TOWARDS ENHANCED STEGANOGRAPHIC METHOD FOR SECURE DATA TRANSMISSION OVER THE INTERNET

BY

AZZAT AHMED ALI AL-SADI

A Thesis Presented to the
DEANSHIP OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

COMPUTER NETWORKS

MAY 2012

This thesis, written by **Azzat Ahmed Al-Sadi** under the direction of this thesis advisor and approved by his thesis committee, has been presented to and accepted by the Dean of Graduate Studies, in partial fulfillment of the requirements for the degree of **MASTER OF SCIENCE IN COMPUTER NETWORKS**.

Thesis committee

Sayed 1/12/2012

Dr. El-Sayed M. El-Alfy

Thesis Advisor

2/12/2012

Dr. Wasfi G. Al-Khatib

Member

2/12/2012

Dr. Zubair A. Baig

Member

Dr. Basem M. Al-Madani

Department Chairman

Dr. Salam A. Zummo

Dean of Graduate Studies

9/12/12

Date

DEDICATION

To my father, whose journey through life has demonstrated the true meaning of hard work, courage, and perseverance; I dedicate this work to your valuable and imprinted words for higher academic achievements.

May your soul rest in eternal peace.

ACKNOWLEDGMENT

All praises to ALLAH (God), the Almighty, for providing me the blessings and ability to complete this work. I would also like to acknowledge the generous support and sponsorship of King Fahd University of Petroleum and Minerals, Hadhramout University of Science and Technology, and Hadhramout Establishment for Human Development. Thanks for granting me this outstanding opportunity to obtain my Master degree.

Moreover, I would like to express my sincere appreciation to my thesis committee, Dr. El-Sayed M. El-Alfy, Dr. Wasfi G. Al-Khatib, and Dr. Zubair A. Baig, for their remarkable feedback, encouragement and trust. Special thanks to Dr. El-Sayed M. El-Alfy for his close guidance and continuous support. I am particularly grateful and fortunate to have worked with him as I have gained valuable insights and acquired the necessary skills to publish into various venues and to proceed with the Ph.D.

Finally, my special thanks to all my family and friends who shared me throughout this experience and provided me with invaluable mental support.

TABLE OF CONTENTS

DEDICATION iii
ACKNOWLEDGMENT iv
LIST OF TABLES vii
LIST OF FIGURES viii
THESIS ABSTRACT ix
ملخّص الرسالة xi
CHAPTER 1 1
INTRODUCTION 1
    1.1. Motivation 3
    1.2. Problem Description 3
    1.3. Thesis Objectives 4
    1.4. Thesis Organization 5
CHAPTER 2 6
BACKGROUND AND LITRATURE REVIEW 6
    2.1. Background 6
        2.1.1. Information Hiding and Data Confidentiality 7
        2.1.2. Security Attacks 10
    2.2. Literature Review 12
CHAPTER 3 20
EXISTING STEGANOGRAPHIC SPATIAL METHODS 20
    3.1. PVD-Related Methods 20
    3.2. Traditional Edge Based Methods 33
CHAPTER 4 37
THE PROPOSED STEGANOGRAPHIC SYSTEM 37
    4.1. Chaotic Block Rotation 38
        4.1.1. Logistic-Chaotic Map 39
        4.1.2. PVD with Chaotic Block Rotation: Embedding 40

4.1.3. PVD with Chaotic Block Rotation: Recovery 42
4.2. Modulus Overlapping PVD Function 42
4.2.1. MOPVD with Chaotic Map: Embedding 43
4.2.2. MOPVD with Chaotic Map: Extraction 46
4.3. Fuzzy Edge Detection Function 46
4.3.1 E-MOPVD Embedding Example 47
4.3.2 E-MOPVD Extracting Procedure 50
CHAPTER 5 51
EVALUATION AND COMPARISONS 51
5.1. Evaluation Criteria 51
5.2. Test Images 54
5.3. Experimental Work 56
5.3.1. Capacity and Invisibility Evaluation 57
5.3.2. Security Evaluation 78
CHAPTER 6 97
CONCLUSIONS AND FUTURE WORK 97
References 105
VITAE 111

LIST OF TABLES

Table 1. Numbers of embedded bits using the edge detection with MOPVD 50
Table 2. Storing edge information in image pixels. 50
Table 3. Comparing the capacity for different methods..63
Table 4. Comparing the PSNR for different methods...67
Table 5. Comparing the MPSNR for different methods. 68
Table 6. Comparing the WPSNR for different methods. 69
Table 7. Comparing the SSIM for different methods...72
Table 8. Capacity of PVD and modified PVD using first range set..................................74
Table 9. Capacity of PVD and modified PVD with different chaotic parameters. 75
Table 10. Capacity of PVD and modified PVD using the second range set. 75
Table 11. Performance of E-CMOPVD as compared to E-MOPVD.................................76

LIST OF FIGURES

Figure 1. The principal components of a steganographic system. 2
Figure 2. The process of embedding secret data using PVD..23
Figure 3. Ranges and their boundaries. 25
Figure 4. An example of four pixel pairs. 29
Figure 5. Demonstration of the difference between PVD and OPVD methods. 30
Figure 6. The components of the proposed system. 38
Figure 7. The embedding procedure flowchart. 41
Figure 8. Block rotations. 41
Figure 9. The embedding process of the CMOPVD method. 45
Figure 10. The storing process of edge information. 47
Figure 11. Sample of the first dataset test images. 55
Figure 12. UCID dataset samples. 56
Figure 13. Visual comparision of cover and stego images (peppers and boat images). 58
Figure 14. The average capacities of methods using the first image dataset. 64
Figure 15. The number of LSB and PVD operations in PVD+LSB method. 64
Figure 16. Number of unused pixels for the OPVD, MOPVD and CMOPVD. 65
Figure 17. Average of PSNR, MPSNR and WPSNR...70
Figure 18. Comparing the average PSNR with the average capacity. 71
Figure 19. Comparing the average SSIM with the average capacity. 73
Figure 20. Comparing the capacity of HP and edge detection methods. 77
Figure 21. Comparing average PSNR of the HP and edge detection methods. 78
Figure 22. Image histogram test (aerial, baboon and peppers images). 79
Figure 23. Pixel-pair difference histogram (aerial, baboon and peppers images).............82
Figure 24. Fourier transform of stego-images for different methods. 86
Figure 25. The FTB sixteen fuzzy templates of size 3×3. 101
Figure 26. Fuzzy Template Based (FTB) edge detector flowchart. 102

THESIS ABSTRACT

Name: Azzat Ahmed Ali Al-Sadi

Title: TOWARDS ENHANCED STEGANOGRAPHIC METHOD FOR SECURE DATA TRANSMISSION OVER THE INTERNET

Major Field: Computer Networks

Date of Degree: May, 2012

Transferring data over the Internet has become a norm in our daily activities. Consequently, the need for more effective and robust security mechanisms to protect confidential data has substantially increased. Steganography is one of the widely used methods to hide secret data into other multimedia data (such as images, text, audio or video). Although a lot of research has been done to design good steganographic approaches, this field is still attracting the attention of many researchers due to the rapid spread of digital media and the fast growing sophistication in hacking methods.

In this work, we studied several existing methods for information hiding in digital images. Then, a new steganographic system was proposed to enhance the capacity, invisibility and security of the resulting stego images. The proposed system depends on three functions: modulus overlapping, chaotic block rotation and fuzzy-edge detection. The modulus overlapping is mainly to increase the embedding capacity by utilizing each pixel in the image individually. The chaotic function is to improve the security further by adding another level of challenge that makes the detection and extraction of embedded data much harder for any unauthorized person. Finally, the fuzzy-edge detection is to enhance the invisibility of the stego-image by dealing with the edge ambiguity problem.

We developed a prototype for the proposed system and several experimental tests were conducted to evaluate and compare its effectiveness with several other related methods. We also explored how robust these methods are to resist a number of well-known steganalytic attacks.

The results showed that the proposed system increased the embedding capacity and security while preserving a satisfactory quality with more than 30dB weighted peak signal to noise ratio. For instance, the average capacity has increased by almost a factor of 2 more than the original PVD (with a slight degradation in the image quality). We have also found that the security of the proposed system is excellent against the histogram attacks comparing to almost all surveyed methods. Using chaotic rotation in the proposed system significantly reduced the unusual steps in the pixel-pair difference histogram.

ملخّص الرسالة

الاسم: عزت أحمد علي السعدي

عنوان الرسالة: نحو إسلوب استيجانوجرافي محسّن لنقل البيانات بشكل آمن عبر الإنترنت

التخصص: شبكات الحاسوب

تأريخ التخرج: جمادى الآخرة 1433 هـ - (مايو 2012 م)

مع زيادة الاعتماد على شبكة الإنترنت في الآونة الأخيرة فى شتى ميادين الحياة ازدادت أهمية المحافظة على سرية البيانات أثناء نقلها على تلك الشبكة. ويعد مجال الاستيجانوجرافي (Steganography) أحد المجالات الحيوية لتحقيق ذلك؛ حيث يقوم بتضمين البيانات المراد الحفاظ على سريتها في وسط آخر (كرسالة نصية، أو صورة، أو مقطع صوت أو فيديو) عند المُرسِل قبل عملية نقلها ثم يقوم باسترجاعها عند المُستقبِل. ورغم أن هناك طرق عديدة لإخفاء البيانات في الصور، لا يزال هذا المجال بحاجة لطرق مبتكرة وفعالة للإخفاء مع تزايد اساليب القرصنة.

في هذة الرسالة تم دراسة عدة طرق مختلفة لإخفاء البيانات في الصور، وتم اقتراح وتطوير نظام إخفاء يهدف إلى زيادة القدرة على تضمين كمية أكبر من البيانات مع زيادة الحماية لهذة البيانات دون تأثير ملحوظ على جودة الصور المستخدمة في عملية الإخفاء. ويعتمد النظام المُقترح في هذة الرسالة على ثلاثة أساليب: المُعامل المُتداخل (Modulus overlapping)، التدوير العشوائي للكتلة (Chaotic block rotation)، وأسلوب تحديد الحواف للصورة باستخدام المنطق الضبابي (Fuzzy-edge detection). حيث يهدف الأسلوب الأول في الأساس إلى زيادة كمية البيانات المخفية في كل عنصر (بكسل) من عناصر الصورة، ويهدف الأسلوب الثاني إلى زيادة حماية البيانات المخفية في الصورة، بينما يهدف الأسلوب الثالث إلى الحفاظ على جودة الصورة بعد إخفاء البيانات فيها. كما قمنا في هذة الرسالة بإجراء العديد من التجارب لتقييم نظام الإخفاء المُقترح ومُقارنته مع طرق مختلفة.

CHAPTER 1

INTRODUCTION

Nowadays, the Internet is playing a major role in developed and developing societies. Hence, enormous amount of confidential information is being transmitted over the Internet. Since this information is vital to government, business, industry and even individuals; continuous technological improvements to secure this information is crucially needed. *Steganography* is one of the extremely important areas of information security. Unlike cryptography which changes the message to make it unreadable by an adversary (a third party) without knowing the key, steganography hides the presence of secret information. It uses a cover (carrier) medium to exchange secret information in undetectable way over a public communication channel. Both cryptography and steganography complement each other; thus, a message can be encrypted then embedded into a different medium. Steganography can be applied to different types of media including text, audio, image, video, etc. However, digital images are popularly used as cover media due to the simplicity of computation and the extensive use of images over the Internet with many different file types (such as bmp, gif and jpg). Figure 1 illustrates the principal components of a steganographic system. At the sender, the secret message is embedded into a cover object to generate a stego object (*a.k.a*. the cover containing the secret message). On the other side of the communication channel, the receiver extracts the secret message from the stego object. While the stego object is transmitted over an unsecure channel, such as the Internet, it can be captured and analyzed by an unauthorized person to reveal the embedded message; which is known as *steganalysis*. A good steganographic approach should be capable of embedding more data in the cover object without creating visible artifacts that can be used by the steganalyst. A key can be used optionally during embedding and hence will be needed during extraction.

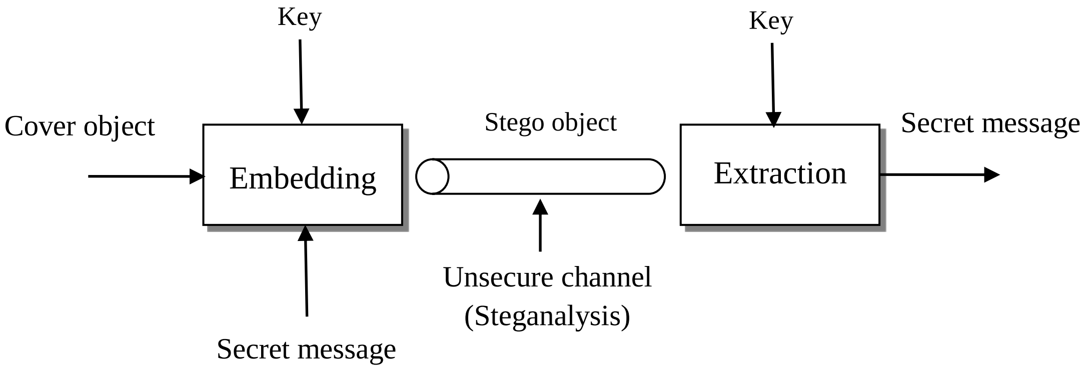


Figure 1. The principal components of a steganographic system.

Several steganographic approaches have been proposed in the literature. Most of these approaches embed more bits in edge pixels of the image. For instance, the Pixel-Value Differencing (PVD) [1] makes use of edges in the image to embed considerably large secret data without great quality loss. PVD utilizes a pixel-pair difference technique to categorize the smoothness properties of each pixel pair and adapts the number of embedded bits accordingly.

1.1. Motivation

Regardless of the amount of work that has been published to extend the idea of PVD, there is no comprehensive study that provides a detailed comparison between these methods. Furthermore, due to the crisp range boundaries, almost all PVD-related methods have a clear impact in the image histogram which makes them more fragile against some attacks. Moreover, the PVD-related methods use the pixel-pair difference to identify the image edges. However, other edge detection techniques such as those employing fuzzy logic can provide finer details about edges [2].

1.2. Problem Description

Although PVD has the potential to hide a large amount of secret data, it has some drawbacks. First of all, only two pixels are considered each time, therefore it cannot sufficiently capture edges in different directions [3]. Second, the falling-off-boundary procedure, applied when the resulting gray value of the pixel exceeds 255, has a significant problem; even with Wu and Tasi's solutions to detect and avoid these pixels in the embedding and the recovery processes. Third, most of the image areas are smooth; consequently, the secret bits will be embedded in ranges with small difference values [4]. Fourth, each pixel in the pixel-pair has its own characteristics; therefore it may hide different amounts of data from its neighbor. Fifth, the two-pixel block is non-overlapping, and it will lower the embedding capacity [3]. Sixth, PVD uses the pixel-pair difference technique to detect edges which is not the best technique for edge detection. However, new techniques such as fuzzy edge detection takes into account the ambiguity of edges in the image. Furthermore, PVD has unusual steps on the pixel-pair difference histogram which makes it vulnerable to security attacks [5].

1.3. Thesis Objectives

The aim of this work is to study and compare several existing approaches for information hiding in digital images. A number of comparison factors will be considered in this study including embedding capacity, visibility, and resistance to steganalytic methods. Then, a new information hiding system is proposed to take advantage of the recent development in chaotic theory and soft computing for edge detection to increase the amount of embedded data while maintaining security. The research objectives can be summarized as follows.

1) Study and benchmark existing techniques for information hiding in digital images that are based on pixel-value differencing (PVD).
2) Apply a number of steganalytic methods on various PVD-related methods to assess their robustness to these attacks.
3) Evaluate different edge-detection techniques and investigate how they can help in increasing the payload capacity of the steganographic embedding process.
4) Propose and assess a steganographic approach based on fuzzy inference systems.

1.4. Thesis Organization

The remaining chapters of this thesis are organized as follows. Chapter 2 provides a background and a literature review of the information hiding and the steganalysis methods. Chapter 3 studies in depth the PVD-related methods including their design methodologies. It also introduces the steganographic approaches which make use of the edge detection mechanisms. Chapter 4 presents and discusses the proposed steganographic system. Chapter 5 describes the experimental results, whereas Chapter 6 concludes this work.

CHAPTER 2

BACKGROUND AND LITRATURE REVIEW

2.1. Background

Steganography comes from the Greek word ‘steganos’ (στεγανός) meaning covered and ‘graphei’ (γραφή) meaning writing or drawing; thus steganography means covered or hidden writing. The basic idea of steganography is to hide the presence of secret data rather than enciphering it [6] [7]. Although steganographic approaches have been in use for a long time since the ancient days, it was only known by this name at the end of the 15th century. One of the earliest examples of steganography was dated back to around 440 B.C. when Histiaeus used to shave the head of his most trusted slave and tattooed his scalp with a message. Once the slave’s hair had grown, the message disappeared and the slave was sent to the receiver with the hidden message [8]. The slave’s hair was used as a cover for the message. Invisible ink was also used in both world wars. Some of the invisible inks were created using juice or milk [9].

With the rapid development and popularity of the Internet technology, secure communication between the sender and the receiver has become a significant challenge. Steganography, or concealing secret data into other media, plays an important role in creating covert channels and in protecting confidential data against unauthorized access and tampering; particularly in such open access environments. The object containing the secret message is called stego object.

There are many steganographic approaches which can be classified into the following types [10]:

- *Technical Steganography:* This type of steganography uses scientific methods to hide the secret message, such as invisible ink, microfilm and microdots that were used in both world wars.
- *Linguistic Steganography:* This type of steganography makes use of the written natural language such as dots and kashida in the Arabic language to hide a secret message.
- *Digital Steganography:* Digital steganography uses the computer technology to hide a secret message in a digital medium. It uses several multimedia covers such as image, audio and video.

Because of the extensive use of digital images in social networking services over the Internet such as Facebook and Netlog, digital steganography techniques are the most popular. In the following parts, we will study several existing approaches for digital image steganography.

2.1.1. Information Hiding and Data Confidentiality

Recently, Information hiding has become one of the important fields of the security because of the increasing number of Internet attacks which target invaluable information. Information hiding is mainly divided into two parts watermarking and steganography. Watermarking embeds a small amount of data into a cover object to protect the author's rights. This embedded data could be visible or invisible to the human eyes. Watermarking aims to prevent the embedded data from being removed by attackers [11] [12]. On the other hand, steganography hides large amounts of secret data into the cover object. It aims to protect the confidentiality of the embedded data.

With quick development in network technology, attacks have advanced rapidly. Therefore, the demand on transferring information securely has increased. This has led to develop new steganographic techniques to protect transmitted information over computer networks. However, steganography can be misused such as in the case of transferring secret information by the Internet worms [13]. Steganography uses several covert channels to transfer the information. It hides the secret data into network protocols, computer programs, text files and images. Because of the extensive use of images in the social networks, most of the steganographic techniques hide the secret data into digital images. This type of steganography is called digital image steganography. Digital image steganography can be divided into two types depending on its domain:

- *Spatial Domain Steganography:* Methods in this category modify the Least Significant Bits (LSB) of the cover-image pixels in the spatial domain. Although this type of steganography has several drawbacks such as its vulnerability to attacks, it is very common due to its simplicity. Several spatial steganography methods have been proposed. The LSB replacement is the most widely used. It directly replaces the least significant bits of the cover image with the secret message bits. This method can embed large secrets in the cover image, but it also introduces a clear distortion in the image histogram as more data is embedded. LSB matching is another technique of spatial domain steganography [14] [15]. It adds or subtracts 1 from the least significant bits of the cover image pixel when its value does not match the secret bits. In [16], the authors proposed a hybrid steganography approach using an optimal LSB substitution and genetic algorithm. Their method not only improves the quality of the stego image but it also protects the secret data. However, the computational requirements are high. Another method is proposed in [17]. This method improves the stego-image quality by decreasing the errors between the stego and the cover. Furthermore it reduces the computational overhead. In [18], the authors use the absolute difference of the neighboring pixels to determine whether the pixel can embed secret data without affecting the image quality or not. If the absolute difference of the neighboring pixels is greater than a predefined threshold, this pixel will not be changed. Another example of spatial domain steganography is the pixel value differencing technique [1]. It embeds the secret data into a pair of pixels. We will discuss this method in detail in Chapter 3.

- *Frequency Domain Steganography:* This type of steganography transforms the image into the frequency domain before the secret data is embedded. It protects the secret data by spreading it across the entire image. There are several mechanisms to transform the image into the frequency domain such as using the Z transform [19], wavelet transform and discrete cosine transform (DCT) [20]. Many methods have been proposed to develop this type of steganography; among them are F5 and OutGuess [21] [22].

In [21], F5 steganographic method is developed to convert the image into the frequency domain using the DCT transform. Then, it embeds the secret bits into the DCT coefficients by subtracting 1 from these coefficients if necessary. The F5 technique successfully resists some attacks including visual and statistical attacks. The OutGuess method consists of two stages. In the first stage, it embeds the secret message into the LSB of randomly selected DCT coefficients while skipping 0's and 1's. In the second stage, the histogram of the stego image is corrected to be similar to the cover image histogram as possible. The OutGuess method also resists some common attacks including the chi-square attack [22].

2.1.2. Security Attacks

Information hiding methods may suffer from several attacks. The art and science of analyzing an object to determine whether it has embedded data (stego-object) or not (cover-object) is known as steganalysis. The discrimination between a stego-object and a cover-object can be with or without the knowledge of the steganographic algorithm that was used for embedding the secret message [23]. Since steganography is such a secure form of communication and since it can easily be misused, steganalysis can be a useful tool under such conditions. Steganalysis has been used legally by governments to prevent terrorist attacks and catch people engaging in illegal activities.

Steganalysis methods can be classified into two general categories: method-specific methods and universal methods [24][25]. The first category targets a specific steganographic approach and attempts to attack that approach. The second category, the universal methods, which is also sometimes known as blind methods, is more general and can be applied to one or more steganographic approaches. In this category, features that are common to different steganographic approaches are first extracted and a classification model is built. The classifier is then used to detect stego-objects. Furthermore, steganalysis can be divided into two types according to the ability of the steganalysis method to reveal or estimate the secret message. Passive attacks can detect the presence of a secret message in the stego-object, and/or can identify which embedding algorithm is used. On the other hand, active attacks can estimate some extra properties such as the size of the embedded message, and/or extract a possibly approximate version of the secret message from the stego-object. Among the most popular statistical attacks is histogram attack where a graphical representation of the distribution of colors or grayscales in an image (*a.k.a*. histogram) is used to visualize the changes made due to embedding. It has been applied to detect embedding by methods based on least-significant bits (LSB) (e.g. LSB replacement and LSB matching) [4], [26], [27]. Although, visual artifacts are generally not noticeable by human eyes in the stego-image, changes in the histogram can be easily observed [28]. We will use the histogram attack later in the experimental part to evaluate the security of a set of common steganographic methods considered in this study.

2.2. Literature Review

There has been a growing interest in digital image steganography and many methods have been developed [29]. A good steganographic method is one that has high embedding capacity (payload) without visible artifacts. It should also resist steganalysis methods. In our study, we will focus on a family of methods used in spatial domain and attempt to make use of edge pixels to hide more data. Based on how these methods detect edges, we can divide them into two main categories. The first category detects edges using a group of pixels then it embeds the secret data in this group. The second category uses a traditional edge detection mechanism to detect edges in the whole image. Then, it embeds the secret data based on the edge information. In this section, we will conduct an intensive literature review of these two categories, including the most common attacks on them.

A) Group of pixels edge based methods

One of the well-known relatively recent approaches of this type is Pixel-Value Differencing (PVD) [1]. This approach was proposed by Wu and Tasi to hide a secret message into 256 gray-valued images. To preserve good quality of the stego-image, their approach utilizes the edge bits for embedding more data. PVD uses the difference of each pair of pixels to determine the number of bits that can be embedded into this pixel-pair. A small difference value indicates that the block is in a smooth area, and a large value indicates that it is in an edge area. The larger the difference, the more data can be embedded.

