P}}_{\mathrm{E}}(1)\right)\right)^2(l-1)\right) \tag{C.13}$$

Appendix C (Continued)

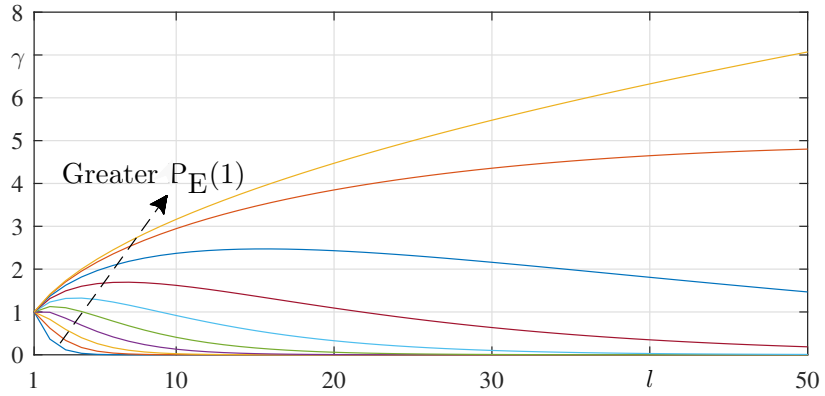

Figure 7: Pool steganalysis error variance behaviour shown by plotting variable γ, defined in Eq. (C.13), versus pool size, l, for different detection errors of single image steganalysis, $\hat{P}_E(1)$.

The variable γ's behavior with respect to l depends on $\hat{P}_E(1)$ value. In Fig. 7, γ is shown for different l and $\hat{P}_E(1)$, which shows that for a all $\hat{P}_E(1)$, $\gamma = 1$ when $l = 1$ and it has one global maximum. It can be seen that utilizing pool steganalysis results in greater variances for some pool sizes comparing to single image steganalysis for higher $\hat{P}_E(1)$, because γ is greater than 1. To find out exactly when this happens, we derive the derivation of γ with respect to l which is given by

$$\frac{\partial \gamma}{\partial l} = \frac{\gamma \cdot \left(1 - \left(\phi^{-1}\left(\hat{P}_E(1)\right)\right)^2 l\right)}{2l} \tag{C.14}$$

Appendix C (Continued)

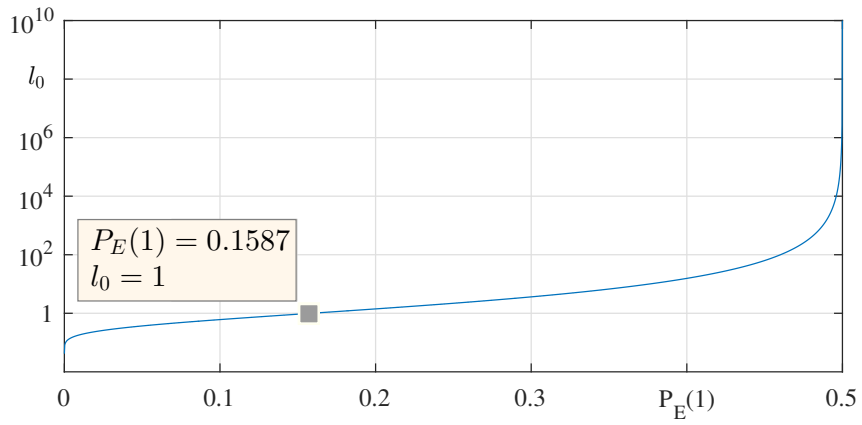

Figure 8: l_0 defined in Eq. (C.15) versus $\hat{\mathrm{P}}_{\mathrm{E}}(1)$.