Although, PVD can embed more data in edges, it does not utilize the smooth areas sufficiently. Moreover, two pixels cannot capture the different directions of edges. PVD detects only the vertical edges. However, edges can also exist in horizontal, vertical and diagonal directions; but won't be detected. Additionally, the two-pixel blocks are non-overlapping which results in lowering the embedding capacity. Furthermore, PVD has a clear impact on the image histogram that exposes it to histogram-based attacks. In [5], the authors presented an analysis of the changes in the histogram of the pixel difference due to embedding of secret data into a cover image using PVD. Moreover, PVD was successfully attacked by generating a substitute image which is created from the pixel-pair difference vector of the stego-image. Then, apply the chi-square steganalysis on the substitute image to detect the presence of the embedded data [30].

In [5], another approach based on PVD is proposed to increase the immunity of PVD to the histogram steganalysis. Instead of the fixed ranges of the original PVD, variable ranges for different blocks are introduced. The authors generated new variable ranges using a pseudo-random parameter. By varying the value of the pseudo-random parameter appropriately, the steps on the histogram of the pixel-pair difference can greatly disappear. But, Sabeti *et al*. [31] attacked this approach using a universal detector based on neural networks.

To improve the stego-image quality and eliminate the PVD histogram steps, Wang *et al.* [32] proposed the PVD with modulus function approach. This approach modified the remainder of the pixel-pair calculations instead of using the difference value. They also overcome the falling-off boundary when the pixel exceeds the value of 255 after data has been embedded by using readjusting conditions. This method increased the PSNR (Peak Signal to Noise Ratio) more than the original PVD method in most of the considered cases. Despite its security against LSB attacks such as RS attack [33], the embedding process can still cause a number of artifacts, such as abnormal increases and fluctuations in the PVD histogram, which has been used as a clue to reveal the existence of hidden data [34] [35]. An attack on the modulus PVD is proposed in [35], using three steganalytic measures and a support-vector machine. In order to enhance the security of the modulus PVD, a turnover policy with a novel adjusting process is proposed in [36] to prevent abnormal increases in the histogram values and remove fluctuations at the border of the various ranges in the PVD histogram. However, the modulus PVD does not tackle the PVD capacity problem.

To further enhance the PVD capacity, Wu *et al.* [37] used a combination of the PVD and LSB replacement methods to embed more data into the smooth areas. This approach is based on the idea of using PVD when the difference between the pixel-pair is large (edge area), and using LSB with three bits per pixel and readjusting equations whenever the difference is small (smooth area). Although, this PVD+LSB with readjusting equations can hide more secrets than the original PVD, it has many characteristics similar to the simple LSB. Cheng *et al.* [4] attacked PVD+LSB using a similar method to Fridrich *et al*.'s steganalysis method [33]; yet this method failed to detect the original PVD. This is because most of image areas are smooth and consequently the majority of the cover image pixels will be altered using the LSB replacement method. PVD+LSB with readjusting equations was also successfully attacked in [30] using the chi-square method.

Yang *et al*. [4] enhances the image quality of the PVD+LSB by using a selective strategy instead of using the 3-bit LSB with readjusting equations whenever the pixel-pair difference of the stego image belongs to a smooth area. Furthermore, they applied the well-known modified LSB substitution method [38] [39] to PVD+LSB with readjust instead of using the simple LSB. However, PVD and its modified version PVD+LSB still use only two pixels in each block to detect the edges, which does not give enough information about the surrounding area. Consequently this may lower the embedding capacity.

To eliminate the drawback of using two pixels for detecting edges, Chang and Tseng [40] hide secret data using the concept of two-sided-match vector quantization (SMVQ) [41]. SMVQ utilizes information from two neighbouring pixels (the upper and the left pixels) to detect edges. Despite that the two-sided match can detect more edges than the PVD, it distinguishes only the horizontal and vertical edges, whereas edges can also present diagonally. To improve the edge detection accuracy, and also to increase the embedding capacity, a Tri-way PVD (TPVD) method was proposed in [42] [43]. This method embeds the secret bits in both horizontal and vertical edges of the cover images in addition to only one diagonal edge. However, the variable amount of embedded data in different directions by the TPVD method causes considerable distortion in the stego image. Furthermore, the fixed ranges create clear steps on the histogram of the pixel-pair difference. Utilizing this distortion and the branch conditions of the Tri-way, Zaker and Hamzeh [44] successfully attacked the TPVD method. The idea of their steganalysis was to find the characteristics of the cover image pixel-pairs differences from the stego image. They used the suspected image as a cover image to embed particular amounts of additional secret bits with the same procedure of TPVD. After that, the histogram of the pixel-pair differences was used to compare the characteristics of the suspected image before and after embedding the secret bits. They used the length of unusual steps at the boundary of ranges to detect the existence of any secret message. This steganalysis not only detects the stego image, but it can also estimate the size of the secret message.

Using a four-pixel block, Yang and Weng [3] proposed a Multi-Pixel Differencing (MPD) approach. The smallest gray value in a four-pixel block is used to create three groups of pixel pairs. Therefore, instead of hiding data using the difference between two pixels, as in PVD, Yang and Weng’s approach can give the differences of the three groups; hence it will increase the size of hidden data. However, the MPD approach relies only on the difference between block pixels. The difference does not hide many bits in the smooth area as we discussed before. Jung *et al*. [45] suggested adding a *threshold* level to discriminate between the edge and smooth areas. This threshold level will also be used as a secret key. They embedded data with LSB method whenever the difference of the pixel-pair is less than the threshold level; otherwise they used the MPD method.

Moreover, for minimizing the distance between the pixel pair for each sub-block in the edge area, a method is used for rearranging the new pixel values.

Despite the different techniques for selecting a group of pixels and the way of identifying edges, all previous methods embed the secret data in a pixel-pair. However, each pixel in a pixel pair can have different values and characteristics, therefore it may hide different amount of data from its neighbor. To utilize the characteristics of each pixel, Chang *et al*. [46] proposed a steganographic approach that hides the secret data in each pixel individually, instead of hiding it in a pixel-pair. They hide the secret bits into the least-significant bits of the second pixel in each block of two pixels. Then this method uses the second pixel of the first block as the first pixel of the second block. Although, this approach improves the PVD capacity, it leaves many pixels without embedding.

B) Traditional edge-based methods

The second steganographic type in our literature survey detects the edge information in the whole image before embedding the secret data. To detect edges, this type uses one of the traditional edge detection methods. Edge-detection methods calculate the edge strength by the amount of change in the gradient values of the image pixels. Since edges can be represented in different directions, a good edge detecting approach is able to consider all edge directions. Several edge detection approaches have been proposed. Some of them use the first-order or the second-order derivatives. These approaches are called classical edge detectors [47] [48]. Other approaches use soft-computing such as fuzzy logic to identify edges [49].

The classical edge-detection methods utilize the principle of matching image segments with the specific edge patterns to identify the edge location and direction. The edge is recognized by convolving the image signal with a set of directional derivative marks. Some examples of classical edge detectors are Roberts, Sobel, Prewitt, Canny and Laplacian edge detectors. These algorithms are simple and easy to apply on images [50]. On the other hand, fuzzy edge detection methods consider the image to be fuzzy. This consideration solves the problem when edge detection becomes difficult because of the vague or blurred characteristics of the edges. Consequently, fuzzy systems add more improvement to the edge detection field. There are many fuzzy edge detection approaches. One such approach is based on the intuitionistic fuzzy distance which is proposed in [51] [52]. In our work, we will refer to this method as Fuzzy Template Based (FTB) edge detector to differentiate it from other edge detection methods.

Utilizing one of the edge detection methods, the traditional edge-based steganographic method first generates the edge image; an image containing information about all edges in the image. Then, either the steganographic method embeds the edge information into the stego image or uses the same cover at the receiver side to reproduce the same edge information.

In [53] [54], the authors generate the edge image using a hybrid edge detection method. Then, they embed the secret data into the cover image without paying attention to the edge information. However, at the receiver side, the original cover image is needed to regenerate the same edge information before extracting the secret data. Because the cover image is needed at the receiver side, this approach is either restricted to some predefined cover images or the cover image should be securely transmitted to the recipient every time. Alternatively, the authors of [55] [56] embed the edge information into the stego image. Therefore, there is no need for the original cover at the receiver side.

More details about the PVD-related methods and the traditional edge based steganographic approaches are discussed in the next chapter.

CHAPTER 3

EXISTING STEGANOGRAPHIC SPATIAL METHODS

In this chapter, we will review and discuss in more details several spatial domain steganographic methods to identify their strengths and weaknesses. These methods are based on two approaches for distinguishing smooth and edge areas in an image: pixel value differencing related methods and traditional edge detection based methods. The goal of this chapter is two folds. First, it helps us shape our ideas towards a more effective steganographic approach. Second, we will conduct several experiments, later in Chapter 5, to compare these methods and also to benchmark our proposed steganographic system.

3.1. PVD-Related Methods

In this section, we discuss seven steganographic methods based on pixel-value differencing to detect edge areas and determine the number of secret bits to be embedded.

A) PVD Method

The original PVD approach was proposed by Wu and Tasi to hide a secret message into 256 gray-valued images [1]. Instead of inserting a fixed number of secret bits directly into the least significant bits of each byte of the cover image, PVD uses the difference of each pair of pixels to determine the number of bits that can be embedded into this pixel-pair. PVD relies on the fact that human eyes can observe small changes in the gray level values of a smooth area but they cannot easily notice the changes at the edge areas. Hence, PVD partitions the cover image into blocks by scanning the cover image from the left-upper corner in a zigzag manner. Each block consists of two consecutive non-overlapping pixels. The differences of the two-pixel blocks are used to categorize the smoothness and contrast properties of the cover image. The pixels around an edge area will have large differences whereas the pixels at a smooth area will have small differences. The larger the difference, the more bits can be embedded into this block.

Wu and Tasi segmented the gray level range (0, 255) into smaller ranges. To facilitate binary data embedding, each range must be a power of 2. Ranges with small widths represent the smooth areas and ranges with large widths represent edge areas. In their paper, they have experimented with two different sets of ranges: {8, 8, 16, 32, 64, 128} and {2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, 64}. Each range is demarcated by u_i and l_i which represent the upper and lower levels of this range, respectively. Each range determines the number of bits that will be hidden in a pixel pair as given by $n_i = log_2(u_i - l_i + 1)$, for range i. Assume P_i and P_{i+1} are two pixels of a pixel-pair, and g_i and g_{i+1} are their gray values. The difference d_i is calculated as $g_{i+1} - g_i$ and its absolute value falls in the range from 0 to 255. Let the difference value after embedding be d_i' which can be calculated from:

$$d_i' = \begin{cases} l_i + b_k, & if\ d \geq 0 \\ -(l_i + b_k), & otherwise \end{cases} \qquad 3.1$$

where b_k is the decimal value of some secret bits to be embedded in this block using the following equation:

$$(g'_i, g'_{i+1}) = \begin{cases} g_i - \lceil m_i \rceil, g_{i+1} + \lfloor m_i \rfloor, \ \text{if } d_i \text{ is odd} \\ g_i - \lfloor m_i \rfloor, g_{i+1} + \lceil m_i \rceil, \ \text{if } d_i \text{ is even} \end{cases} \qquad 3.2$$

where $m_i = (d'_i - d_i)/2$.

The new pixel values may fall outside the boundary (0, 255), which is not a valid gray level value; hence the secret data will not be embedded in these pixels. Wu and Tasi proposed a falling-off-boundary procedure to discover these pixels and skip them. Figure 2 illustrates the main steps in the embedding process of PVD.

Although PVD has the potential to hide a reasonable amount of secret data, it has some drawbacks. First of all, only two pixels are considered each time, therefore it cannot capture the different edge features sufficiently [3]. Second, the falling-off-boundary procedure has a significant problem even with the solution proposed by Wu and Tasi. Third, most of the image areas are smooth, so few secret bits will be hidden using the ranges with small values [4]. Fourth, each pixel in the pixel-pair can have different values, therefore it may hide different amount of data from its neighbor. Fifth, the two-pixel blocks are non-overlapping, and this will lower the embedding capacity [3].

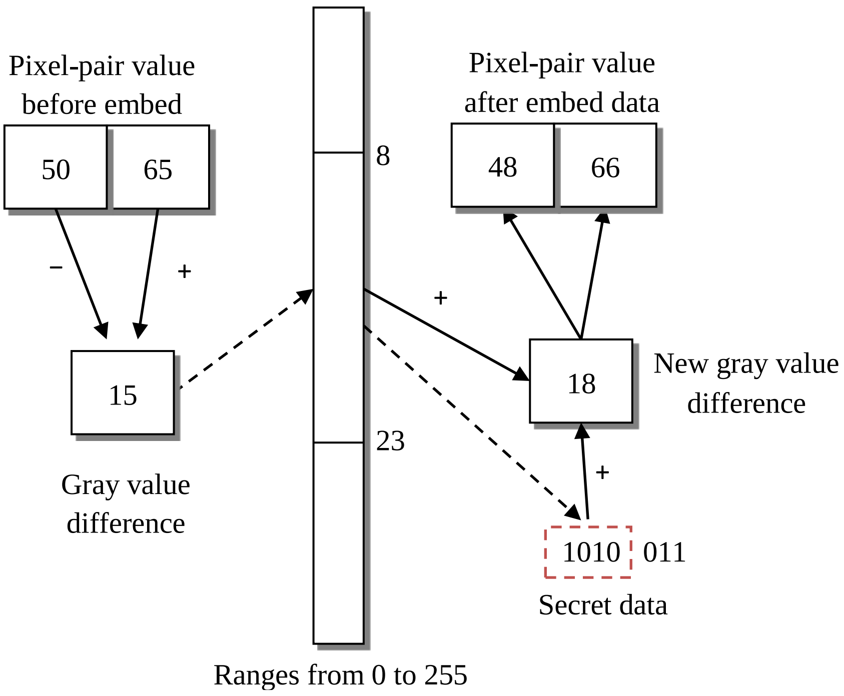

Pixel-pair value before embed
50
65
−
+
15
Gray value difference
8
23
Ranges from 0 to 255
+
Pixel-pair value after embed data
48
66
18
New gray value difference
+
1010 011
Secret data

Figure 2. The process of embedding secret data using PVD.

Moreover, the pixel value differencing method is not very sensitive to straightforward histogram analysis as compared to LSB. However, by drawing the histogram for the differences of pixel pairs, variations before and after embedding can be clearly observed. The histogram of the differences of pixel pairs has a smooth shape of a normal distribution whereas it has remarkable steps for the stego-image. This is due to the quantization ranges of the PVD method. When different differences fall in the same range, the calculation of the new differences will start from the same lower boundary of that range. In general, the number of occurrences of a pixel difference decreases with the increase of the absolute value of the difference. In [5], the authors presented an analysis of the changes in the histogram of the pixel difference due to the embedding of secret data in a cover image using PVD. This analysis can be summarized as follows. The secret bits are assumed to be uniformly distributed (e.g., as a result of encryption before embedding) in the range [0, w_i-1], where w_i is width of range i. When $i > 0$, it will make the number of differences falling into r_i, r_0 and r_{i-1} and their boundaries are [l_i,u_i], [-u_0, u_0] and [-l_i, –u_i], respectively; as shown in Figure 3. The pixel difference histogram of the stego-image $\tilde{h}(d)$ will be approximated by [5]:

$$\tilde{h}(0) \approx (1-\alpha)\times h(0) + \frac{\alpha \times r_0}{w_0} \tag{3.3}$$

$$\tilde{h}(d) \approx (1-\alpha)\times h(d) + \frac{\alpha}{w_0} \times \sum_{j=0}^{u_0} h(j), 0 < d \le u_0 \tag{3.4}$$

$$\tilde{h}(d) \approx (1-\alpha)\times h(d) + \frac{\alpha}{w_0} \times \sum_{j=-u_0}^{-1} h(j), -u_0 \le d < 0 \tag{3.5}$$

$$\tilde{h}(d) \approx \begin{cases} (1-\alpha)\times h(d) + \dfrac{\alpha \times r_i}{w_i}, l_i \le d \le u_i \\ (1-\alpha)\times h(d) + \dfrac{\alpha \times r_{-i}}{w_i}, -u_i \le d \le -l_i \end{cases}, i > 0 \tag{3.6}$$

$$\alpha = \frac{no\ of\ blocks\ contains\ secret\ data}{Total\ number\ of\ blocks} \tag{3.7}$$

A gap will appear between $\tilde{h}(d)$ and $\tilde{h}(d+1)$ when their differences belong to two different ranges, because the difference between r_i / w_i and r_{i+1} / w_{i+1} is greater than the difference between $h(d)$ and $h(d+1)$ [5].

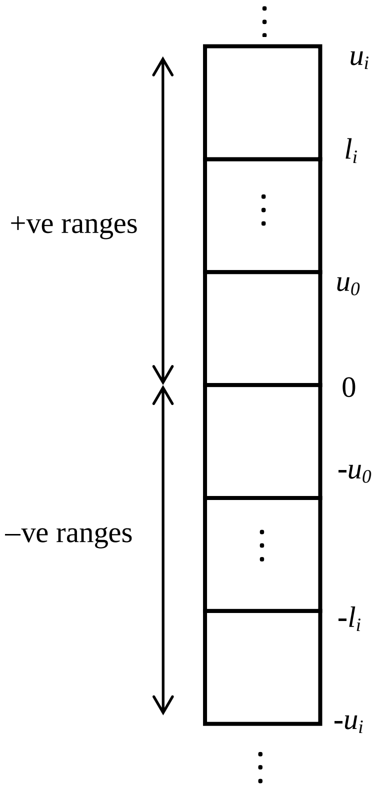


Figure 3. Ranges and their boundaries.

Despite the PVD drawbacks, it can be enhanced in terms of security, capacity and image quality. In the following, we will discuss a number of other methods that have been proposed to extend the original PVD method in various ways.

B) PVD+LSB Method

Because the PVD method does not utilize the smooth area to hide large number of secret data, its capacity is relatively low. In order to achieve higher capacities, Wu et al. [37] used a combination of PVD and LSB to hide the secret data. This method is based on the idea of using PVD when the difference between pixels in a pixel-pair is large (edge area), and uses 3-bits LSB per pixel with readjusting equations whenever the difference is small (smooth area). The discrimination between the edge area and the smooth area is determined by comparing the difference between the pixel-pairs with a threshold value, *div*. This threshold value is controlled by users and used as a secret key.

During the embedding process, the difference d_i is calculated similar to the original PVD. If $d_i < div$, then the pixel-pair belongs to a smooth area and 3 bits of the secret message will be directly embedded into the least significant bits of each pixel using LSB. The new difference, $d_i^{'}$, will be calculated after the data is embedded and compared with the threshold, *div*. If $d_i^{'} \geq div$, a readjusting equation will be used; otherwise, the pixels belong to an edge area and the original PVD method is used instead. The readjustment equation is as given by:

$$\left(g^{'}_{i}, g^{'}_{i+1}\right) = \begin{cases} \left(g^{'}_{i} - 8, g^{'}_{i+1} + 8\right), \ if \ g^{'}_{i} \geq g^{'}_{i+1} \\ \left(g^{'}_{i} + 8, g^{'}_{i+1} - 8\right), \quad otherwise \end{cases} \tag{3.8}$$

The PVD+LSB method can have about 1.57 to 1.97 greater capacities than the original PVD method, but the value of the PSNR will be dropped by about 2.1 to 4 *dB* [57]. The high value of PSNR when using PVD only results from the scare modification of pixels of the cover image especially in the smooth areas. Further discussion of the drawbacks of PVD+LSB can be found in [4].

C) Side-Match Method

Using only two pixels in each block does not give much information about the surrounding area. Also it can cause noticeable distortion in the stego-image. To eliminate this defect, Chang and Tseng [40] used a concept of two-sided match vector quantization (SMVQ) which was developed by Kim [41] to hide secret data. SMVQ utilizes the information from two neighbouring pixels (the upper and the left pixels) to predict the state of the current pixel if it is located in an edge area or not. Three-sided and four-sided match methods for VQ encoding were also proposed in [58]. Using a raster scan, Chang and Tseng scanned the whole image except the first row and first column. Assume the current pixel is P_x and its upper and left neighboring pixels are denoted by P_{ux} and P_{lx} respectively. The difference $d_x = (g_{ux} + g_{lx})/2 - g_x$ where g_{ux} and g_{lx} are the gray values of pixels P_{ux} and P_{lx}, respectively. If the value $d_x \in \{-1, 0, 1\}$, then one bit of the secret data is embedded into the least-significant bit of the pixel P_x using the conventional LSB substitution approach. Otherwise, the number of bits to be hidden, n, and the difference $d_x^{'}$ are calculated using the following equations:

$$n = log_2 \lfloor d_x \rfloor \; ,if \;\; d_x > 1 \qquad 3.9$$

$$d_x^{\,'} = \begin{cases} 2^n + b, & if \;\; d_x > 1 \\ -\left(2^n + b\right), & otherwise \end{cases} \qquad 3.10$$

where *b* is the decimal value of the secret data to be embedded, $P_x^{'}$ is the stego-image’s pixel that contains the secret data, and its gray value is $g_x^{'} = (g_{ux} + g_{lx})/2 - d_x^{'}$.

D) MPD Method

To reduce the error of the Side-Match method, and also to increase the embedding capacity, Yang and Weng [3] proposed a Multi-Pixel Differencing (MPD) method. Similar to PVD, their approach uses raster scanning, but instead of taking two pixels as a block, they select a block of four pixels to hide the secret data. The four pixels must satisfy the condition $g_0 \leq g_1, g_2, g_3$, i.e. g_0 is the pixel with the smallest value and g_1, g_2, g_3 are the next pixels in the clockwise direction in the same order. But if there exist more than one pixel with the smallest gray value in a block, g_0 is assigned to the first pixel of those candidates in the sequence. Three groups are created as follows: group_1 ($g_1 - g_0$), group_2 ($g_2 - g_0$), and group_3 ($g_3 - g_0$). Each group difference falls in one of the predetermined ranges r_i. Therefore, instead of hiding data using the difference between two pixels, as in [1], Yang and Weng's approach can give the differences of the three groups; hence it will increase the hidden data. This approach can embed n_1, n_2, and n_3 for group 1, 2 and 3, respectively.

E) Tri-way PVD Method

The original PVD method can only hide up to seven bits at most in each pixel-pair. Moreover, it detects the edge by only two horizontal pixels. To increase the capacity and to get more edge information, Chang *et al*. [42], [43] proposed a Tri-way PVD scheme which utilizes a block of four pixels to detect edges in different directions and to hide secret data. Their approach divides the cover image into non-overlapping 2×2 blocks with four combinations of pixel-pairs as shown in Figure 4. The four block pixels will be

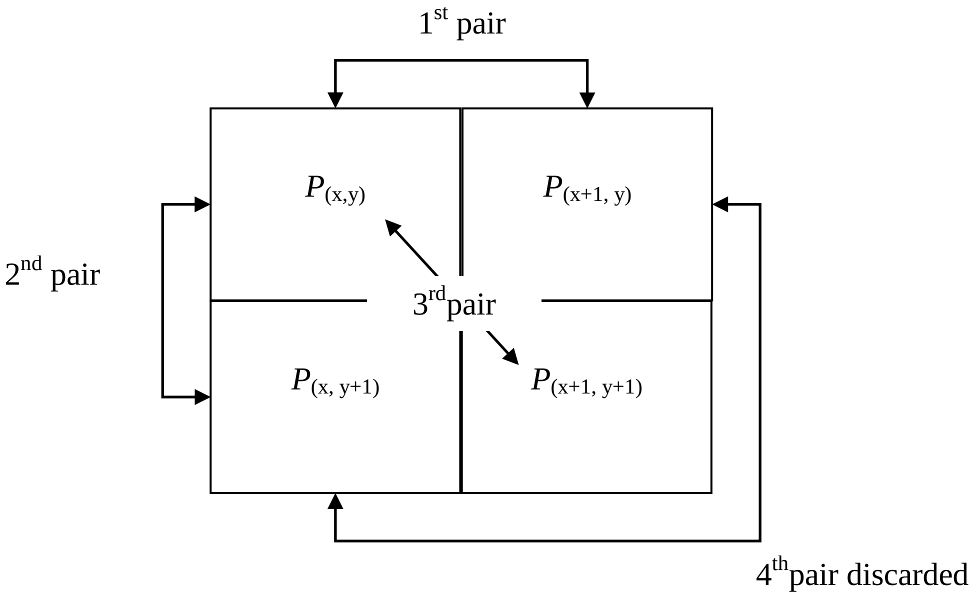


Figure 4. An example of four pixel pairs.

denoted as $P_{(x,y)}, P_{(x+1,y)}, P_{(x,y+1)}$ and $P_{(x+1,y+1)}$ where x and y are the coordinates of the pixel position in the image. The four pixel pairs will be named as P_0, P_1, P_2 and P_3 where $P_0 = (P_{(x,y)}, P_{(x+1,y)})$, $P_1 = (P_{(x,y)}, P_{(x,y+1)})$, $P_2 = (P_{(x,y)}, P_{(x+1,y+1)})$, and $P_3 = (P_{(x,y+1)}, P_{(x+1,y+1)})$. However, the fourth pixel-pair is discarded; this is because changing its pixel values will affect the first and the second pixel pairs. Therefore, the Tri-way PVD method can embed secret bits in horizontal, vertical, and diagonal edges of the cover image.

To reduce the distortion from hiding data in different directions, Chang *et al.* proposed branch conditions technique. If one of the branch conditions occurs, the original PVD will be used instead of hiding data using the Tri-way PVD method. The original PVD method will hide the data in P_0 and P_3 pixel pairs. The branch conditions are:

- $\text{embedding\_bit}\ (P_0) \geq 5$, and $\text{embedding\_bit}\ (P_1) \geq 4$
- $\text{embedding\_bit}\ (P_0) < 5$, and $\text{embedding\_bit}\ (P_2) \geq 6$

F) OPVD Method

To enhance the capacity of PVD further, Chang *et al*. [46] proposed a concept of overlapping to increase the number of differenced pixel-values. By using this concept, they achieved higher average hiding capacity over the original PVD of Wu and Tasi. Chang *et al*.'s approach is based on hiding a secret data using individual pixel, instead of hiding it in a pixel-pair. They hide secret bits into the least significant bits of the second pixel in each block of two pixels. Then, this approach uses the second pixel of the first block as the first pixel of the second block. Figure 5 demonstrates the concept of overlapping pixel-value differencing (OPVD) and contrasts it to PVD.

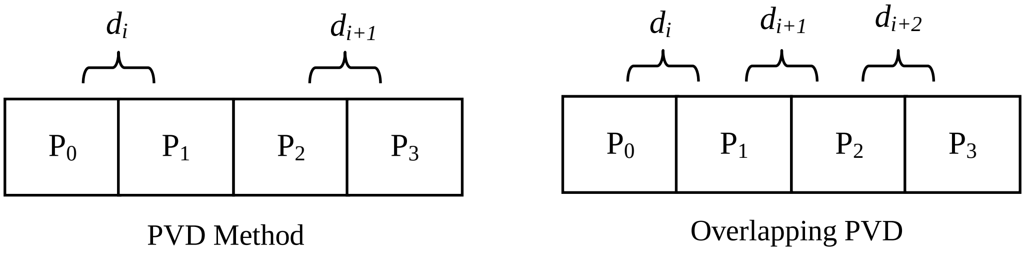


Figure 5. Demonstration of the difference between PVD and OPVD methods.

If the pixel-pair difference before and after hiding the secret data belongs to the same range, the embedding process is successful; otherwise the secret bits cannot be embedded and the second pixel P_2 is adjusted to indicate this situation. P_2 is moved to the smallest value or the biggest value of the range according to the following equations:

$$d_j^{'} = \begin{cases} b_j L, & if \ \left| b_j L - d_j \right| \le \left| b_j H - d_j \right| \\ b_j H, & Otherwise \end{cases} \tag{3.11}$$

Whether the embedding process is successful or not, the second pixel will be the first pixel in the next pixel pair. Although the OPVD method can hide larger data than the original PVD, it has some drawbacks:

1) The arrangement of the table of ranges has a great influence on the image quality and the hiding capacity.
2) It suffers from the problem of unused pixels which reduces its embedding capacity.
3) Using simple LSB method to hide up to seven bits per pixel deforms the stego-image histogram.

G) Modulus-PVD Method

Wang *et al*. [32] proposed PVD with modulus function steganographic method to enhance the image quality by reducing the difference between the pixel pair before and after embedding of secret data. Instead of using the difference value, their approach modified the remainder of the pixel-pair. As a result, this method increases the PSNR more than the original PVD method. In addition, the falling-off boundary problem when the pixel exceeds the value of 255 after data has been embedded is solved by using readjusting conditions. The modulus PVD can be briefly described in the following steps:

- Find the difference between consecutive pixels similar to the original PVD and determine the range where this difference falls.
- Compute the remainder using the following equation:

$$F_{rem(i)} = (P_i + P_{i+1}) \bmod t_i^{'} \tag{3.12}$$

  where $t_i^{'} = 2^{t_i}$ and t_i is the hiding capacity of the pixel block.

- Embed *n* secret bits into the pixel block such that the equivalent decimal value *b* is equal to F_{rem}.

To keep the difference in the same range before and after the embedding, a method to alter the remainder of the pixel-pair is proposed in [32].

3.2. Traditional Edge Based Methods

Another direction in steganography is the application of traditional edge detectors. The proposed methods in this category of steganography differ in way the sender shares the edge information with the receiver. Some methods are cover dependent which means both the stego-image and the cover image are needed at the receiver side to recover the secret data. Other methods are cover independent; thus only the stego-image is needed at the receiver side.

A) Cover-Dependent Steganography

This type uses the cover image at the sender side to produce the edge information and to identify the edge pixels. Then based on the edge information, more secret bits will be embedded in edge areas. However, during the embedding process, the edge's information will be changed. Therefore, at the receiver side the cover image is needed to generate the original edge information and to extract the secret bits correctly. Algorithms in [53], [54] are examples of this type. The main disadvantage of this type is the inflexibility. Because the cover image is needed at the receiver side, this type will be restricted on some predefined cover images or the cover image should be transmitted securely every time.

In [53], the edge image is generated from the cover image using a hybrid edge detection mechanism. To increase the number of detected edges, this method uses a combination of several edge detection algorithms namely Sobel, Prewitt, Zero crossing, Robert, Log and Canny algorithms. Furthermore, the authors use two shared keys for encrypting the secret bits before embedding. The first key is used to encrypt five secret bits for each edge pixel, whereas the second key is used to encrypt two secret bits for each non-edge pixel. Although, this method embeds more bits in the edge pixels, the problem of transmitting the original cover still exists. Moreover, the exchange of the long shared keys is another drawback.

In [54] the authors generate the edge image by utilizing the information from three neighboring pixels to identify the smoothness and contrast of the target pixel. The secret message bits are embedded in the smooth pixels. This method embeds variable amount of secret bits. The mechanism of this method is based on embedding more than two bits, since there is a similarity between the message and the LSB of the target pixel. Otherwise they embed only two bits. When the embedded data are more than two bits, the information about the last occurrence of the embedded data is saved in the three LSB bits of the corresponding edge pixel.

B) Cover-Independent Steganography

The second steganographic type is characterized by the ability to extract the edge information at the receiver side without the need for a cover image. To preserve the same edge information at the receiver side, this type of steganography encodes edges information and embeds this information with the secret bits into the cover image. The receiver side must know the locations of the encoded information and must have the ability to decode this information in order to retrieve the edge pixels. After retrieving the edge information, the receiver side can extract the secret bits correctly. Algorithms in [55][56] are some examples of this type.

In [55], the edge image is obtained from the grayscale image using a hybrid edge detector. Then the edge image is divided into a set of blocks. Each block consists of n pixels; where n must be no greater than nine. But to achieve good image quality, n should be less than or equal to five. These n pixels are denoted as $P_1, P_2 \dots P_n$. Authors of [55] use the first pixel P_1 in n-pixel block to store the status of the remaining pixels in that block. The status of the remaining pixels is defined as '1' if the pixel is an edge pixel. Otherwise the status of the pixel is defined as '0'. The status of the pixels will be stored in the LSB bits of P_1 using the LSB substitution method. For example, for a block of three pixels (P_1, P_2, and P_3), $n = 3$. Assume P_2 is an edge pixel whereas P_3 is a smooth pixel. To store the status of these pixels in P_1, the LSB bits of P_1 will be replaced by '10'.

The number of secret bits that will be embedded in the block pixels will vary depending on the pixel status. If the pixel is located in an edge area, three bits from the secret message will be embedded in the LSB of this pixel otherwise, only one secret bit will be embedded in the LSB of the smooth pixel.

Although this method can embed large amount of secret bits, it wastes about quarter of the image capacity in indexing the edge and the smooth pixels. The authors of [55] recommended using a block of four pixels because increasing the block size will affect the image quality. Furthermore, the authors claimed that the embedding of three bits in edge pixels and one bit in smooth pixels can preserve acceptable image quality. But using three bits from the first pixel in each block to store the status of the remaining pixels can greatly impact the quality since the majority of these pixels will be located in smooth areas of the image [4]. Consequently, the capacity and the quality of this method can be improved further. This can be achieved by reducing the number of bits that can used to identify edges and smooth pixels and by reducing the number of pixels that can be used to store the status, this method can be improved even without changing the number of embedded bits in edge and smooth pixels as we will show in Section 5.3.1.

CHAPTER 4

THE PROPOSED STEGANOGRAPHIC SYSTEM

As mentioned before, security, capacity and invisibility are three crucial aspects for a good steganographic method. Although PVD has some advantages in terms of embedding capacity and PSNR [57] [59], it can be enhanced further. In addition, PVD has some drawbacks including: unusual steps in the pixel-pair histogram, inability to detect edges in different directions, and insufficient utilization of smooth areas. In this chapter, we are going to present a better steganographic system that can achieve good steganography aspects including security, capacity and invisibility. Each one of those aspects has its own purpose and requirements. Our proposed system is based on three main functions: chaotic block rotation, modulus with overlapping, and fuzzy logic. These functions mainly aim to improve security, increase embedding payload, and detect edges more efficiently, respectively. These components are depicted in Figure 6 and will be explained in more details in the following sections.

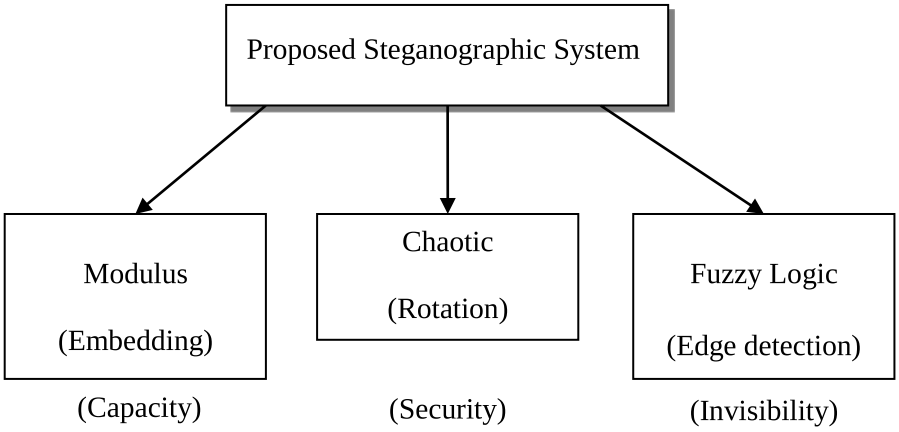


Figure 6. The components of the proposed system.

4.1. Chaotic Block Rotation

The objective of the chaotic block rotation function is to increase the security of the steganographic system without considerable effect on the image quality or the image capacity. It adds another level of security that makes the extraction of the embedded data harder for the unauthorized person. Furthermore, block rotation helps to defeat the pixel-pair difference histogram attack [5] for the original PVD because it breaks the systematic way of embedding by randomizing the pixel-pair differences directions as we will discuss in Section 5.3.2. The chaotic rotation is based on dividing the image into 2 × 2 blocks and rotating the blocks into two different directions. This rotation depends on a stochastic sequence that is easy to generate at both the sender and the receiver side. The generation of this sequence is performed using a logistic-chaotic map that depends on two parameters: initial condition and control parameter. These two values act as a shared stego-key between the sender and the receiver.

4.1.1. Logistic-Chaotic Map

The logistic-chaotic map is a simple approach for generating a stochastic sequence from a non-linear difference equation based on two parameters: an initial condition, x_0, and a control parameter, r. Mathematically, it can be expressed as follows:

$$x_{n+1} = r x_n (1 - x_n) \tag{4.1}$$

where n is the state number (time index). The initial state value, $x_{0,}$ is a number in the range (0, 1), whereas the control value, r, is a real number and should be in the range (3.57, 4) to achieve the maximum randomization. The generated sequence of random numbers depends only on these two parameters and hence they should be shared between the sender and the receiver to generate the same random sequence. By varying the value of the control parameter, the generated sequence is bifurcated and it has been shown that an infinite random sequence can be generated when $r = 3.599692$ [60].

Due to its relative simplicity, the logistic map is one of the excellent chaos mechanisms. It was first popularized by Robert May in 1976 to estimate the population of a specific year [60]. It has been previously used in encryption and steganography, but in different ways than what we propose here in this thesis. In [61], the authors utilize the logistic maps with Chebychev chaotic for image encryption. To increase the immunity to attacks, their approach is based on double logistic systems to generate chaotic sequences. Utilizing the chaotic theory in [62], the authors hide a secret message after encryption into the spread spectrum of the digital images. To achieve this, they generate three different chaotic keys. The first key is for message encryption whereas the second and third chaotic keys are used for message modulation and embedding into a cover image, respectively. Moreover, in [63] a steganographic method is proposed for embedding data into the frequency domain of JPEG images. This approach utilizes the chaotic logistic map to shuffle the order of bits in the message. The parameters of the logistic map are selected using genetic algorithms to increase the image quality. The secret message is embedded into the image frequency coefficients using an adaptive version of the LSB method.

4.1.2. PVD with Chaotic Block Rotation: Embedding

One of the main PVD drawbacks is the unusual histogram steps as discussed in Section 3.1.A. In this part, chaotic rotation will be applied on the original PVD and its impact on security will be studied. The embedding procedure for PVD with chaotic block rotation is shown in Figure 7. It includes the following steps. First of all, the cover image is partitioned into non-overlapping blocks of size 2×2. This is achieved by scanning the cover image starting at the upper-left corner. Each block consists of two consecutive non-overlapping pairs. Then, based on the secret key generated using the logistic chaotic map, a rotation direction is selected which can be either left (counter-clockwise) or right (clockwise) as illustrated in Figure 8. Changing the direction of the block introduces more challenge in tracking embedded bits without the secret key. Moreover, this modification helps the original PVD to pass the histogram analysis attack. After rotation, the secret message will be embedded in a similar manner to the original PVD algorithm as described in Section 3.1. This procedure is repeated till the end of the image.

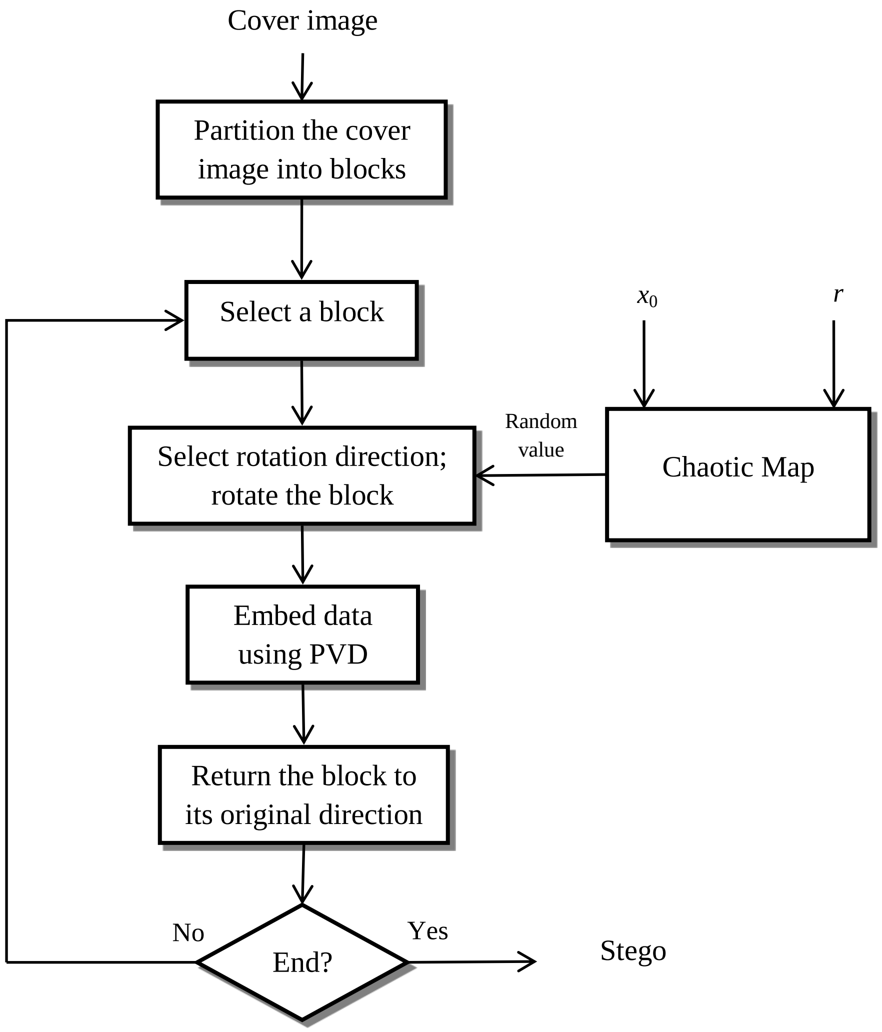

Cover image
Partition the cover image into blocks
Select a block
Select rotation direction; rotate the block
Random value
Chaotic Map
x_0
r
Embed data using PVD
Return the block to its original direction
End?
No
Yes
Stego

Figure 7. The embedding procedure flowchart.

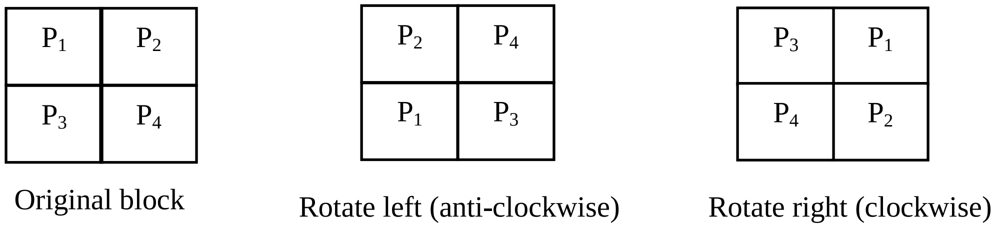


Figure 8. Block rotations.

4.1.3. PVD with Chaotic Block Rotation: Recovery

At the receiver side, only the control value r and the initial value x_0 are needed to recover the secret message. The recovery procedure will be similar to embedding but using reverse operations. Because of the stochastic nature of the sequence generated by the chaotic map that depends only on the initial condition and the control parameter, the prediction of the rotation directions will be a challenge. This adds a confusion level that makes the relationship between the embedded secret bits and their positions in the image pixels complex and unpredictable. Thus, the extraction of the embedded message will be more difficult for unauthorized persons. Furthermore, the proposed chaotic block rotation improves the histogram of the pixel-pair differences which has been found to be a good steganalytic tool for detecting the existence of embedded data. More details and experimental results will be discussed in Section 5.3.2.

4.2. Modulus Overlapping PVD Function

To further increase the embedding capacity of the pixel-value differencing method, we proposed the modulus overlapping pixel-value difference (MOPVD). This method utilizes the concepts of pixel-value differencing and overlapping to hide more secret data bits in a similar way to OPVD. It uses a sliding window of two pixels and determines the amount of secret bits to be embedded based on the difference between the pixels within the window. Embedding is then performed in the second pixel only. The window is shifted by one pixel; thus the second pixel in the previous window becomes the first pixel in the new window. The process is repeated until the end of the cover image. Unlike

OPVD, which skips too many pixels during embedding due to the out-of-range problem, see Section 3.1.F, the proposed MOPVD overcomes this problem. By adjusting the range of the new difference *d’* to fall in the same range as *d*, our method can utilize more unused pixels. This adjustment affects only the value of the pixel, but it does not change the value of the embedded data. Also the embedding procedure in our method differs from the one used in OPVD. OPVD uses LSB for embedding. However, our method embeds by modifying the remainder of the pixel-pair as we will discuss in the following Section. This has the advantage of avoiding the LSB security limitations and its noise on the stego-image histogram.

4.2.1. MOPVD with Chaotic Map: Embedding

To achieve good security against the histogram attack, we can combine the chaotic block rotation with the MOPVD method. We call this new method Chaotic MOPVD (CMOPVD). The embedding procedure of the proposed CMOPVD method is shown in Figure 9, and it can be described by the following steps:

- Apply the proposed chaotic block rotation.
- Each pixel-pair in the rotated block is modified separately. Assume the pixel-pair block F_i has pixels P_{ix} and P_{iy}, the following parameters are calculated: the difference $d_i = |P_{iy} - P_{ix}|$, the width of the range $w_i = u_i - l_i + 1$ where u_i and l_i are the range upper and lower bounds respectively, the number of secret bits to be embedded $n_i = log_2(w_i)$ and its equivalent decimal value is b_i, and the block remainder $F_{irem} = (P_{ix} + P_{iy}) mod\, 2^{n_i}$.

- Embed n_i bits of secret data into the second pixel P_{iy} such that $F'_{irem} = b_i$ as shown in equation (4.2).

$$P'_{iy} = \begin{cases} P_{iy} - m_1, & \text{if } F_{irem} > b_i \text{ and } m_1 \le 2^{n_i - 1} \\ P_{iy} + m_2, & \text{if } F_{irem} > b_i \text{ and } m_1 > 2^{n_i - 1} \\ P_{iy} + m_1, & \text{if } F_{irem} \le b_i \text{ and } m_1 \le 2^{n_i - 1} \\ P_{iy} - m_2, & \text{if } F_{irem} \le b_i \text{ and } m_1 > 2^{n_i - 1} \end{cases} \qquad 4.2$$

  where $m_1 = |F_{irem} - b_i|$, $m_2 = 2^{n_i} - |F_{irem} - b_i|$ and P'_{iy} is the value of the second pixel of the pixel-pair after embedding.

- Check the new difference of the pixel-pair after the embedding, $d' = |P'_{iy} - P_{ix}|$, to ensure that it is in the same range as the old difference d. If they are in different ranges, P'_{iy} is adjusted by adding or subtracting 2^{n_i} . This adjustment will return the value of d' to the same range of d without affecting the embedded secret bits.
- After this modification to preserve the same range, only few pixel values may fall out of the range (0, 255) which is not a proper gray level, these pixels are not used for embedding and they will be marked by moving them to the nearest limit of the range (0,255).