Since l takes only natural numbers in practice, γ is a decreasing function of l if its derivation shown in Eq. (C.14) goes to zero for $l \leq 1$. The derivation of γ with respect to pool size, l, is zero if $l = l_0$ where l_0 is

$$l_0 = \left(\phi^{-1}\left(\hat{\mathrm{P}}_{\mathrm{E}}(1)\right)\right)^{-2} \tag{C.15}$$

$$l_0 \leq 1 \Rightarrow \hat{\mathrm{P}}_{\mathrm{E}}(1) \leq 0.1587 \tag{C.16}$$

Fig. 8 depicts l_0 vs $\hat{\mathrm{P}}_{\mathrm{E}}(1)$. Therefore, for any $\hat{\mathrm{P}}_{\mathrm{E}}(1) > 0.1587$, our approximation show that the variance, $\hat{\sigma}_l$, increases as l increases until $l = l_0$. Then, the variance of detection error decreases. The same behaviour is also observed in practice in Sec. 4.5.6 for empirical detection error which reassures the precision of the proposed approximation and mathematical model for pool steganalysis.

Appendix D: Copyright Permissions

In this appendix, the copyright permissions for the articles used in this thesis are provided. The list of the articles include IEEE Transactions on Information Forensics and Security 2019 (TIFS 2019) (Sharifzadeh et al., 2019b), and IEEE International Conference on Acoustics, Speech and Signal Processing 2019 (ICASSP 2019) (Sharifzadeh et al., 2019a).

Title: Quantized Gaussian Embedding Steganography

Conference Proceedings: ICASSP 2019 - 2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)

Author: Mehdi Sharifzadeh

Publisher: IEEE

Date: May 2019

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

Title: Adaptive Batch Size Image Merging Steganography and Quantized Gaussian Image Steganography

Author: Mehdi Sharifzadeh

Publication: Information Forensics and Security, IEEE Transactions on

Publisher: IEEE

Date: Dec 31, 1969

Thesis / Dissertation Reuse

The IEEE does not require individuals working on a thesis to obtain a formal reuse license, however, you may print out this statement to be used as a permission grant:

Requirements to be followed when using any portion (e.g., figure, graph, table, or textual material) of an IEEE copyrighted paper in a thesis:

1) In the case of textual material (e.g., using short quotes or referring to the work within these papers) users must give full credit to the original source (author, paper, publication) followed by the IEEE copyright line © 2011 IEEE.
2) In the case of illustrations or tabular material, we require that the copyright line © [Year of original publication] IEEE appear prominently with each reprinted figure and/or table.
3) If a substantial portion of the original paper is to be used, and if you are not the senior author, also obtain the senior author's approval.

Requirements to be followed when using an entire IEEE copyrighted paper in a thesis:

1) The following IEEE copyright/ credit notice should be placed prominently in the references: © [year of original publication] IEEE. Reprinted, with permission, from [author names, paper title, IEEE publication title, and month/year of publication]
2) Only the accepted version of an IEEE copyrighted paper can be used when posting the paper or your thesis on-line.
3) In placing the thesis on the author's university website, please display the following message in a prominent place on the website: In reference to IEEE copyrighted material which is used with permission in this thesis, the IEEE does not endorse any of [university/educational entity's name goes here]'s products or services. Internal or personal use of this material is permitted. If interested in reprinting/republishing IEEE copyrighted material for advertising or promotional purposes or for creating new collective works for resale or redistribution, please go to http://www.ieee.org/publications_standards/publications/rights/rights_link.html to learn how to obtain a License from RightsLink.

If applicable, University Microfilms and/or ProQuest Library, or the Archives of Canada may supply single copies of the dissertation.

BACK CLOSE WINDOW

CITED LITERATURE

[Anderson, 1996] Anderson, R.: Stretching the limits of steganography. In International Workshop on Information Hiding, pages 39–48. Springer, 1996.

[Bas et al. , 2011] Bas, P., Filler, T., and Pevnỳ, T.: break our steganographic system: The ins and outs of organizing boss. In International Workshop on Information Hiding, pages 59–70. Springer, 2011.

[Böhme, 2005] Böhme, R.: Assessment of steganalytic methods using multiple regression models. In International Workshop on Information Hiding, pages 278–295. Springer, 2005.

[Böhme, 2010] Böhme, R.: Advanced statistical steganalysis. Springer Science & Business Media, 2010.

[Cachin, 1998] Cachin, C.: An information-theoretic model for steganography. In International Workshop on Information Hiding, pages 306–318. Springer, 1998.