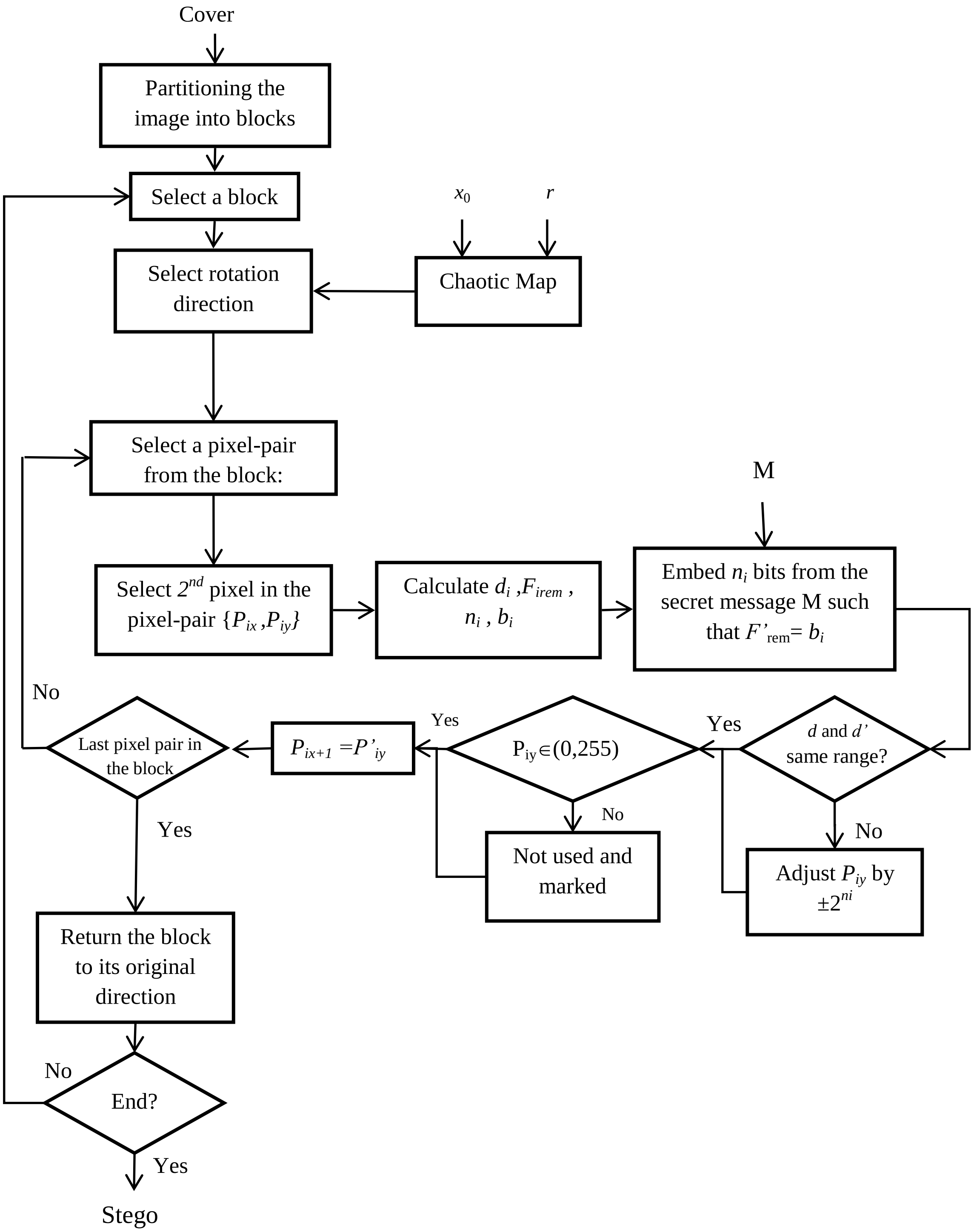

Cover
Partitioning the image into blocks
Select a block
x_0
r
Select rotation direction
Chaotic Map
Select a pixel-pair from the block:
M
Select 2^{nd} pixel in the pixel-pair $\{P_{ix}, P_{iy}\}$
Calculate d_i, F_{irem}, n_i, b_i
Embed n_i bits from the secret message M such that $F'_{rem} = b_i$
No
Last pixel pair in the block
$P_{ix+1} = P'_{iy}$
Yes
$P_{iy} \in (0,255)$
Yes
d and d' same range?
No
No
Yes
Not used and marked
Adjust P_{iy} by $\pm 2^{ni}$
Return the block to its original direction
No
End?
Yes
Stego

Figure 9. The embedding process of the CMOPVD method.

4.2.2. MOPVD with Chaotic Map: Extraction

The extraction of the secret bits from the block $F_i^{'}$ is straight forward. After rotating the stego-image blocks using the chaotic sequence, the extracted secret bits are the binary transformation of the pixel block remainder $F_{irem}^{'}$.

4.3. Fuzzy Edge Detection Function

In this section, we will discuss the third function of our system which utilizes the Fuzzy Template Based (FTB) edge detector to identify the image edges more efficiently (For more details about FTB, see Appendix 1). The procedure of this function consists of two steps. Firstly, an edge image will be generated using the FTB edge detector. Then the edge information and the secret data will be embedded into the cover image to generate the stego-image. Similar to [55] which was discussed in Section 3.2.B, the proposed method embeds the edge information into the stego image. However, in contrary to [55] which stores information about edge and smooth pixels, the proposed method stores only information about the edge pixels. The proposed method stores this information in the first pixels from every row of the stego image. Storing only edge information will decrease the number of used pixels to embed this information. Moreover, to embed the secret data in the remaining pixels, the proposed method uses a modified version of MOPVD function. Unlike the MOPVD, the modified MOPVD does not use the pixel-pair difference to detect the edge pixels. It depends on the edge information from the edge image that was generated using FTB edge detector. But similar to the MOPVD, the modified MOPVD embeds the secret using the overlapping pixel concept with modulus. The combination of FTB and modified MOPVD functions is denoted as E-MOPVD.

Similar to other data hiding methods, the proposed E-MOPVD method consists of two procedures: embedding and extracting procedures.

4.3.1 E-MOPVD Embedding Example

Assume a cover-image of size 512×512 and the corresponding edge image are as shown in Figure 10. The locus of the edge pixels in each row will be stored in the first few pixels of the cover image of the same row which is also illustrated in Figure 10.

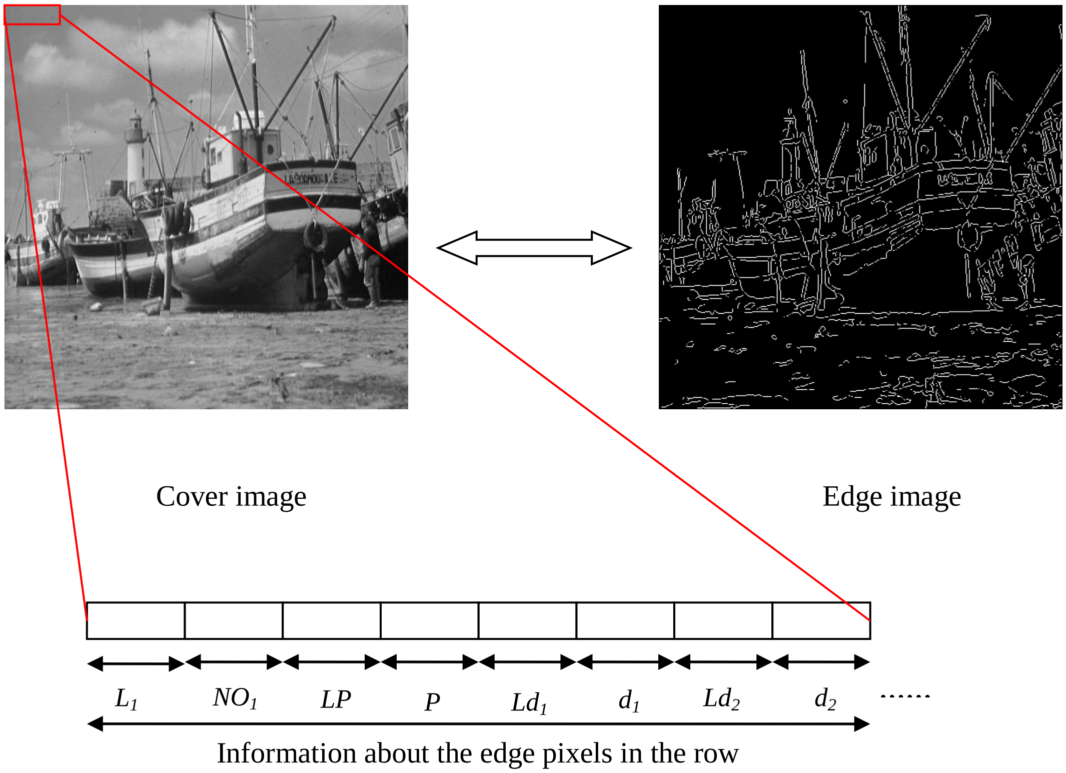


Figure 10. The storing process of edge information.

The meanings of the notations used in that figure are as follows:

- L_1: The digits length for the number of edges in the image row. Since the size of the image is 512×512, we used only two bits from the LSB of the first pixel from each row to represent this variable.
- NO_1: The number of edges in the selected row. Each digit of this number will be stored in the LSB of one pixel. The number of used pixels for this part will vary from one to three pixels. We used four LSB bits from each pixel to store a single digit.
- LP: The length of digits of the first edge position in the selected row. This length will be represented by the LSB of one pixel only.
- P: The first edge position in the selected row. Each digit will be represented by the LSB of one pixel only.
- Ld_1: To identify the next edge position in the same row, the difference (dif_1) between the first edge position and the second edge position is calculated. Ld_1 is the length of digits of this difference. We used only the LSB of one pixel for this value.
- d_1: The value of the dif_1 difference. Depending on the number of digits, the number of pixels will be used for embedding this value will vary from one to three pixels.
- Ld_2: The length of the digits of the difference (dif_2) between the second edge position and the third edge position.

- d_2: The value of the dif_2 difference. Depending on the number of digits, the number of pixels will be used for embedding this value will vary from one to three pixels.

In the case of consecutive edge pixels, the LSBs of the *Ld* part will be zeros. This means that the selected pixel represented a consecutive edge pixel position. Using this mechanism, we can represent the consecutive edges by constant overhead which is one pixel for each edge. Embedding is then performed for the secret message utilizing the edge information and using the MOPVD procedure as follows:

- Embed three secret bits in the LSB of the first pixel after the pixels that store the edge information.
- This pixel will be the first pixel in the first pixel-pair block which used to store the secret bits. Use the edge information to know the status of the pixels in this pixel-pair block.
- Based on the status of the pixels, the number of secret bits will be embedded in the second pixel using the modified MOPVD version. The numbers of embedded bits are illustrated in Table 1.
- Then the second pixel of the first block will be the first pixel in the second block.

Notice that LSB method is only used for one pixel, because the MOPVD procedure starts storing information from the second pixel of the pixel block.

Table 1. Numbers of embedded bits using the edge detection with MOPVD

First pixel status	**Second pixel status**	**Number of secret bits**
Smooth	Smooth	Three bits
Smooth	Edge	Five bits
Edge	Edge	Six bits

4.3.2 E-MOPVD Extracting Procedure

The receiver will read the edge information from the reserved first pixels in every row. Then, it reads three secret LSB bits from the first pixel, after that it uses this pixel as the first pixel in the pixel-pair and extracts the information using the MOPVD extraction mechanism. For example, assume we have 3 edge pixels in the first row. These pixels are located in the pixels 70, 75 and 99. Table 2 illustrates the process of storing edge information in the image pixels.

Table 2. Storing edge information in image pixels.

Variable	Value	Pixels
L_1	1	LSB of P_1
NO_1	3	LSB of P_2
LP	2	LSB of P_3
P	7	LSB of P_4
P	0	LSB of P_5
Ld_1	1	LSB of P_6
d_1	5	LSB of P_7
Ld_2	2	LSB of P_8
d_2	1	LSB of P_9
d_2	4	LSB of P_{10}

CHAPTER 5

EVALUATION AND COMPARISONS

In this chapter, we will conduct several experiments to evaluate and compare the proposed steganographic system with the existing methods in the literature. The evaluation and comparison will be in terms of capacity, invisibility and security.

5.1. Evaluation Criteria

In order to evaluate and compare the performance of steganographic methods, three common evaluation criteria are used. These criteria are payload capacity, invisibility, and security of the stego images.

A) Payload Capacity

This measure assesses how much of the secret data can be embedded into the cover image without jeopardizing the quality of the cover image. A good steganographic method should have high payload capacity. In this study, the payload capacity is measured by bits. In addition, we will also count the number of unused pixels to measure the wasted capacity of some steganographic methods.

B) Invisibility

Unlike data encryption, the changes made to the cover image by the embedding procedure of the steganographic method should not be observable by human eyes. In other words, a good steganographic method should not have any visual artifacts in the stego-image. This criterion is also known as imperceptibility. Among the measures that are commonly used to assess invisibility are the peak signal-to-noise ratio (PSNR), the weighted peak signal-to-noise ratio (WPSNR) and the structural similarity (SSIM) index. These measures assess the perceptual distortion caused to the image as a result of the embedding process. The higher the values of these measures are, the closer the stego-image is to the cover image.

The PSNR between a cover image X and its corresponding stego-image Y is calculated from:

$$PSNR(X,Y) = 10 \times \log_{10} \frac{[\max_{ij}(x_{ij})]^2}{MSE(X,Y)} dB \qquad 5.1$$

where dB is the decibel unit and $MSE(X, Y)$ is the mean square error which is calculated as follows:

$$MSE(X,Y) = \left(\frac{1}{m \times n}\right) \sum_{i=0}^{m-1} \sum_{j=0}^{n-1} \left(x_{ij} - y_{ij}\right)^2 \qquad 5.2$$

where $m \times n$ represents the size of each image, and x_{ij} and y_{ij} are the pixels in the cover image and stego image at location (i, j). We also computed a special value for PSNR assuming the maximum pixel value to be 255 and named it maximum PSNR (denoted as MPSNR).

The WPSNR improves the classical PSNR by taking into account the human visual system (HVS) characteristics. We used the code available at [64] which computes WPSNR using the contrast sensitivity function (CSF) to weight spatial frequency of error image [65].

The last quality measure that we have used is a relatively recent measure known as structural similarity (SSIM) index [66], [67]. This new similarity metric focuses on the similarity of structural information instead of the pixel-based comparison. It is computed from:

$$SSIM\left(X,Y\right)=\left(\frac{2\mu_X\mu_Y+C_1}{\mu_X^2+\mu_y^2+C_1}\right)\left(\frac{2\sigma_X\sigma_Y+C_2}{\sigma_X^2+\sigma_Y^2+C_2}\right)\left(\frac{\sigma_{XY}+C_3}{\sigma_X\sigma_Y+C_3}\right) \qquad 5.3$$

where μ_X and μ_Y represent the sample means of X and Y, σ_X and σ_Y represent the sample standard deviations of X and Y, and σ_{XY} denote the sample cross correlation between X and Y after removing their means. The constants C_1, C_2, and C_3 are small positive values that stabilize each term to avoid numerical instability caused by near zero sample means, variances, or correlations. Similar to the original paper [66], C_3 is set to $C_2/2$ to simplify the above equation to:

$$SSIM\left(X,Y\right)=\left(\frac{2\mu_X\mu_Y+C_1}{\mu_X^2+\mu_y^2+C_1}\right)\left(\frac{2\sigma_{XY}+C_2}{\sigma_X^2+\sigma_Y^2+C_2}\right) \qquad 5.4$$

C) Security

Steganography may be vulnerable to different attacks such as visual attacks and histogram attacks. Security criterion is used to assess the robustness of an information hiding method against each attack. In experimental work, we will evaluate security of several methods using histogram attacks. We used histogram attacks since almost all of the compared methods use the pixel-pair difference concept which affects the histogram of the pixel-pair difference as we discussed in Section 3.1.A. The histogram attack includes: pixel-pair difference histogram attack, Fourier attack, and image histogram attack.

5.2. Test Images

Two benchmark image dataset collections are used as cover images in our experimental work. The first dataset consists of ten 512×512 gray-level test images: Tank, Plane, Elaine, Car, Bridge, Aerial, Boat, Lena, Peppers and Baboon. These images are commonly used in many publications on image processing, image compression and steganography. Figure 11 illustrates a sample of images from the first dataset. The second dataset is the Uncompressed Color Image Database (version 2) (UCID) [68], [69]. It was initially created for the purpose of content-based image retrieval and later used in a number of papers on steganography and steganalysis such as [70], [71]. It has a total of 1338 uncompressed TIFF images including indoor and outdoor on a variety of topics such as natural scenes and man-made objects. Examples of the images included in this dataset are shown in Figure 12. This dataset has a large number of images of different sizes. To standardize the size of images and to speed up processing, all images in the UCID dataset are resized to 384×512. Then, the images are converted into gray-level and saved as bitmap images before the experiments.

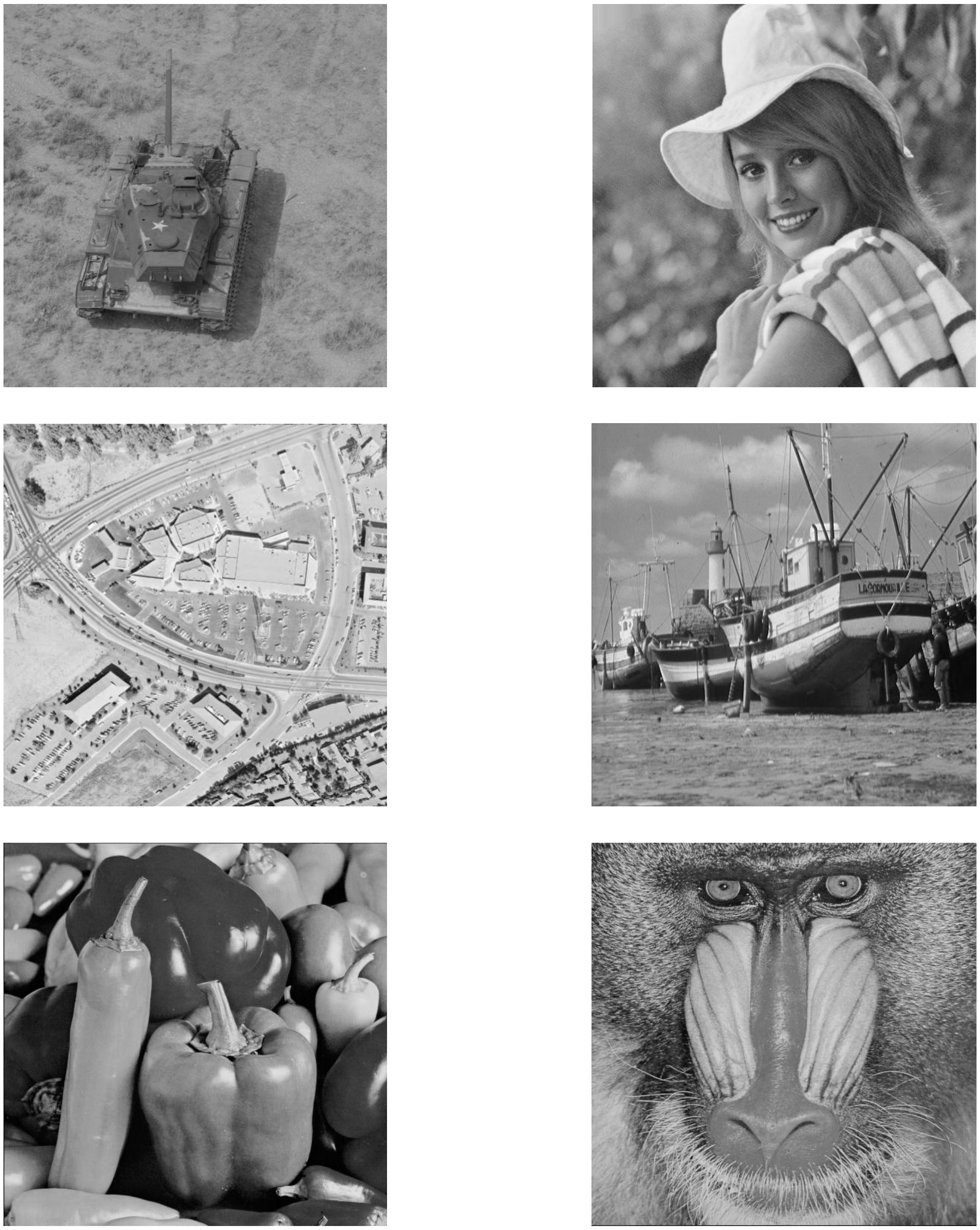


Figure 11. Sample of the first dataset test images.

Figure 12. UCID dataset samples.

5.3. Experimental Work

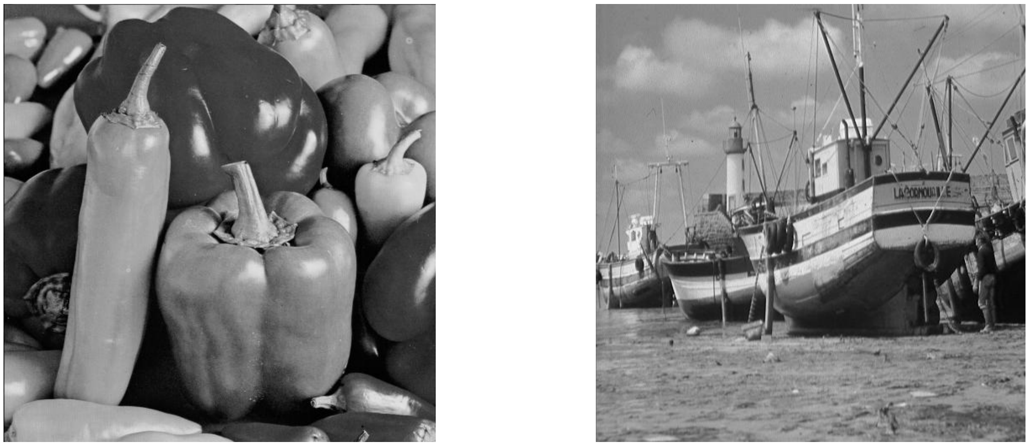
In this section, we will discuss the experimental work and results of the proposed steganographic system components. The conducted experiments include testing each component of the proposed system separately and altogether to evaluate their effectiveness. We will also compare the results with several existing steganographic methods from the literature. All methods are implemented in MATLAB Release R2010a [72]. In all experiments, the same secret message is randomly generated and embedded into cover images to generate the stego images. Moreover, the first benchmark image dataset will be the main dataset in evaluating the capacity, invisibility and security against the histogram attacks. On the other hand, the UCID dataset will be mostly used when comparing the performance of the proposed edge steganographic method with different edge detection methods. First of all, we will compare the resulted stego images for several methods to prove that these images have no visual artifacts. Then, we will calculate the embedding capacity and number of unutilized pixels to evaluate the capacity performance. The PSNR, MPSNR, WPSNR and SSIM will be used as the quality performance measures. Moreover, in our experiments we will apply the histogram attacks to evaluate the security of the steganographic methods.

5.3.1. Capacity and Invisibility Evaluation

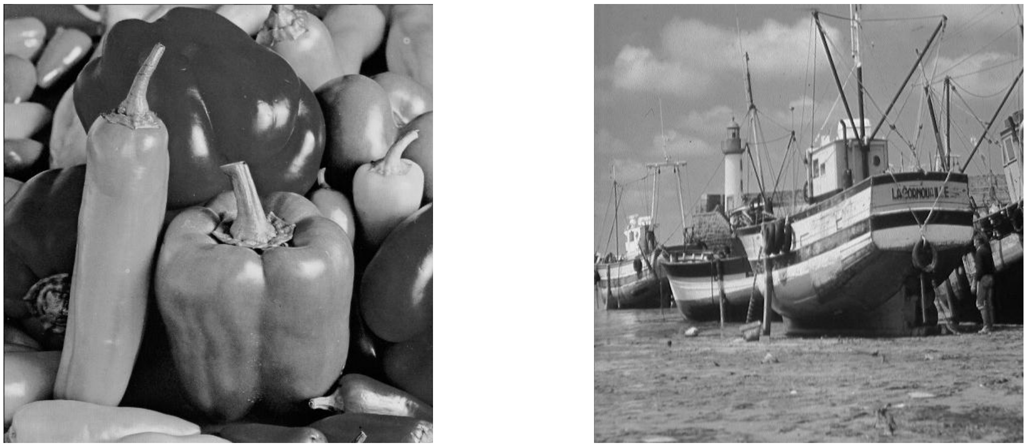
In this section, we report the results of several experimental tests to evaluate and compare the capacity and invisibility of the proposed system functions (MOPVD, CMOPVD, and E-MOPVD). There are three parts. In the first part, we compare the capacity and invisibility of the proposed system functions with existing methods (PVD, OPVD, Modulus PVD, MPD, PVD+LSB, Side-Match, and Tri-way PVD). The second part discusses the effect of different parameters of the proposed system on capacity and security of this system. The third part compares the proposed E-MOPVD with the HP method in [55] under the same conditions.

A) Capacity and Invisibility Comparison

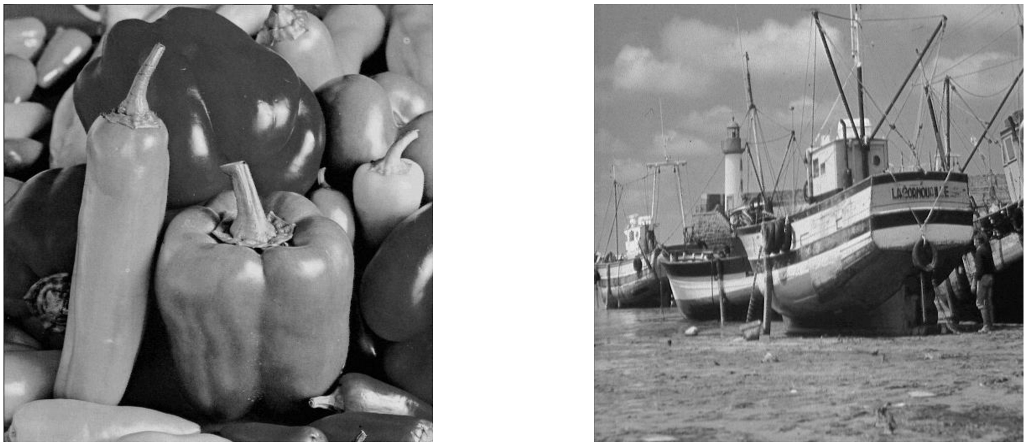
Figure 13 compares the cover images and the stego-images for several steganographic methods. It is observed that there are no obvious visual artifacts found in the stego-images. Therefore, the stego-images cannot be identified easily by human eyes.