[Cheddad et al. , 2010] Cheddad, A., Condell, J., Curran, K., and Mc Kevitt, P.: Digital image steganography: Survey and analysis of current methods. Signal processing, 90(3):727–752, 2010.

[Cogranne, 2015] Cogranne, R.: A sequential method for online steganalysis. In 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pages 1–6. IEEE, 2015.

[Cogranne et al. , 2017] Cogranne, R., Sedighi, V., and Fridrich, J.: Practical strategies for content-adaptive batch steganography and pooled steganalysis. In International Conference on Acoustics, Speech and Signal Processing (ICASSP), pages 2122–2126. IEEE, 2017.

[Denemark et al. , 2014] Denemark, T., Fridrich, J., and Holub, V.: Further study on the security of s-uniward. In IS&T/SPIE Electronic Imaging, pages 902805–902805. International Society for Optics and Photonics, 2014.

[Denemark and Fridrich, 2017] Denemark, T. and Fridrich, J.: Model based steganography with precover. Electronic Imaging, 2017(7):56–66, 2017.

[Denemark et al. , 2014] Denemark, T., Sedighi, V., Holub, V., Cogranne, R., and Fridrich, J.: Selection-channel-aware rich model for steganalysis of digital images. In 2014 IEEE International Workshop on Information Forensics and Security (WIFS), pages 48–53. IEEE, 2014.

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

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

[Filler et al. , 2009] Filler, T., Ker, A. D., and Fridrich, J.: The square root law of steganographic capacity for markov covers. In Media Forensics and Security, volume 7254, page 725408. International Society for Optics and Photonics, 2009.

[Fridrich et al. , 2001] Fridrich, J., Goljan, M., and Du, R.: Reliable detection of lsb steganography in color and grayscale images. In Proceedings of the 2001 workshop on Multimedia and security: new challenges, pages 27–30. ACM, 2001.

[Fridrich and Kodovsky, 2012] Fridrich, J. and Kodovsky, J.: Rich models for steganalysis of digital images. IEEE Transactions on Information Forensics and Security, 7(3):868–882, 2012.

[Fridrich et al. , 2007] Fridrich, J., Pevnỳ, T., and Kodovskỳ, J.: Statistically undetectable jpeg steganography: dead ends challenges, and opportunities. In Proceedings of the 9th workshop on Multimedia & security, pages 3–14. ACM, 2007.

[Fridrich and Kodovskỳ, 2013] Fridrich, J. J. and Kodovskỳ, J.: Multivariate gaussian model for designing additive distortion for steganography. In ICASSP, pages 2949–2953, 2013.

[Guo et al. , 2014] Guo, L., Ni, J., and Shi, Y. Q.: Uniform embedding for efficient jpeg steganography. IEEE transactions on Information Forensics and Security, 9(5):814–825, 2014.

[Guo et al. , 2015] Guo, L., Ni, J., Su, W., Tang, C., and Shi, Y.-Q.: Using statistical image model for jpeg steganography: uniform embedding revisited. IEEE Transactions on Information Forensics and Security, 10(12):2669–2680, 2015.

[Holub et al. , 2014] Holub, V., Fridrich, J., and Denemark, T.: Universal distortion function for steganography in an arbitrary domain. EURASIP Journal on Information Security, 2014(1):1–13, 2014.

[Holub and Fridrich, 2012] Holub, V. and Fridrich, J.: Designing steganographic distortion using directional filters. In 2012 IEEE International workshop on information forensics and security (WIFS), pages 234–239. IEEE, 2012.

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

[Holub and Fridrich, 2014] Holub, V. and Fridrich, J.: Low-complexity features for jpeg steganalysis using undecimated dct. IEEE Transactions on Information Forensics and Security, 10(2):219–228, 2014.

[Holub and Fridrich, 2015] Holub, V. and Fridrich, J.: Phase-aware projection model for steganalysis of jpeg images. In Media Watermarking, Security, and Forensics 2015, volume 9409, page 94090T. International Society for Optics and Photonics, 2015.

[Johnson and Jajodia, 1998] Johnson, N. F. and Jajodia, S.: Exploring steganography: Seeing the unseen. Computer, 31(2), 1998.

[Joshi and Fischer, 1995] Joshi, R. L. and Fischer, T. R.: Comparison of generalized gaussian and laplacian modeling in dct image coding. IEEE Signal Processing Letters, 2(5):81–82, 1995.

[Ker, 2006] Ker, A. D.: Batch steganography and pooled steganalysis. In Information Hiding, volume 4437, pages 265–281. Springer, 2006.

[Ker, 2007] Ker, A. D.: Batch steganography and the threshold game. In Security, Steganography, and Watermarking of Multimedia Contents, page 650504, 2007.

[Ker et al. , 2013] Ker, A. D., Bas, P., Böhme, R., Cogranne, R., Craver, S., Filler, T., Fridrich, J., and Pevnỳ, T.: Moving steganography and steganalysis from the laboratory into the real world. In Proceedings of the first ACM workshop on Information hiding and multimedia security, pages 45–58. ACM, 2013.

[Ker and Pevny, 2012] Ker, A. D. and Pevny, T.: Batch steganography in the real world. In Proceedings of the on Multimedia and security, pages 1–10. ACM, 2012.

[Ker, 2004] Ker, A. D.: Improved detection of lsb steganography in grayscale images. In International workshop on information hiding, pages 97–115. Springer, 2004.

[Ker, 2007] Ker, A. D.: A capacity result for batch steganography. IEEE Signal Processing Letters, 14(8):525–528, 2007.

[Ker, 2008a] Ker, A. D.: Perturbation hiding and the batch steganography problem. In Information Hiding, pages 45–59. Springer, 2008.

[Ker, 2008b] Ker, A. D.: Steganographic strategies for a square distortion function. In Security, Forensics, Steganography, and Watermarking of Multimedia Contents X, volume 6819, page 681904. International Society for Optics and Photonics, 2008.

[Ker and Pevnỳ, 2011] Ker, A. D. and Pevnỳ, T.: A new paradigm for steganalysis via clustering. In Media Watermarking, Security, and Forensics III, volume 7880, page 78800U. International Society for Optics and Photonics, 2011.

[Ker et al. , 2008] Ker, A. D., Pevnỳ, T., Kodovskỳ, J., and Fridrich, J.: The square root law of steganographic capacity. In Proceedings of the 10th ACM workshop on Multimedia and security, pages 107–116. ACM, 2008.

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

[Kodovskỳ and Fridrich, 2012] Kodovskỳ, J. and Fridrich, J.: Steganalysis of jpeg images using rich models. In Media Watermarking, Security, and Forensics 2012, volume 8303, page 83030A. International Society for Optics and Photonics, 2012.

[Kodovsky et al. , 2011] Kodovsky, J., Fridrich, J., and Holub, V.: On dangers of overtraining steganography to incomplete cover model. In Proceedings of the thirteenth ACM multimedia workshop on Multimedia and security, pages 69–76. ACM, 2011.

[Lam and Goodman, 2000] Lam, E. Y. and Goodman, J. W.: A mathematical analysis of the dct coefficient distributions for images. IEEE transactions on image processing, 9(10):1661–1666, 2000.

[Lee, 2010] Lee, W. H.: Continuous and discrete properties of stochastic processes. Doctoral dissertation, University of Nottingham, 2010.

[Li et al. , 2014] Li, B., Wang, M., Huang, J., and Li, X.: A new cost function for spatial image steganography. In 2014 IEEE International Conference on Image Processing (ICIP), pages 4206–4210. IEEE, 2014.

[Li et al. , 2015] Li, B., Ng, T.-T., Li, X., Tan, S., and Huang, J.: Statistical model of jpeg noises and its application in quantization step estimation. IEEE Transactions on Image Processing, 24(5):1471–1484, 2015.

[Luo et al. , 2010] Luo, W., Huang, J., and Qiu, G.: Jpeg error analysis and its applications to digital image forensics. IEEE Transactions on Information Forensics and Security, 5(3):480–491, 2010.