Cover image (Peppers) Cover image (Boat)

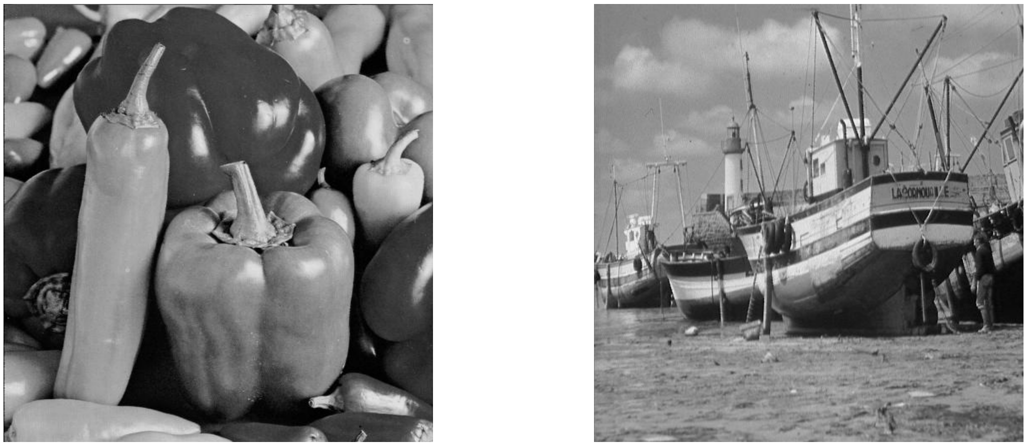


PVD stego images

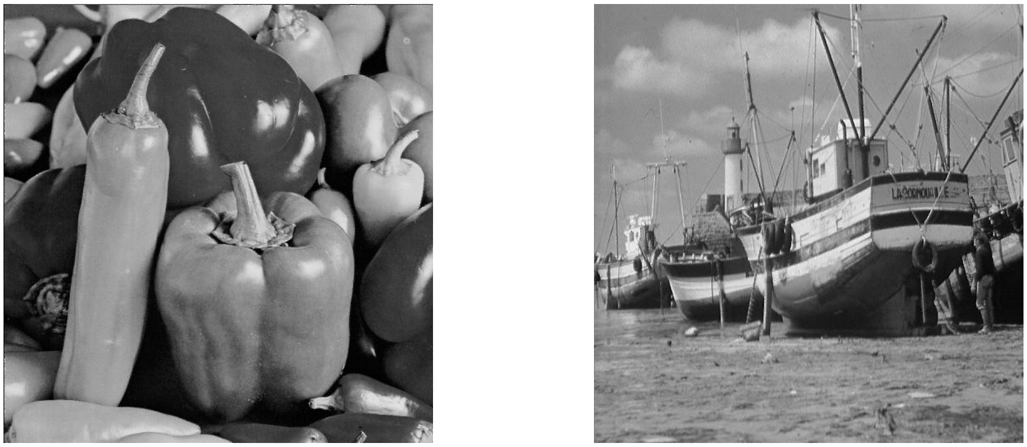


OPVD stego images

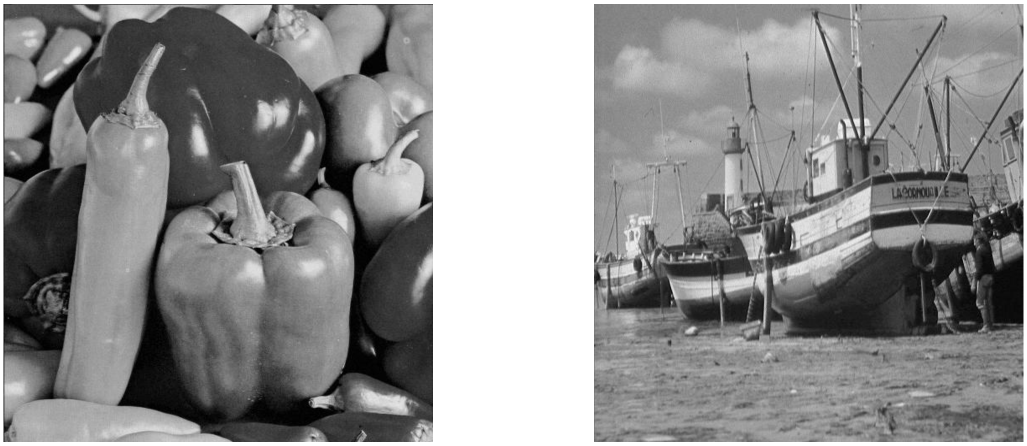
Figure 13. Visual comparision of cover and stego images (peppers and boat images).



Modulus PVD stego images

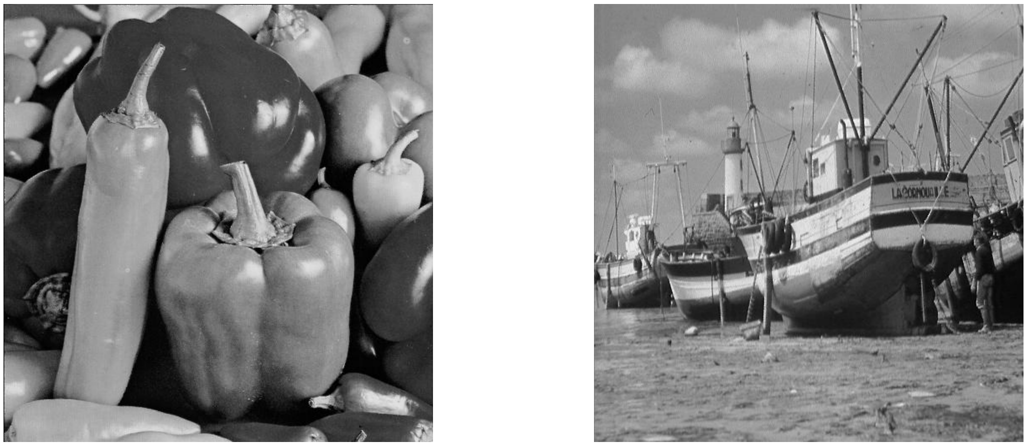


MPD stego images

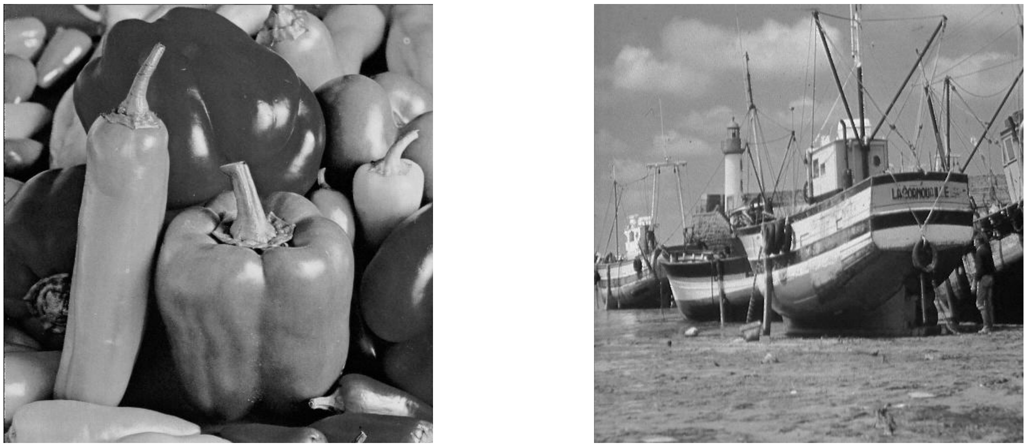


PVD+LSB stego images

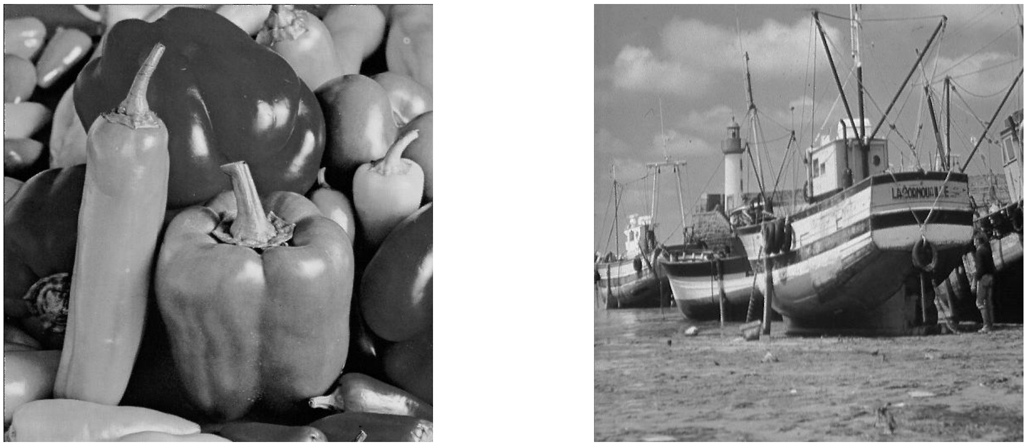
Figure 13 (Cont.). Visual comparision of cover and stego images (peppers and boat images).



Side-Match stego images

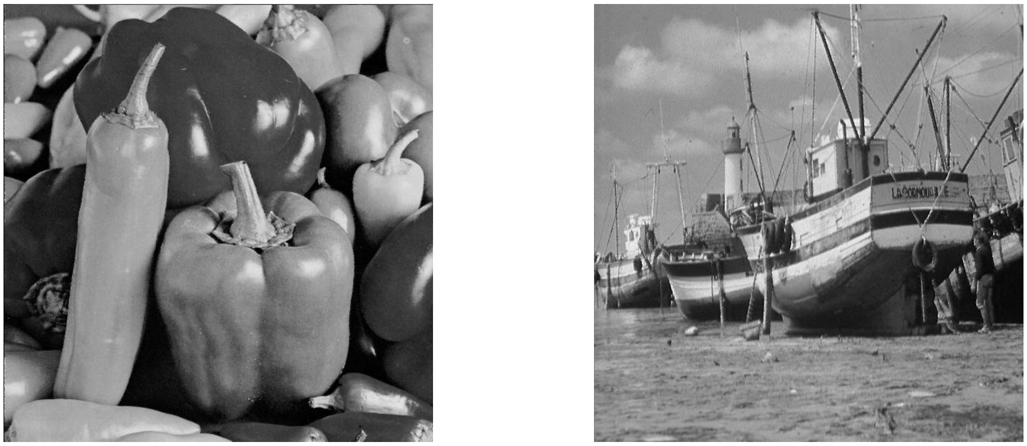


Tri-way PVD stego images

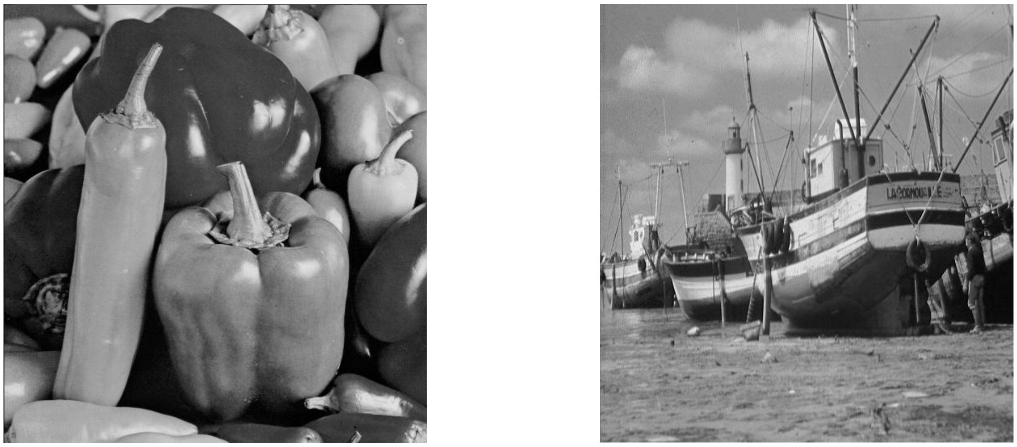


MOPVD stego images

Figure 13 (Cont.). Visual comparision of cover and stego images (peppers and boat images).



CMOPVD stego images



E-MOPVD stego images

E-MOPVD with rotation stego images

Figure 13 (Cont.). Visual comparision of cover and stego images (peppers and boat images).

Table 3 shows the capacity comparison in bits for several steganographic methods using the first image dataset described in Section 5.2. The average embedding capacities are shown in Figure 14. It can be noticed from this figure that the PVD and the modulus methods have the lowest embedding capacities. We can also notice that the proposed CMOPVD and MOPVD methods have comparably high embedding capacities. These results are due to the utilization of each pixel individually in the embedding process and due to the reduction of the number of unutilized pixels. On the other hand, the proposed E-MOPVD has less embedding capacity than the MOPVD due to the utilization of some pixels for storing the edge information. Although its capacity is a bit lower than PVD+LSB, the latter method has similar characteristics to the LSB more than the PVD characteristics as explained in Section 3.1.B. This can be demonstrated through Figure 15, where the number of LSB operations is much more than the number of PVD operations in the PVD+LSB method. Since most of the image areas are smooth, the pixel-pair difference is small. Consequently, the number of LSB operations is very high in the PVD+LSB.

Table 3. Comparing the capacity for different methods

Images	**Methods**									
	Modulus	**MPD**	**OPVD**	**PVD+LSB**	**PVD**	**Tri-way**	**Side-Match**	**MOPVD**	**CMOPVD**	**E- MOPVD**
Tank	403990	577224	569546	777582	403990	601090	484284	810471	829009	756676
Plane	397911	416953	691264	784441	397904	592366	323841	794828	799570	769262
Elaine	408594	597740	540597	773285	408582	601665	530462	820296	836900	765016
Car	400521	576218	599239	779871	400504	601874	453488	801871	830299	739924
Bridge	446618	613575	507228	755148	442290	625310	654123	884191	922234	738624
Aerial	432439	648951	558191	766311	430783	614761	551471	863764	906991	740942
Boat	421083	613182	549603	770337	419317	607517	525530	844988	853903	757362
Lena	409810	578463	590087	776078	409804	600087	428690	820123	833076	763429
Peppers	407643	585548	587361	778277	402552	599329	455900	810493	824475	753921
Baboon	457105	723909	477795	751735	456867	627616	725997	920830	981513	730115

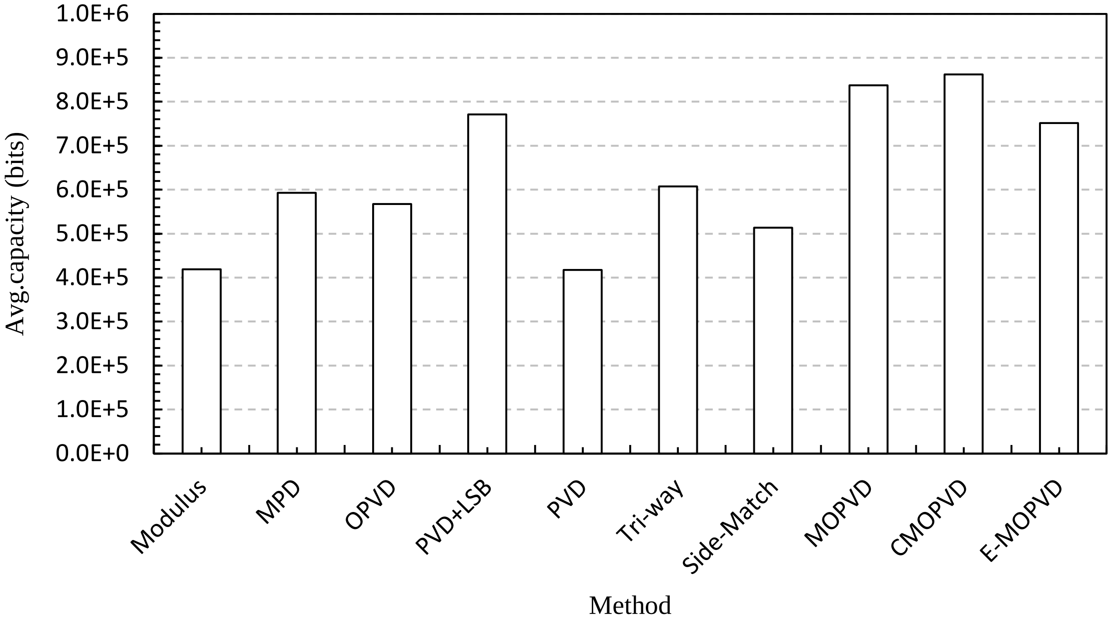


Figure 14. The average capacities of methods using the first image dataset.

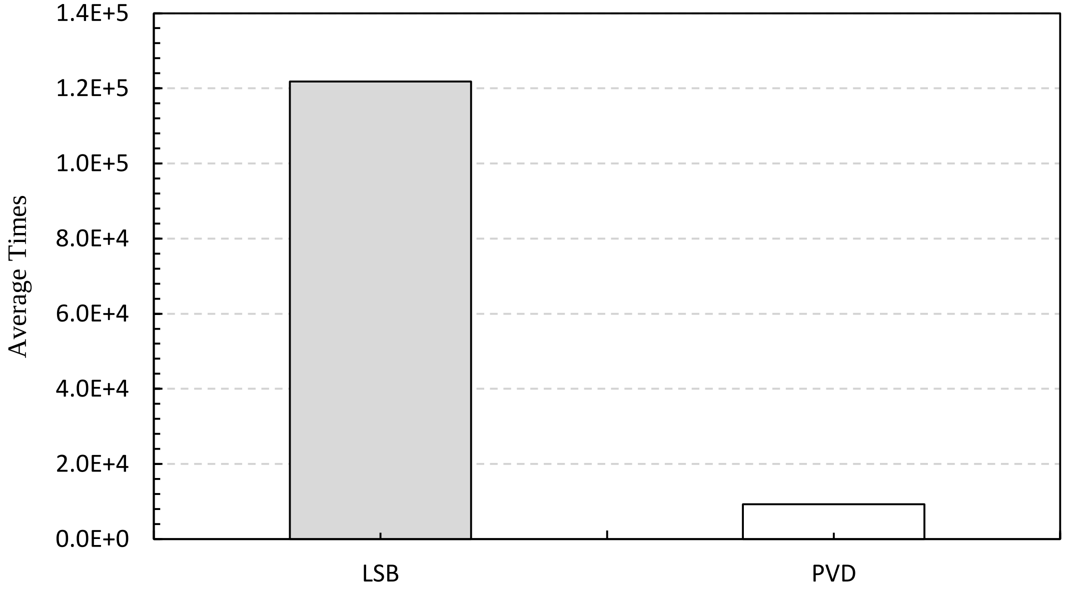


Figure 15. The number of LSB and PVD operations in PVD+LSB method.

In the following test, we will compare the number of unused pixels in the proposed MOPVD and CMOPVD with the OPVD method. The first dataset is used in this test. Figure 16 shows that the proposed MOPVD with and without block rotation can reduce the problem of unused pixels in the OPVD method significantly. Because of the large variation between the numbers, the scale in Figure 16 is changed to logarithmic scale.

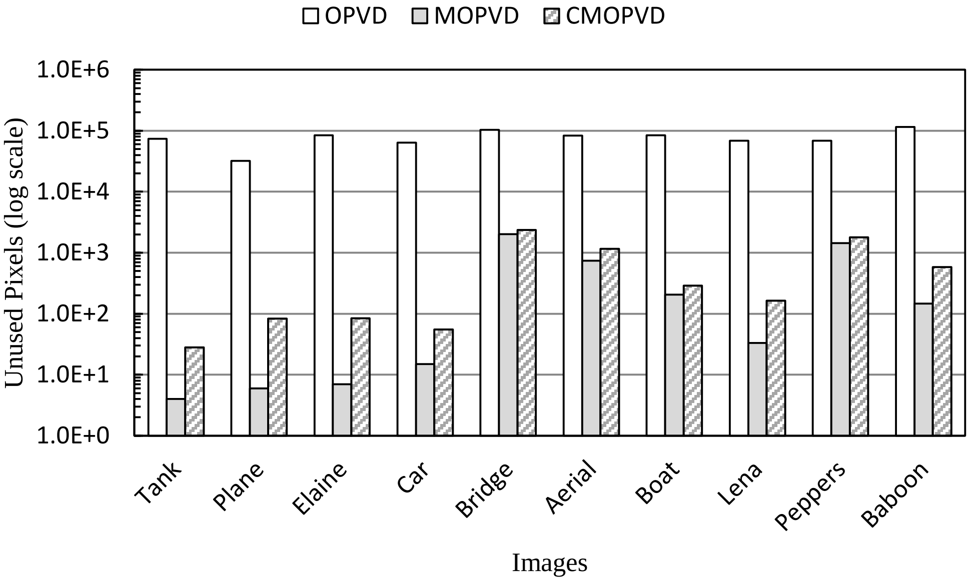


Figure 16. Number of unused pixels for the OPVD, MOPVD and CMOPVD.

Tables 4, 5 and 6 illustrate several comparisons of ten steganographic methods using the PSNR, MPSNR and WPSNR respectively. We can notice that the PVD and modulus have the highest PSNR. This is because their embedding mechanisms have lowered the embedding capacities, as illustrated before in Figure 14. The averages for the PSNR, MPSNR and WPSNR over the ten images are shown in Figure 17. Although the proposed CMOPVD has the lowest average PSNR, the average PSNR is still above 30 dB which is the acceptable quality to the human eyes without noticing any distortions in the sego image. Despite the proposed E-MOPVD and MOPVD has the same PSNR, the E-MOPVD improved the WPSNR more than the MOPVD. The WPSNR takes into account the human visual sensitivity system. Therefore, the proposed E-MOPVD improves invisibility of the MOPVD stego-image. Figure 18 shows a trade-off between average capacity and average PSNR for steganographic methods. We can notice that one of the proposed methods (namely CMOPVD) has the highest average embedding capacity but the lowest average PSNR (yet it is still higher than 30dB). In addition, it has good security characteristics due to the chaotic block rotation, as it will be shown later in the next section.

We also computed the SSIM measure for the ten methods for the ten test images and the results are shown in Table 7 and the corresponding trade-off curve between the average SSIM and average capacity is shown in Figure 19. We can notice that the proposed methods have higher average capacities with a slight degradation in the SSIM which would not be clear to the human eye.

Table 4. Comparing the PSNR for different methods.

Images	**Methods**									
	Modulus	**MPD**	**OPVD**	**PVD+LSB**	**PVD**	**Tri-way**	**Side-Match**	**MOPVD**	**CMOPVD**	**E- MOPVD**
Tank	43.99	37.08	37.97	36.38	41.22	37.07	38.74	35.64	34.72	35.64
Plane	45.03	39.75	37.95	37.38	42.00	39.09	41.52	38.07	36.79	38.07
Elaine	44.41	37.42	38.96	37.11	41.49	37.72	38.36	37.57	34.69	37.57
Car	45.44	38.16	38.93	37.45	42.68	38.06	40.39	36.89	35.63	36.89
Bridge	41.00	33.15	37.21	36.83	37.67	35.77	34.03	36.81	30.18	36.81
Aerial	41.54	33.01	37.05	36.86	38.45	36.35	34.76	36.23	29.93	36.23
Boat	42.06	35.58	37.94	37.07	39.50	36.67	37.24	36.04	32.65	36.04
Lena	43.63	37.24	37.99	37.11	40.78	37.75	39.23	36.32	33.94	36.32
Peppers	42.68	35.74	37.65	36.55	40.53	36.54	37.96	36.51	33.26	36.51
Baboon	39.28	29.90	35.53	35.39	36.07	34.06	31.06	35.80	27.18	35.80

Table 5. Comparing the MPSNR for different methods.

Images	Methods									
	Modulus	**MPD**	**OPVD**	**PVD+LSB**	**PVD**	**Tri-way**	**Side-Match**	**MOPVD**	**CMOPVD**	**E-MOPVD**
Tank	45.15	38.24	39.13	37.54	42.38	38.24	39.90	37.55	35.88	36.81
Plane	45.20	39.92	38.12	37.55	42.17	39.26	41.70	38.31	36.96	38.24
Elaine	44.83	37.84	39.38	37.52	41.91	38.14	38.78	36.85	35.11	37.98
Car	45.61	38.33	39.10	37.63	42.85	38.24	40.56	38.53	35.80	37.06
Bridge	41.00	33.15	37.21	36.83	37.67	35.77	34.03	32.03	30.18	36.81
Aerial	41.54	33.01	37.05	36.86	38.45	36.35	34.76	32.72	29.93	36.23
Boat	42.06	35.58	37.94	37.07	39.50	36.67	37.24	33.31	32.65	36.04
Lena	43.97	37.58	38.34	37.46	41.13	38.10	39.58	35.97	34.28	36.66
Peppers	43.62	36.67	38.58	37.48	41.47	37.47	38.89	36.01	34.19	37.45
Baboon	40.17	30.80	36.43	36.29	36.97	34.96	31.96	30.82	28.08	36.70

Table 6. Comparing the WPSNR for different methods.

Images	Methods									
	Modulus	**MPD**	**OPVD**	**PVD+LSB**	**PVD**	**Tri-way**	**Side-Match**	**MOPVD**	**CMOPVD**	**E-MOPVD**
Tank	39.12	30.67	38.53	44.18	38.42	38.90	31.28	31.07	31.07	38.37
Plane	38.29	45.82	50.00	35.69	38.50	52.73	46.08	48.14	44.74	51.36
Elaine	35.07	37.24	45.38	42.36	57.12	36.44	37.51	36.99	36.96	36.43
Car	37.04	40.83	47.57	51.16	51.94	36.44	39.51	51.31	37.81	47.59
Bridge	53.56	53.63	51.48	52.16	58.98	51.10	48.82	47.52	45.67	50.86
Aerial	50.70	41.49	42.23	49.68	44.18	40.86	40.61	40.96	40.43	41.62
Boat	54.73	56.13	52.56	52.97	60.97	51.71	51.63	48.55	48.08	50.52
Lena	44.30	36.13	30.45	36.39	35.21	43.98	31.95	32.54	32.53	33.38
Peppers	37.68	31.60	49.12	49.43	40.12	31.63	31.74	31.48	31.48	33.83
Baboon	50.98	30.90	50.43	41.73	45.81	33.56	31.01	30.73	30.65	46.80

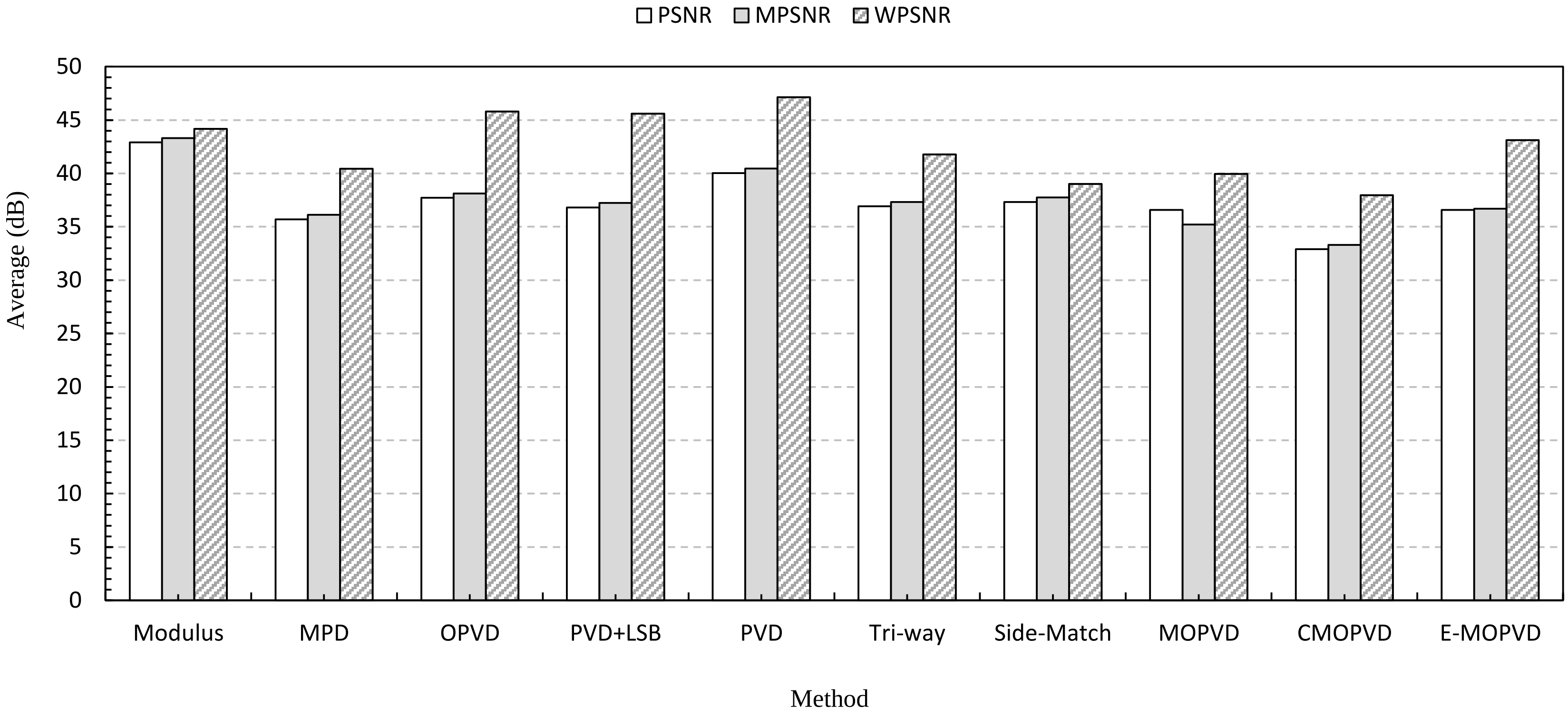


Figure 17. Average of PSNR, MPSNR and WPSNR.

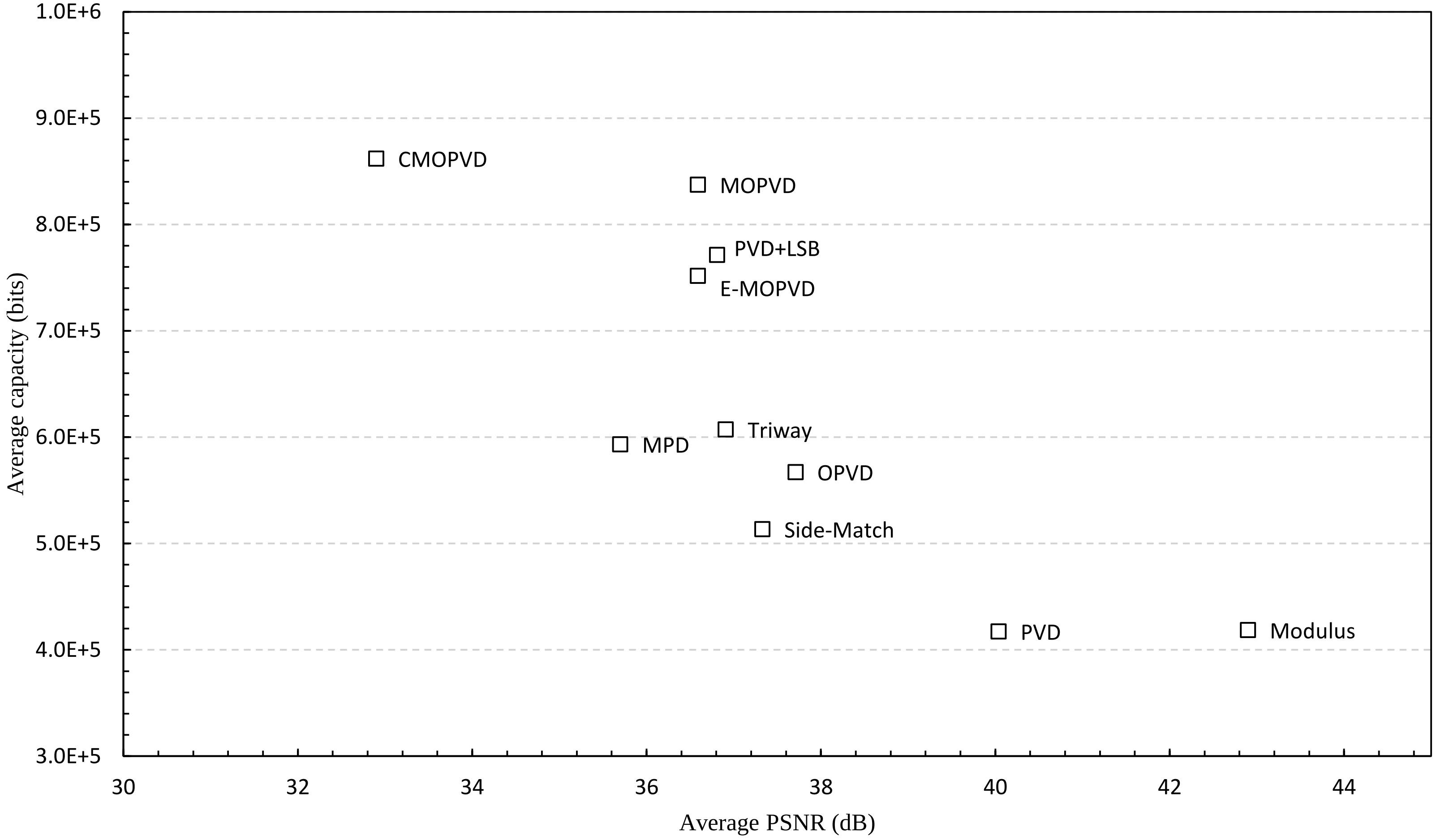


Figure 18. Comparing the average PSNR with the average capacity.

Table 7. Comparing the SSIM for different methods.

Images	Methods									
	Modulus	**MPD**	**OPVD**	**PVD+LSB**	**PVD**	**Tri-way**	**Side-Match**	**MOPVD**	**CMOPVD**	**E- MOPVD**
Tank	0.990	0.959	0.957	0.938	0.981	0.950	0.976	0.953	0.940	0.935
Plane	0.982	0.961	0.905	0.891	0.962	0.921	0.984	0.933	0.928	0.913
Elaine	0.990	0.955	0.957	0.936	0.980	0.949	0.968	0.947	0.934	0.949
Car	0.990	0.960	0.954	0.939	0.982	0.952	0.981	0.959	0.942	0.935
Bridge	0.993	0.968	0.978	0.972	0.985	0.971	0.974	0.957	0.939	0.967
Aerial	0.992	0.963	0.968	0.960	0.985	0.967	0.982	0.960	0.939	0.953
Boat	0.990	0.959	0.959	0.942	0.981	0.953	0.977	0.947	0.942	0.940
Lena	0.988	0.955	0.942	0.925	0.977	0.943	0.983	0.948	0.937	0.921
Peppers	0.988	0.954	0.944	0.925	0.978	0.942	0.978	0.949	0.938	0.932
Baboon	0.993	0.959	0.978	0.970	0.987	0.971	0.971	0.956	0.931	0.975

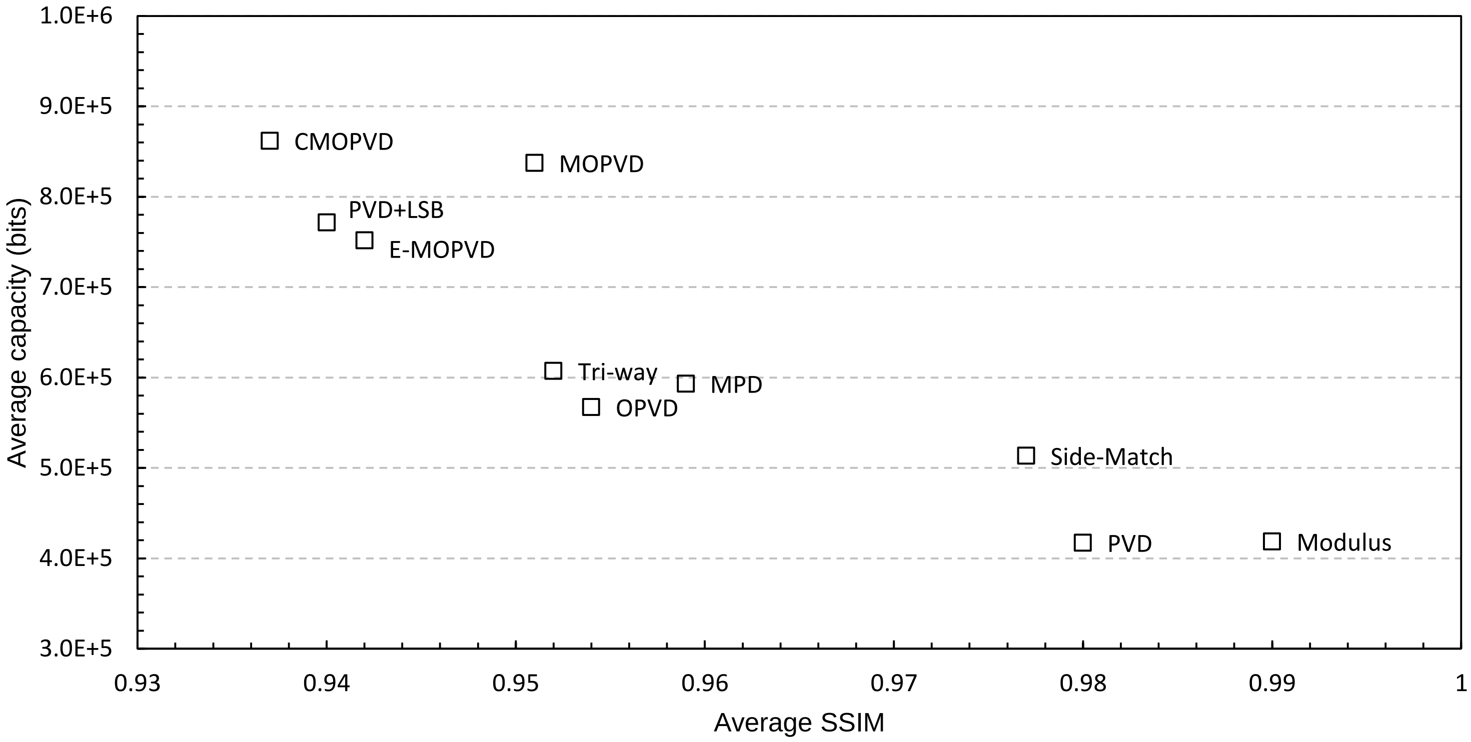


Figure 19. Comparing the average SSIM with the average capacity.

B) Impact of Chaotic Map

Here, we study the effect of the proposed chaotic block rotation component individually. We will apply this component to the original PVD. This component uses the logistic chaotic map to rotate each image block either clockwise or anti-clockwise as we discussed it in Section 4.1.1. Furthermore, we consider two sets of ranges for the PVD: {8, 8, 16, 32, 64, 128} and {2, 2, 4, 4, 4, 8, 8, 16, 16, 32, 32, 64, 64}, which have been used in the original PVD. The results in Tables 8 and 9 illustrate that changing the control value *r* does not affect the embedding capacity and the image quality. It only increases the variation in the block rotation which adds more complications in the secret extraction to the attackers. Moreover, the embedding capacity and the PSNR for the original PVD and the PVD with chaotic block rotation are almost the same. Only the changes in those values in Tables 8 and 9 come from the selection of different edges because of rotation. Table 10 illustrates the results of the same test, but when using the second set of ranges and full capacity.

Table 8. Capacity of PVD and modified PVD using first range set.

r=3.9 x_0=0.9	**Original PVD**		**Modified PVD**			
	Capacity (bits)	***PSNR (dB)***	***Capacity***		***PSNR***	
			bits	***%relative change***	***dB***	***%relative change***
Baboon	456867	36.94	477556	4.53%	35.08	−5.03%
Lena	409804	41.11	402605	−1.76%	42.29	2.87%
Peppers	402552	41.55	402244	−0.08%	41.67	0.29%

Table 9. Capacity of PVD and modified PVD with different chaotic parameters.

r=3.59 **x_0=0.9**	**Original PVD**		**Modified PVD**			
	Capacity (bits)	***PSNR (dB)***	***Capacity***		***PSNR***	
			bits	***%relative change***	***dB***	***%relative change***
Baboon	456867	36.94	477556	4.53%	35.09	−5%
Lena	409804	41.11	402605	−1.76%	42.34	2.99%
Peppers	402552	41.55	402244	−0.08%	41.70	0.36%

Table 10. Capacity of PVD and modified PVD using the second range set.

r=3.9 **x_0=0.9**	**Original PVD**		**Modified PVD**			
	Capacity (bits)	***PSNR (dB)***	***Capacity***		***PSNR***	
			bits	***%relative change***	***dB***	***%relative change***
Baboon	297442	43.29	322883	8.55%	40.95	−5.41%
Lena	213626	47.87	196209	−8.15%	49.57	3.55%
Peppers	214997	47.98	211611	−1.57%	47.83	−0.31%

We also study the effect of adding the chaotic block rotation to the proposed edge detection steganography. We will use the abbreviation E-CMOPVD for the proposed E-MOPVD with chaotic block rotation. The evaluation of the capacity and quality of this method is as shown in Tables 11; for the ease of reference we also added the performance of E-MOPVD to clearly see the impact (though, these metrics have been mentioned before). The table also shows the number of pixels used to store edge information and the number of out of range cases. It can be seen that the block rotation component does not significantly affect the capacity or the quality of the stego images.

Table 11. Performance of E-CMOPVD as compared to E-MOPVD.

Images	EMOPVD						E-CMOPVD					
	Capacity	MPSNR	PSNR	WPSNR	Pixels for edge info	Out range cases	Capacity	MPSNR	PSNR	WPSNR	Pixels for edge info	Out range cases
Tank	756676	36.81	35.64	38.37	20471	1	758477	36.90	35.73	42.79	19274	0
Plane	769262	38.24	38.07	51.36	11137	0	769482	38.03	37.85	50.68	11490	2
Elaine	765016	37.98	37.57	36.43	13210	6	764162	37.70	37.28	36.41	14281	13
Car	739924	37.06	36.89	47.59	25972	15	744874	37.05	36.87	50.28	23801	23
Bridge	738624	36.81	36.81	50.86	24624	1817	743264	37.10	37.10	51.17	22202	1673
Aerial	740942	36.23	36.23	41.62	28217	521	745898	36.34	36.34	43.76	25944	289
Boat	757362	36.04	36.04	50.52	21359	36	756141	36.51	36.51	50.88	21097	31
Lena	763429	36.66	36.32	33.38	17423	2	762367	36.67	36.32	34.29	18140	0
Peppers	753921	37.45	36.51	33.83	16616	1286	753047	37.26	36.32	32.34	17603	1291
Baboon	730115	36.70	35.80	46.80	30603	94	742260	37.03	36.14	45.72	24404	48

C) Effectiveness of the Proposed Edge Detection

Finally, we evaluate the proposed edge steganography and compare it with the high payload (HP) method [55] which uses the same traditional edge detection mechanism. To demonstrate that our method is much better than the one in [55], we evaluate both methods under the same embedding criteria and compare them using different edge detection mechanisms in terms of average embedding capacity and average PSNR. Our method is modified to embed only one bit in the LSB of the smooth pixel whereas three bits are embedded in the LSB of the edge pixel, which is the same number of embedded bits used by [55]. The UCID image dataset [69] is used in this experiment. Figures 20 and 21 show various comparisons. Obviously, the proposed method has achieved higher performance measures (both capacity and PSNR).

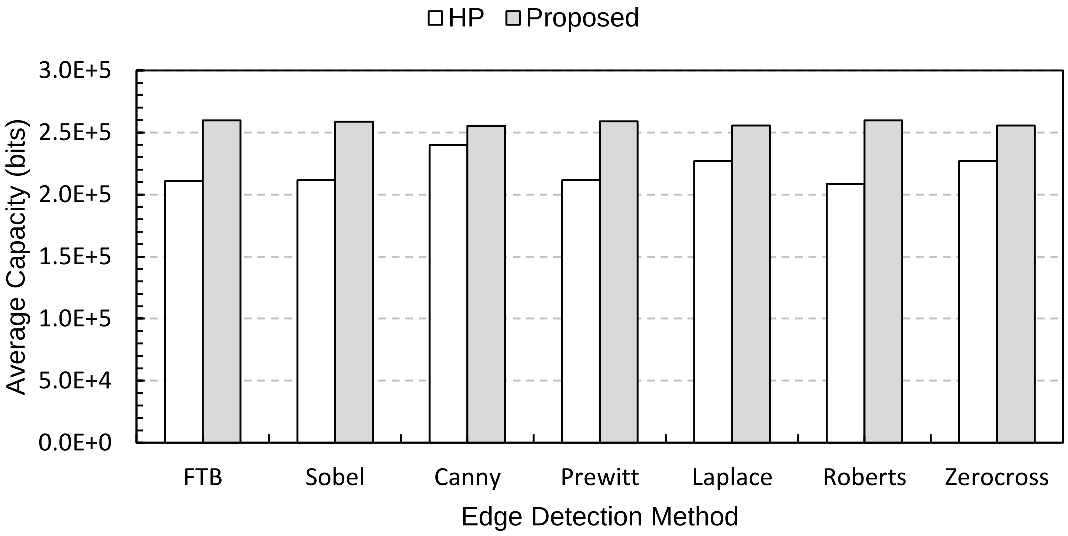


Figure 20. Comparing the capacity of HP and edge detection methods.

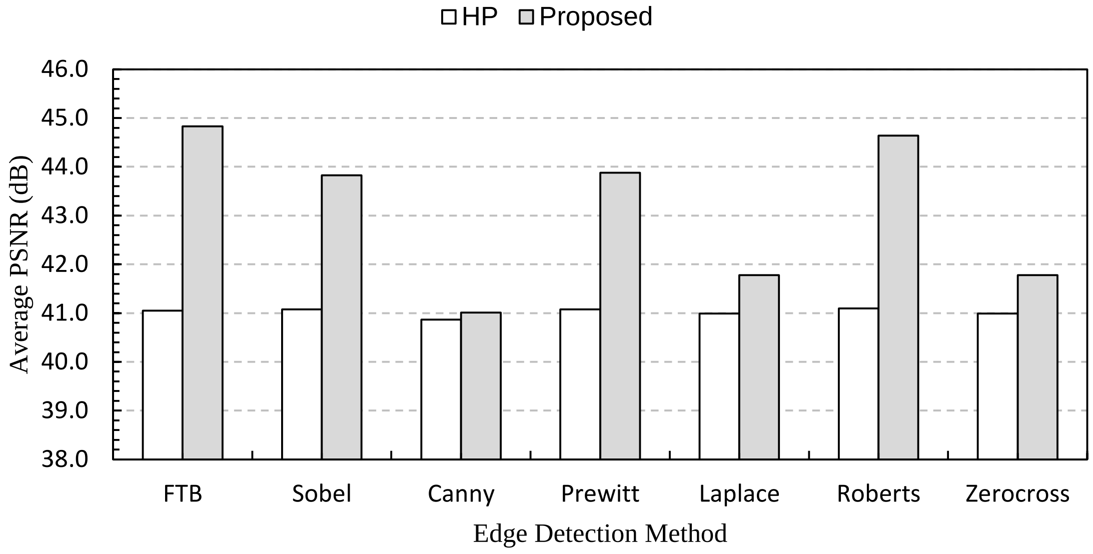
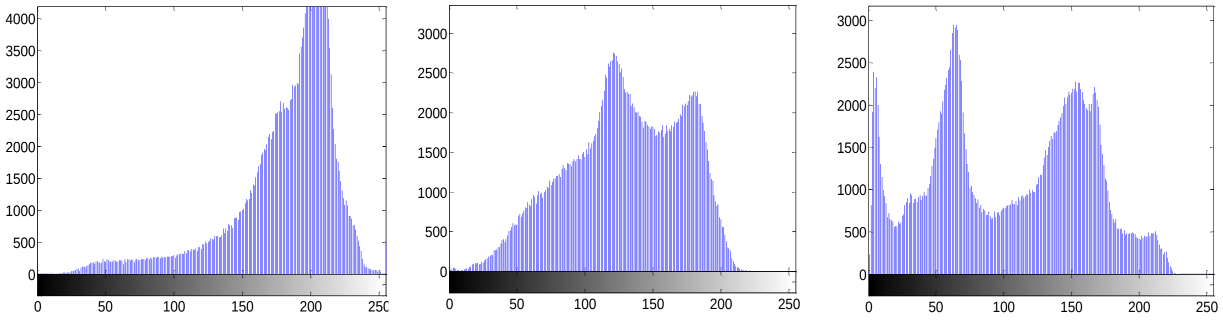


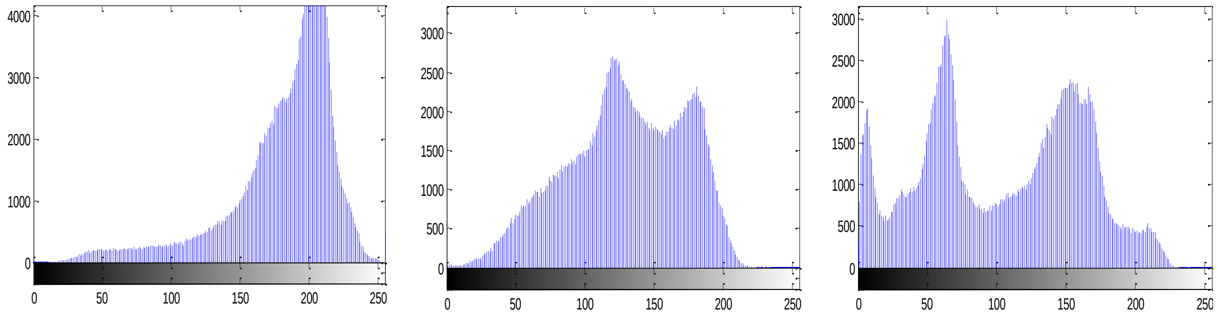
Figure 21. Comparing average PSNR of the HP and edge detection methods.