[Mathai, 1982] Mathai, A.: Storage capacity of a dam with gamma type inputs. Annals of the Institute of Statistical Mathematics, 34(1):591–597, 1982.

[Moulin and O'Sullivan, 2003] Moulin, P. and O'Sullivan, J. A.: Information-theoretic analysis of information hiding. IEEE Transactions on information theory, 49(3):563–593, 2003.

[Moulin and Wang, 2007] Moulin, P. and Wang, Y.: Capacity and random-coding exponents for channel coding with side information. IEEE Transactions on Information Theory, 53(4):1326–1347, 2007.

[Pan et al. , 2016] Pan, Y., Ni, J., and Su, W.: Improved uniform embedding for efficient jpeg steganography. In International Conference on Cloud Computing and Security, pages 125–133. Springer, 2016.

[Pevny et al. , 2010a] Pevny, T., Bas, P., and Fridrich, J.: Steganalysis by subtractive pixel adjacency matrix. IEEE Transactions on Information Forensics and Security, 5(2):215–224, 2010.

[Pevnỳ et al. , 2010b] Pevnỳ, T., Filler, T., and Bas, P.: Using high-dimensional image models to perform highly undetectable steganography. In International Workshop on Information Hiding, pages 161–177. Springer, 2010.

[Pevnỳ and Nikolaev, 2015] Pevnỳ, T. and Nikolaev, I.: Optimizing pooling function for pooled steganalysis. In 2015 IEEE International Workshop on Information Forensics and Security (WIFS), pages 1–6. IEEE, 2015.

[Ryabko and Ryabko, 2009] Ryabko, B. Y. and Ryabko, D.: Asymptotically optimal perfect steganographic systems. Problems of Information Transmission, 45(2):184–190, 2009.

[Sadek et al. , 2015] Sadek, M. M., Khalifa, A. S., and Mostafa, M. G.: Video steganography: a comprehensive review. Multimedia tools and applications, 74(17):7063–7094, 2015.

[Sedighi et al. , 2016] Sedighi, V., Cogranne, R., and Fridrich, J.: Content-adaptive steganography by minimizing statistical detectability. IEEE Transactions on Information Forensics and Security, 11(2):221–234, 2016.

[Sedighi and Fridrich, 2016] Sedighi, V. and Fridrich, J.: Effect of saturated pixels on security of steganographic schemes for digital images. In Image Processing (ICIP), 2016 IEEE International Conference on, pages 2747–2751. IEEE, 2016.

[Sedighi et al. , 2015] Sedighi, V., Fridrich, J., and Cogranne, R.: Content-adaptive pentary steganography using the multivariate generalized gaussian cover model. In Media Watermarking, Security, and Forensics 2015, volume 9409, page 94090H. International Society for Optics and Photonics, 2015.

[Shannon, 1959] Shannon, C. E.: Coding theorems for a discrete source with a fidelity criterion. IRE Nat. Conv. Rec, 4(142-163):1, 1959.

[Sharifzadeh et al. , 2017] Sharifzadeh, M., Agarwal, C., Salarian, M., and Schonfeld, D.: A new parallel message-distribution technique for cost-based steganography. arXiv preprint arXiv:1705.08616, 2017.

[Sharifzadeh et al. , 2019a] Sharifzadeh, M., Aloraini, M., and Schonfeld, D.: Quantized gaussian embedding steganography. In International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.

[Sharifzadeh et al. , 2019b] Sharifzadeh, M., Aloraini, M., and Schonfeld, D.: Adaptive batch size image merging steganography and quantized gaussian image steganography. IEEE Transactions on Information Forensics and Security, 2019.

[Simmons, 1984] Simmons, G. J.: The prisoners problem and the subliminal channel. In Advances in Cryptology, pages 51–67. Springer, 1984.

[Song et al. , 2015] Song, X., Liu, F., Yang, C., Luo, X., and Zhang, Y.: Steganalysis of adaptive jpeg steganography using 2d gabor filters. In Proceedings of the 3rd ACM workshop on information hiding and multimedia security, pages 15–23. ACM, 2015.

[Upham, 1993] Upham, D.: Steganographic algorithm jsteg. Software available at http://zooid. org/˜ paul/crypto/jsteg, 1993.