5.3.2. Security Evaluation

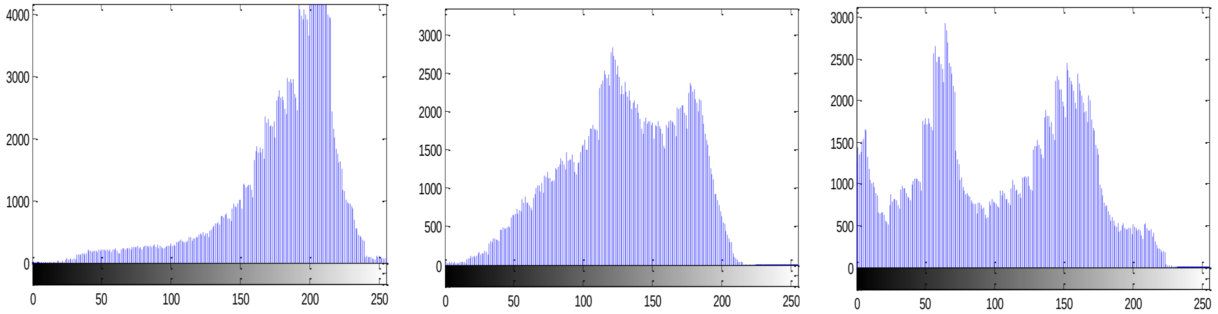
In this part of the experimental work, we will study the security of the proposed system functions (MOPVD, CMOPVD, E-MOPVD, and E-CMOPVD) and several existing methods (PVD, OPVD, Modulus PVD, MPD, PVD+LSB, Side-Match, and Tri-way PVD) by applying different histogram tests. Firstly, we run the image histogram test. In this test, the values of all image pixels are calculated, then the histogram for those values are drawn. Figure 22 shows the image histogram for the different steganographic methods.



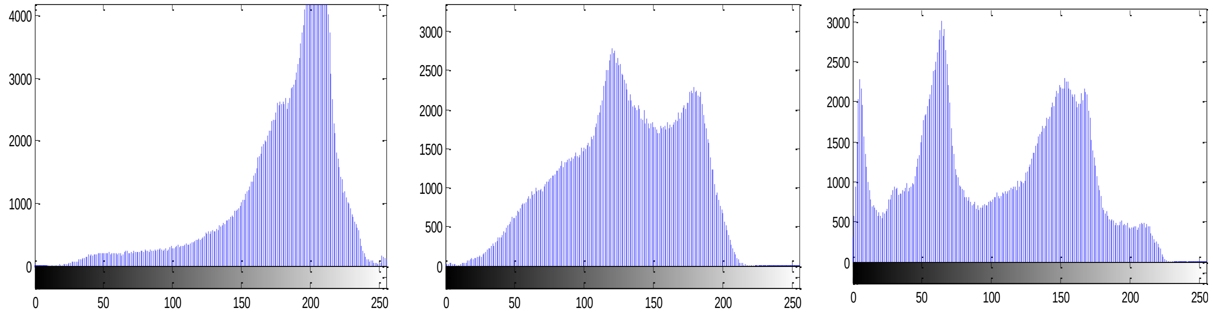
(a) Histogram of original images



(b) Histogram of PVD stego images

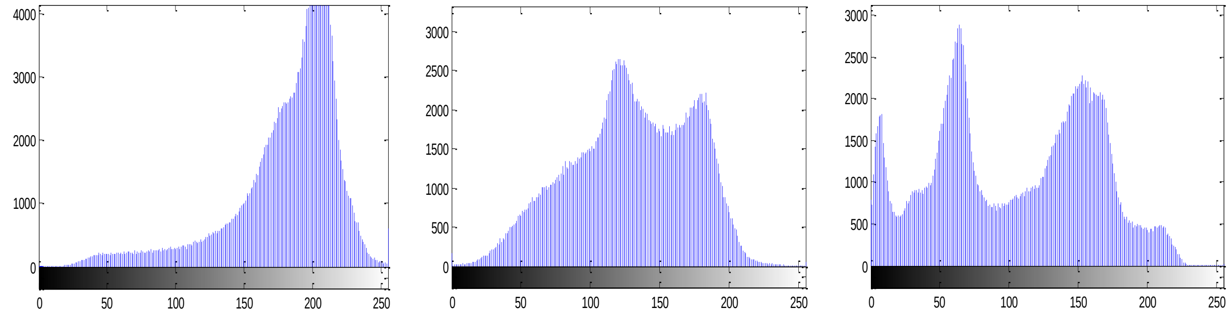


(c) Histogram of OPVD stego images

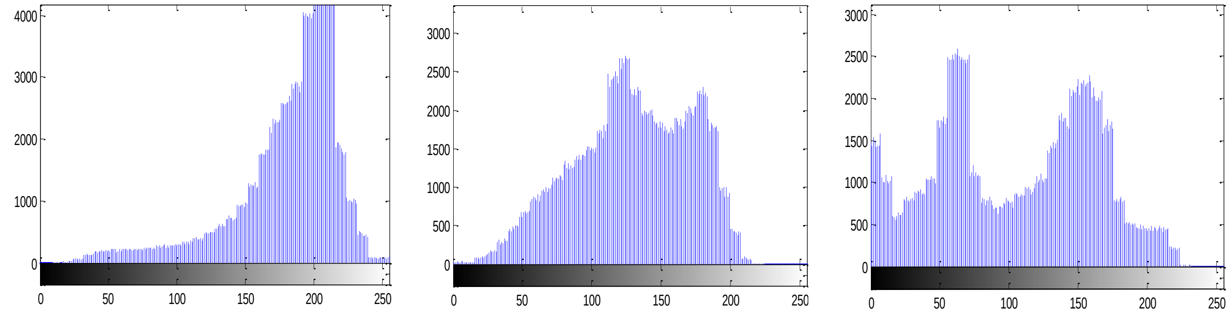


(d) Histogram of Modulus PVD stego images

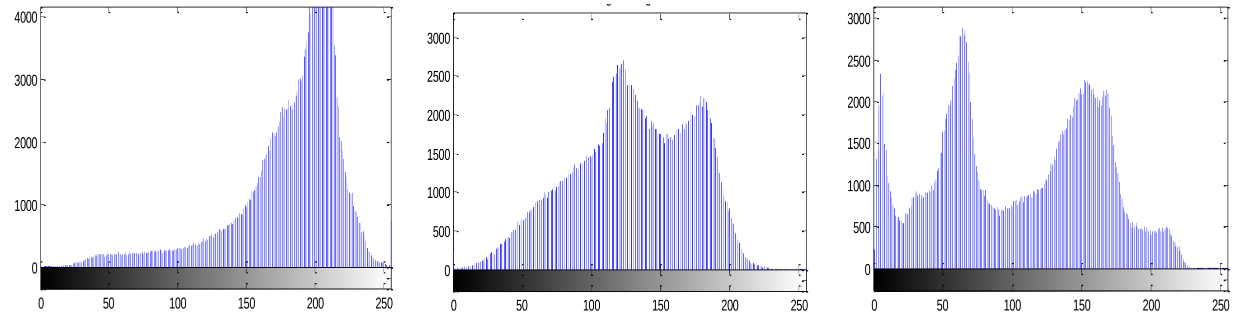
Figure 22. Image histogram test (aerial, baboon and peppers images from left to right).



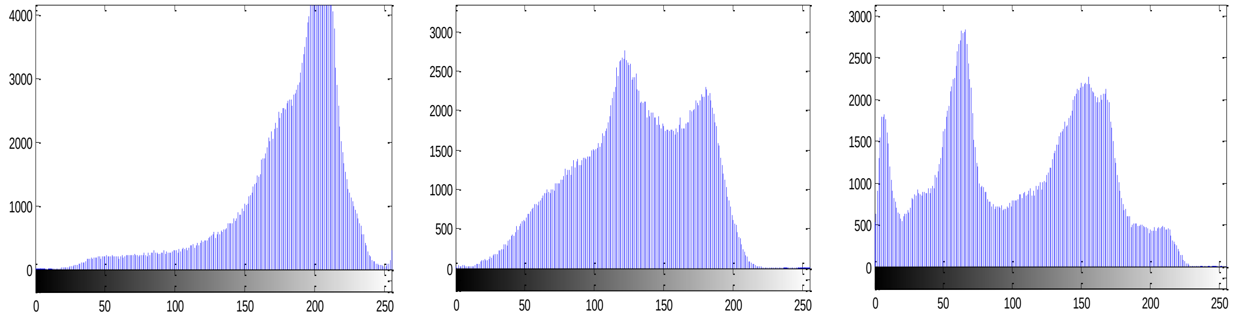
(e) Histogram of MPD stego images



(f) Histogram of PVD+LSB stego images

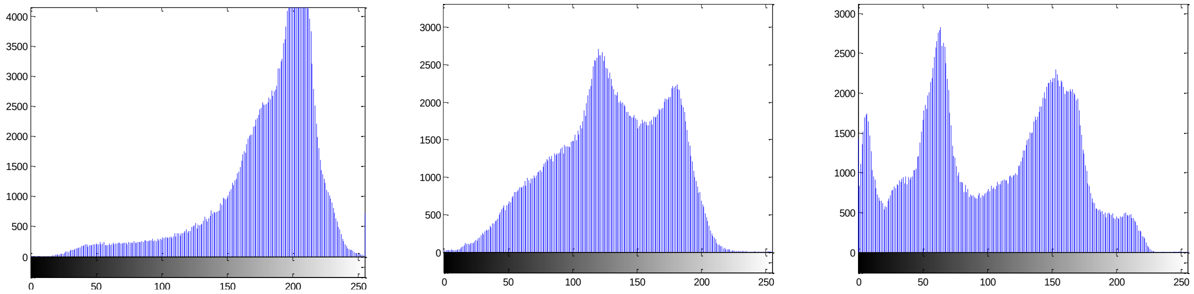


(g) Histogram of Side-Match stego images

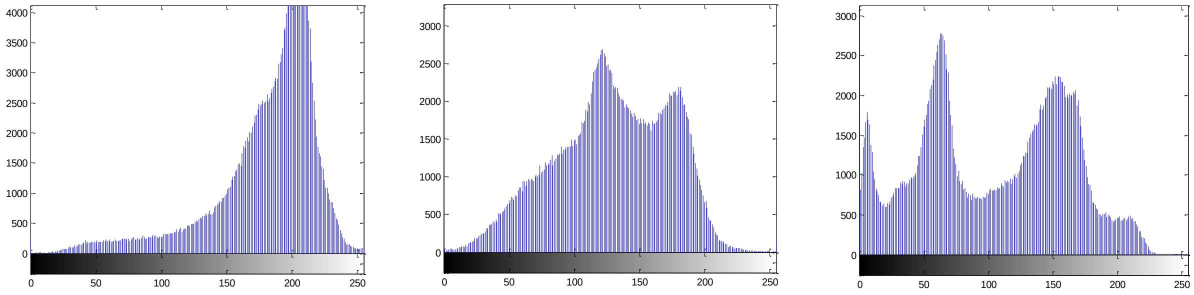


(h) Histogram of Tri-way PVD stego images

Figure 22 (Cont.). Image histogram test (aerial, baboon and peppers images from left to right).



(i) Histogram of MOPVD stego images

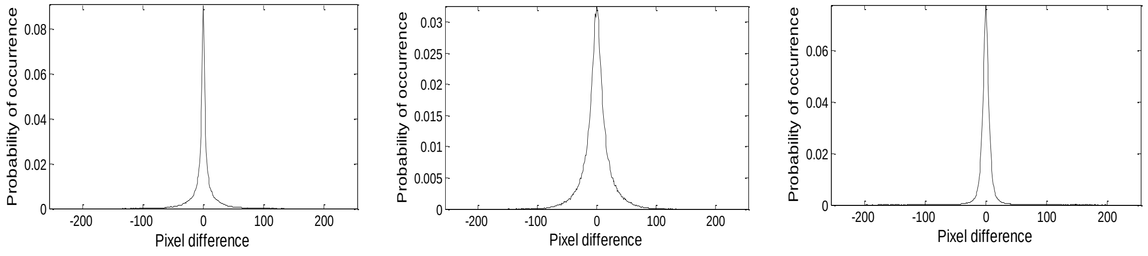


(j) Histogram of CMOPVD stego images

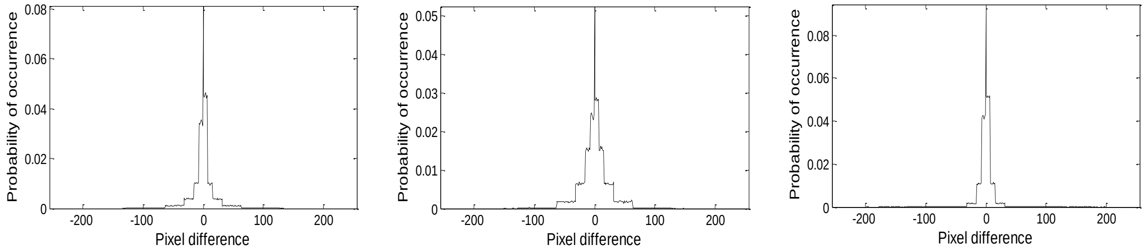
Figure 22 (Cont.). Image histogram test (aerial, baboon and peppers images from left to right).

Since PVD+LSB and the OPVD methods mainly embed the secret data using simple LSB, we can notice the clear impact of the LSB method in the image histogram, see Figure 22 (f). We can notice that the proposed MOPVD with or without rotation does not have remarkable distortion on the stego-image histogram as illustrated in Figure 22 (i and j). We also applied the pixel-pair difference histogram test on the stego-images. In this test, only the pixel-pair difference are calculated and drawn. Figure 23 shows the results for the different methods. From Figure 23 (j), we can observe that block rotation with MOPVD successfully removes the unusual steps from the pixel-pair difference histogram (that may occur when using MOPVD alone; see Figure 23 (i)). However, using

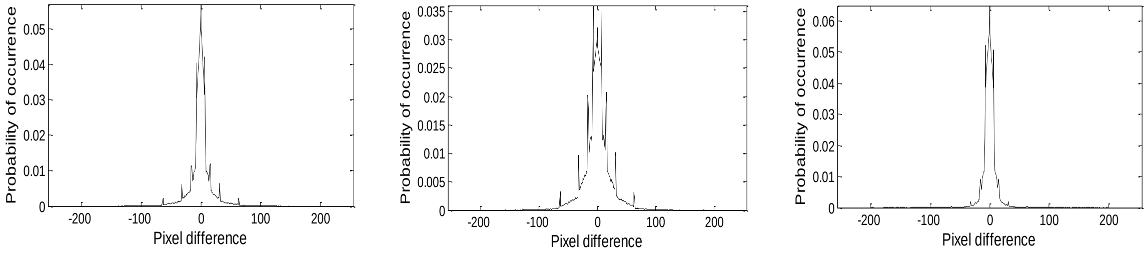
E-MOPVD method, even without rotation, successfully passes the pixel-pair histogram attack as shown in Figure 23 (k), and (l).



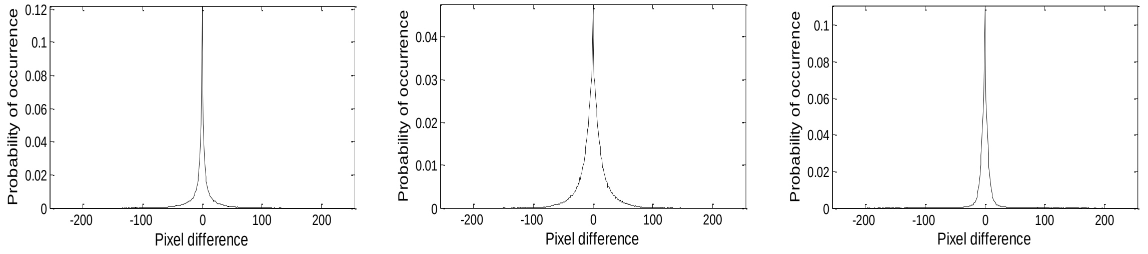
(a) original images



(b) PVD

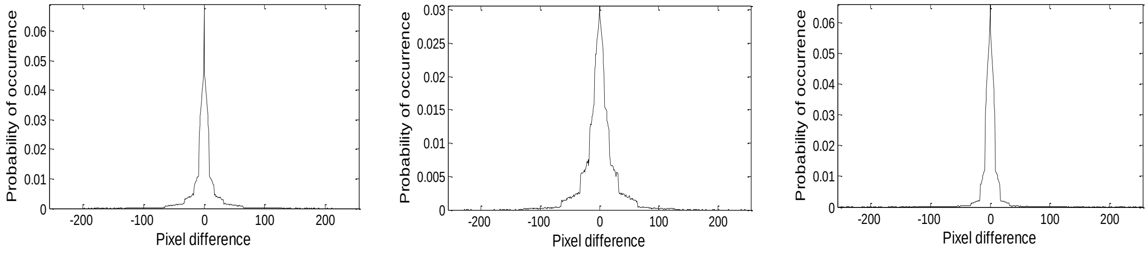


(c) OPVD

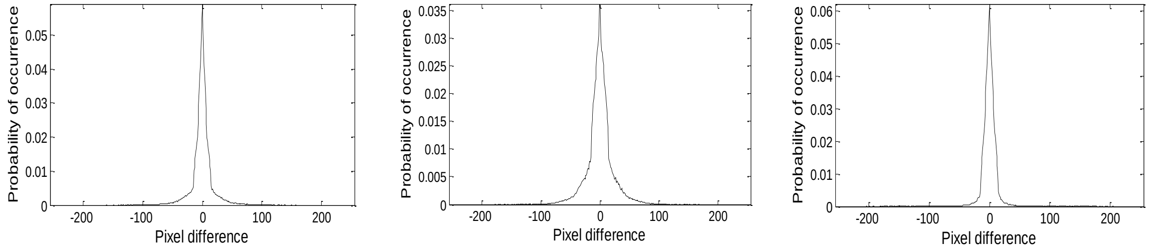


(d) Modulus PVD

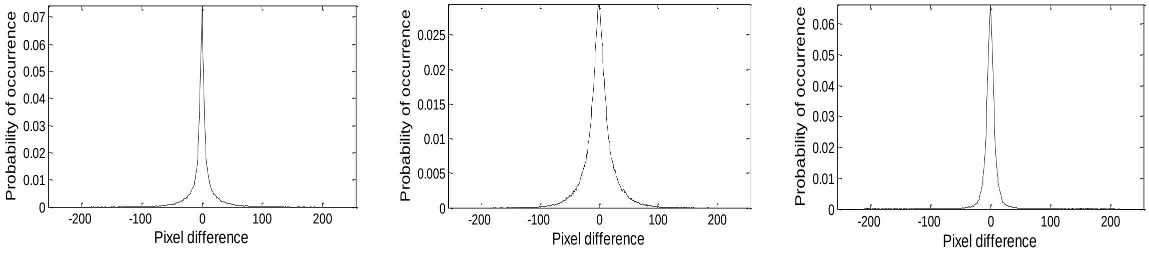
Figure 23. Pixel-pair difference histogram (aerial, baboon and peppers images from left to right).



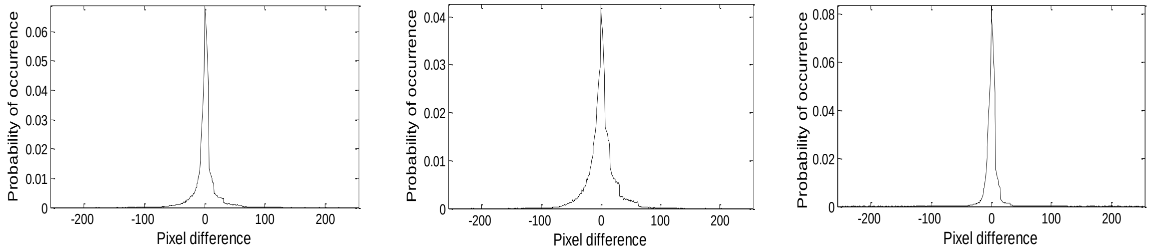
(e) MPD



(f) PVD+LSB

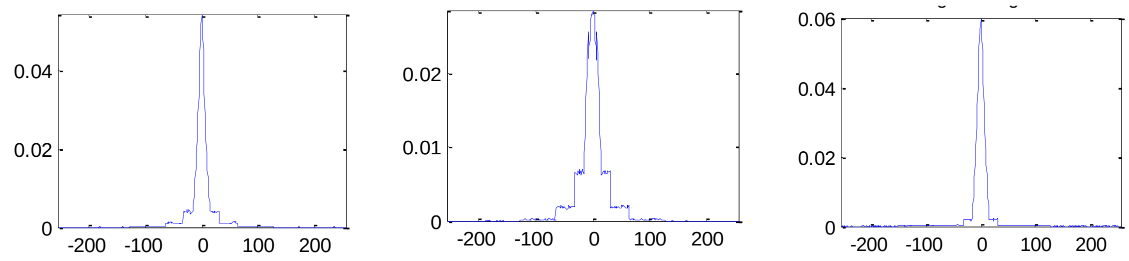


(g) Side-Match

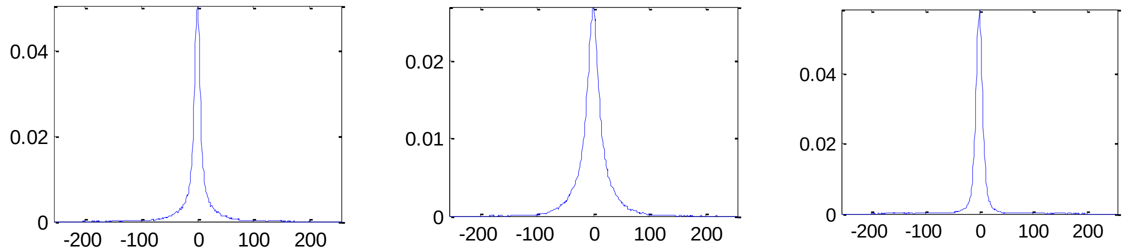


(h) Tri-way PVD

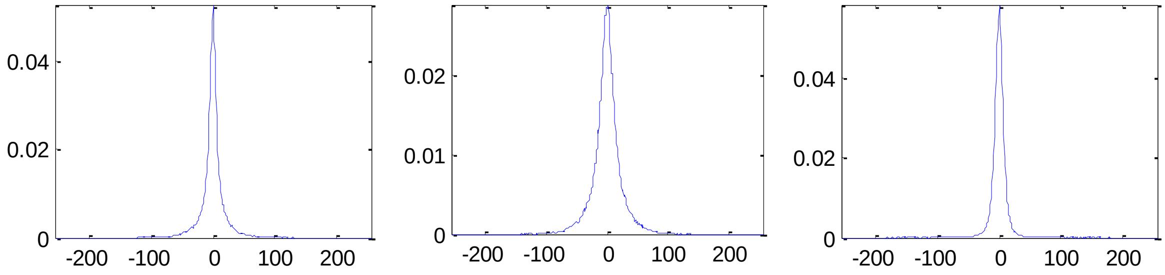
Figure 23 (Cont.). Pixel-pair difference histogram (aerial, baboon and peppers images from left to right).