[Wang and Moulin, 2008] Wang, Y. and Moulin, P.: Perfectly secure steganography: Capacity, error exponents, and code constructions. IEEE Transactions on Information Theory, 54(6):2706–2722, 2008.

[Westfeld, 2001] Westfeld, A.: F5a steganographic algorithm. In International workshop on information hiding, pages 289–302. Springer, 2001.

[Zakaria et al. , 2019] Zakaria, A., Chaumont, M., and Subsol, G.: Pooled steganalysis in jpeg: how to deal with the spreading strategy? arXiv preprint arXiv:1906.11525, 2019.

[Zhang et al. , 2016] Zhang, W., Zhang, Z., Zhang, L., Li, H., and Yu, N.: Decomposing joint distortion for adaptive steganography. IEEE Transactions on Circuits and Systems for Video Technology, 27(10):2274–2280, 2016.

[Zhao et al. , 2016] Zhao, Z., Guan, Q., Zhao, X., Yu, H., and Liu, C.: Embedding strategy for batch adaptive steganography. In International Workshop on Digital Watermarking, pages 494–505. Springer, 2016.

VITA

Name Mehdi Sharifzadeh

Education

Ph.D., Electrical and Computer Engineering
University of Illinois at Chicago, Chicago, Illinois, United States, 2019

M.Sc., Electrical and Computer Engineering
University of Illinois at Chicago, Chicago, Illinois, United States, 2018

B.Sc., Electrical Engineering
Sharif University of Technology, Tehran, Iran, 2012

Publications

- M.Sharifzadeh, M. Aloraini, and D. Schonfeld, Adaptive Batch Size Image Merging Steganography and Quantized Gaussian Image Steganography", IEEE Transactions on Information Forensics and Security (TIFS 2019).
- M. Sharifzadeh, M. Aloraini, and D. Schonfeld, Quantized Gaussian Embedding Steganography, IEEE International Conference on Acoustics, Speech and Signal (ICASSP 2019)
- M. Aloraini, L. Sha, M.Sharifzadeh, and D. Schonfeld, Dictionary Learning and Sparse Coding for Digital Image Forgery Detection, Electronic Imaging 2019.
- M. Aloraini, M. Sharifzadeh,C. Agarwal, and D. Schonfeld, Statistical Sequential Analysis for Object-based Video Forgery Detection, Electronic Imaging 2019.
- B Abbasi, M Sharifzadeh, E Noohi, S Parastegari, M efran, Grasp Taxonomy for Robot Assistants Inferred from Finger Pressure and Flexion, International Symposium on Medical Robotics (ISMR 2019)
- C Agarwal, M Sharifzadeh, D Schonfeld, CrossEncoders: A complex neural network compression framework, Electronic Imaging 2018.
- M. Sharifzadeh, C. Agarwal,M. Aloraini, and D. Schonfeld, Convolutional Neural Network Steganalysiss Application to Steganography, Visual Communications and Image Processing (VCIP 2017).
- M Salarian, M Sharifzadeh, R Ansari, Image Based Localization Based on Feature Scale Consistency in BOF Vector, IEEE International Symposium on Multimedia (ISM 2017).
- M Sharifzadeh, C Agarwal, M Salarian, D Schonfeld, A New Parallel Message-distribution Technique for Cost-based Steganography, preprint arXiv:1705.08616 (arXiv 2017).
- C Agarwal, M Sharifzhadeh, J Klobusicky, D Schonfeld, CrossNets: A New Approach to Complex Learning, preprint arXiv:1705.07404 (arXiv 2017).

- M. Sharifzadeh, and D. Schonfeld, Statistical and Information Theoretic Optimization and Performance Bounds of Video Steganography, Annual Allerton Conference on Communication, Control, and Computing 2015.

Internships

- **Software Engineering Intern** at **Google Brain**, Mountain View, California, Summer 2018.
- **Data Scientist Intern** at **Western Digital**, Milpitas, California, Summer 2017
- **Data Scientist Intern** at **SanDisk**, Milpitas, California, Summer 2016