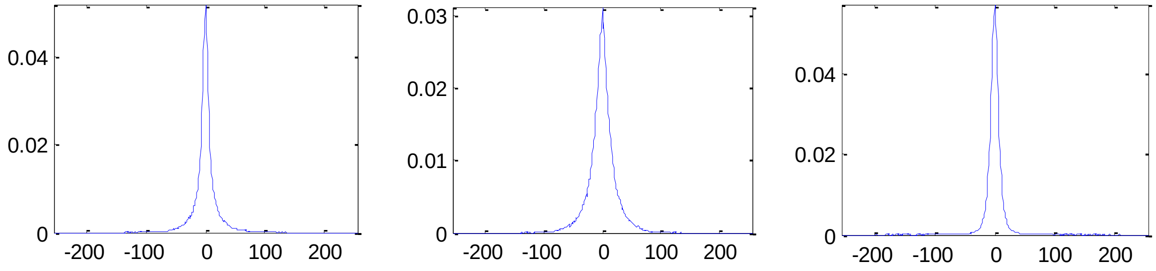
(i) MOPVD



(j) CMOPVD



(k) E-MOPVD



(l) E-CMOPVD

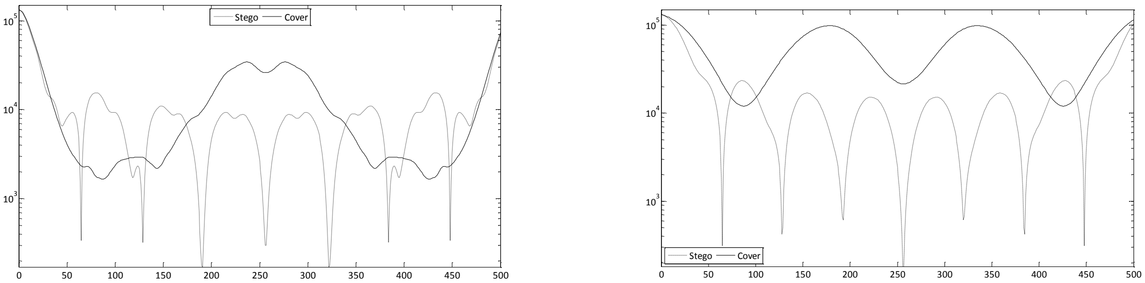
Figure 23 (Cont.). Pixel-pair difference histogram (aerial, baboon and peppers images from left to right).

Finally, we applied the fast Fourier transform steganalysis [5] on different methods by drawing the logarithmic value of the absolute discrete Fourier transform coefficients of the pixel-pair difference histograms before and after embedding. To calculate this, we used the following equation.

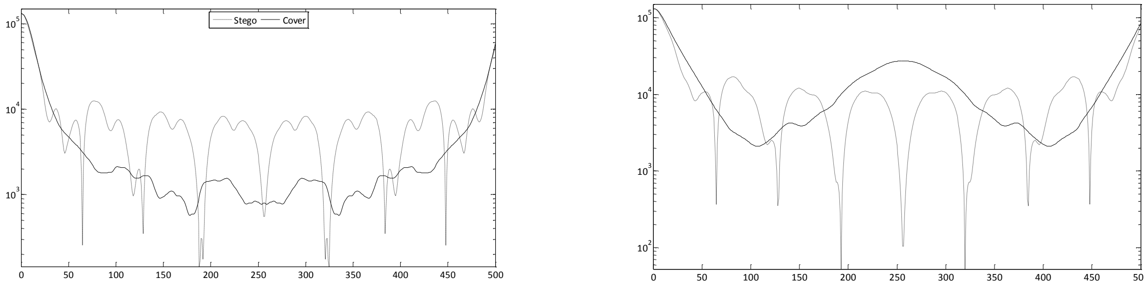
$$F_k = \log\left[abs\left(\sum_{n=0}^{N-1} f_n \times e^{-i2\pi n\frac{k}{N}}\right)\right] \qquad 5.5$$

where N is the number of the image pixel-pairs, f_n is the pixel-pair difference.

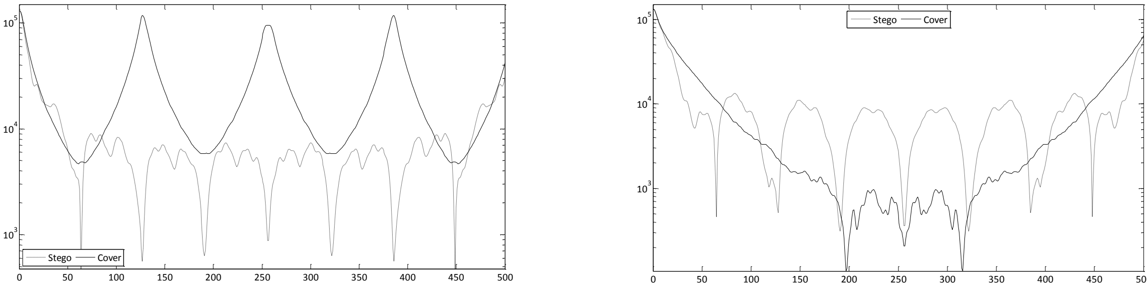
Figure 24 shows the results of this experiment for different methods for the ten test images in the first dataset. From this figure, we can notice a pattern in the form of successive peaks for the stego-images resulting from the PVD, OPVD and Tri-way PVD methods. However, for other methods this pattern doesn't appear clearly for most of the tested images. For instance, it disappears in the baboon stego-image histogram for our proposed CMOPVD method, whereas it appears in the car stego-image histogram for the same method.



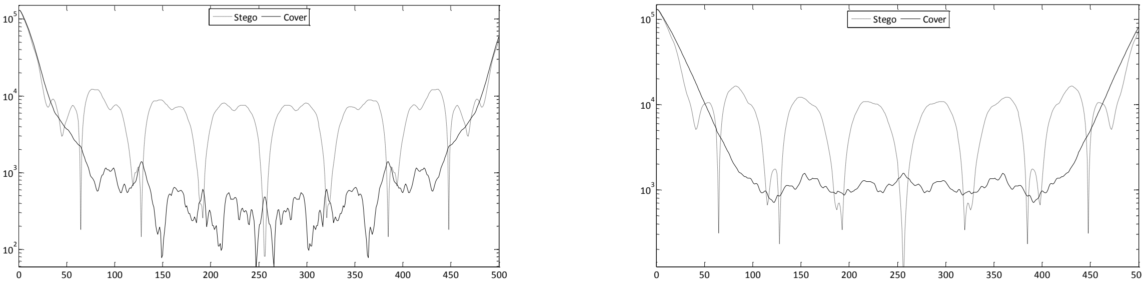
PVD (Tank) PVD (Plane)



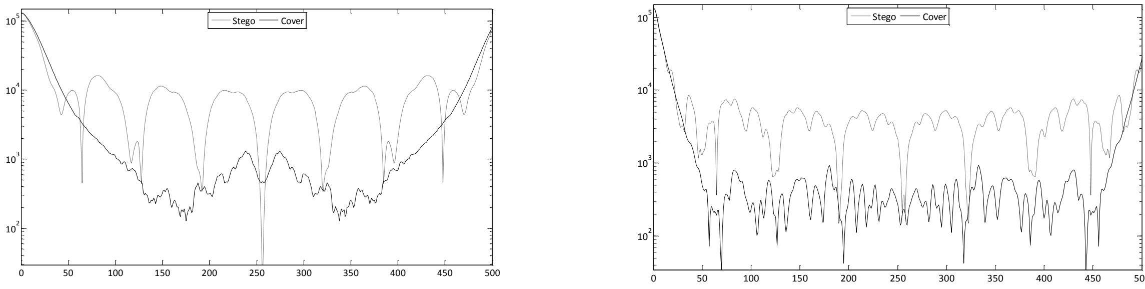
PVD (Elaine) PVD (Car)



PVD (Bridge) PVD (Aerial)

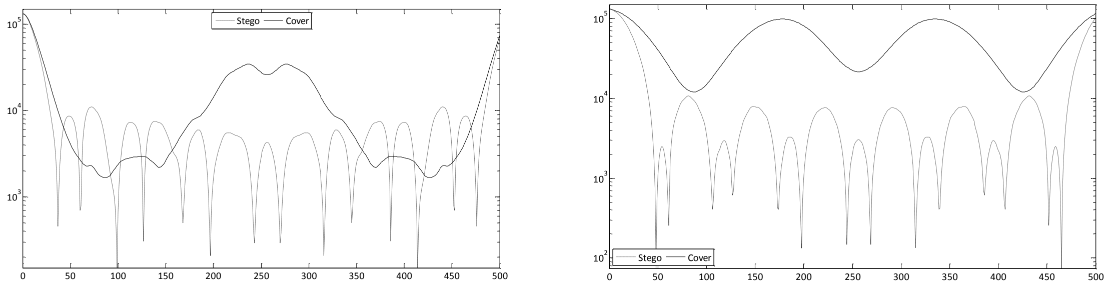


PVD (Boat) PVD (Lena)

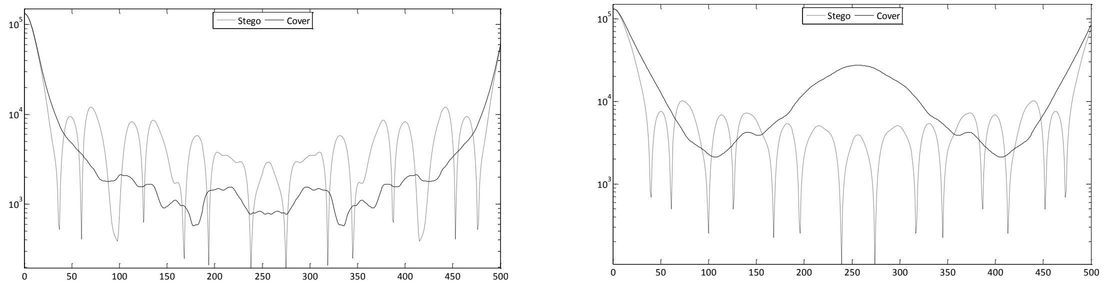


PVD (Peppers) PVD (Baboon)

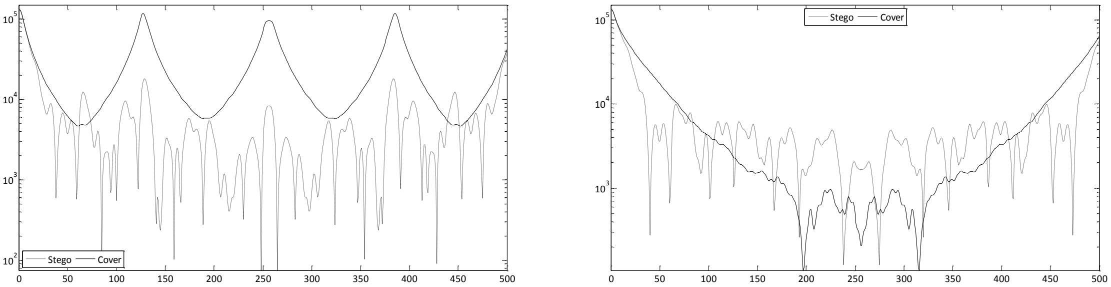
Figure 24. Fourier transform of stego-images for different methods.



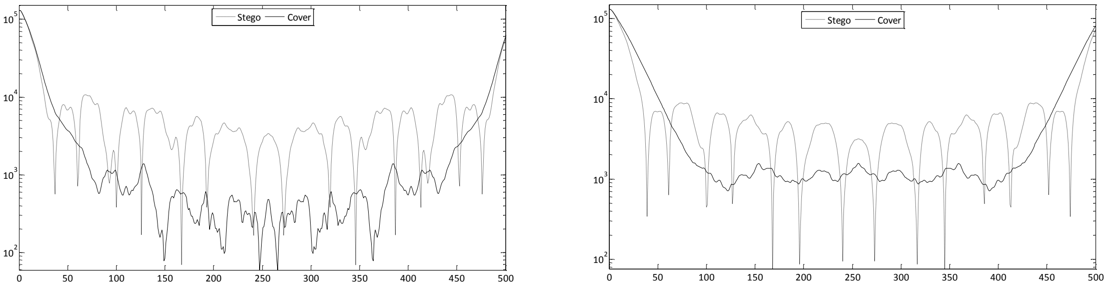
OPVD (Tank) OPVD (Plane)



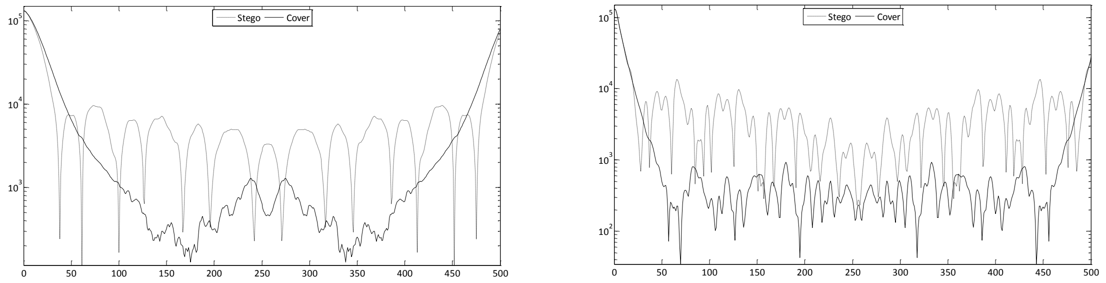
OPVD (Elaine) OPVD (Car)



OPVD (Bridge) OPVD (Aerial)

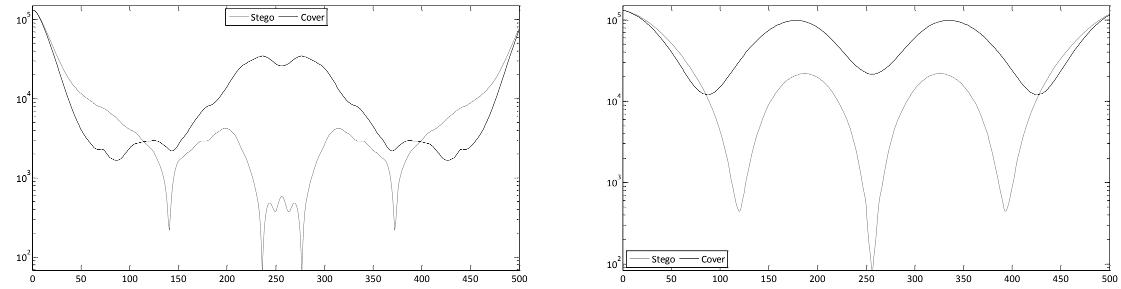


OPVD (Boat) OPVD (Lena)



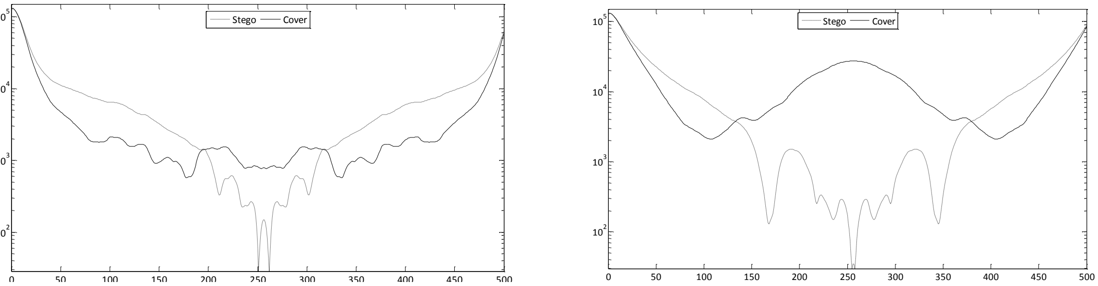
OPVD (Peppers) OPVD (Baboon)

Figure 24 (Cont.). Fourier transform of stego-images for different methods.



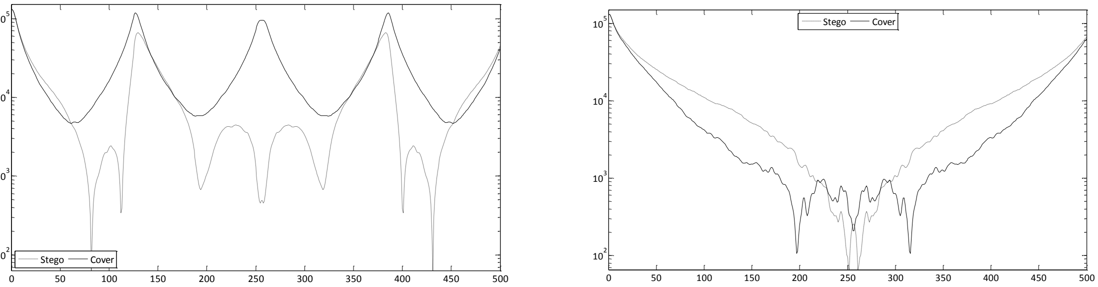
Modulus PVD (Tank)

Modulus PVD (Plane)



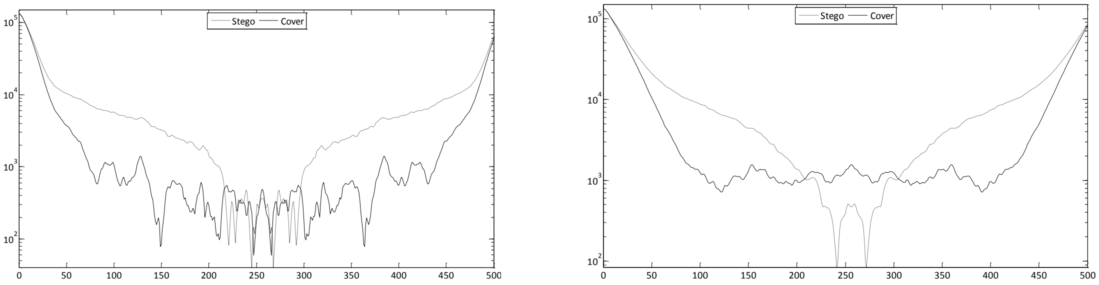
Modulus PVD (Elaine)

Modulus PVD (Car)



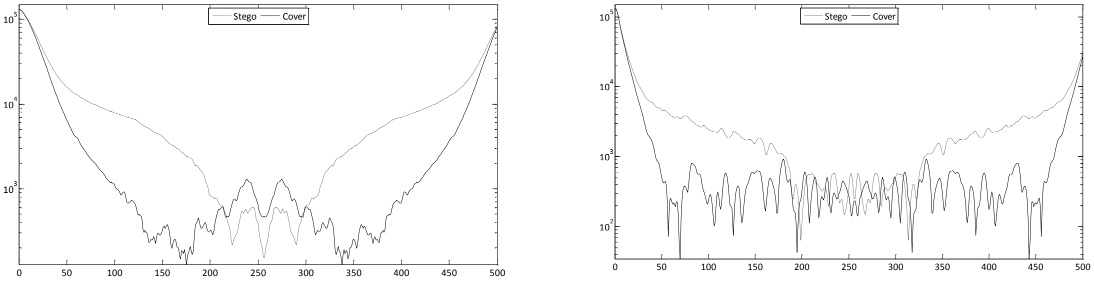
Modulus PVD (Bridge)

Modulus PVD (Aerial)



Modulus PVD (Boat)

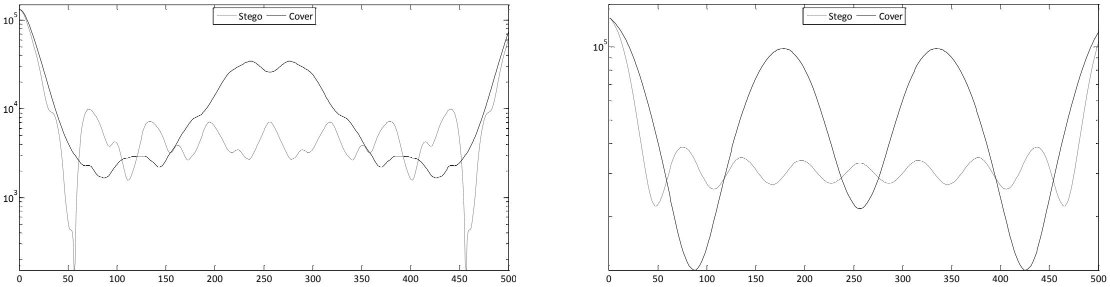
Modulus PVD (Lena)



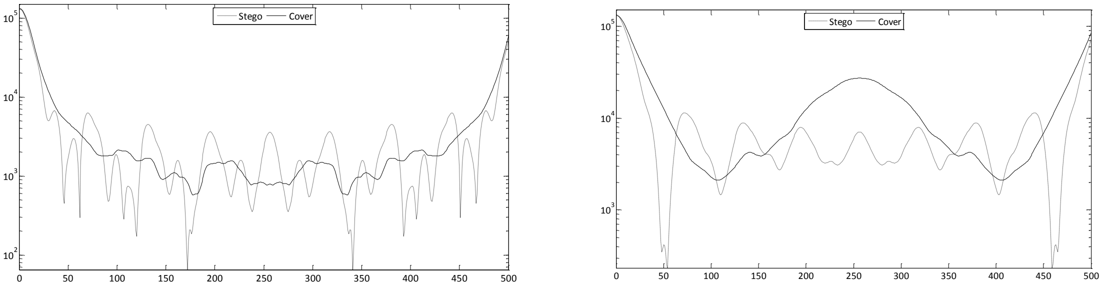
Modulus PVD (Peppers)

Modulus PVD (Baboon)

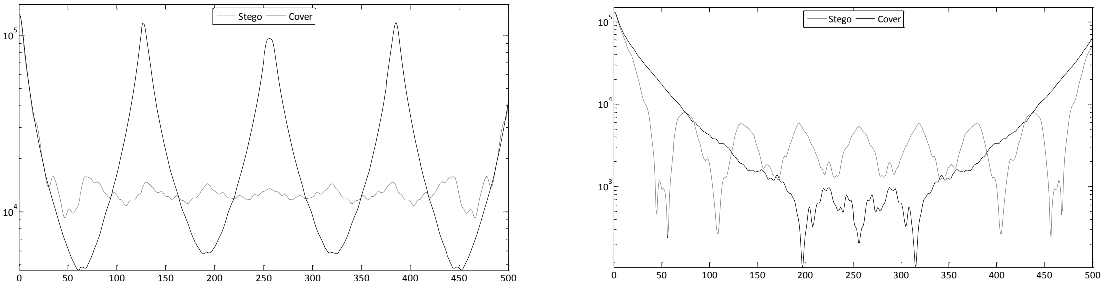
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



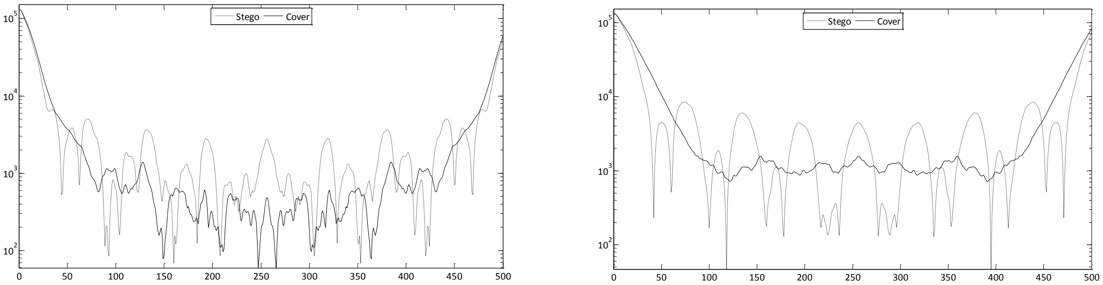
MPD (Tank) MPD (Plane)



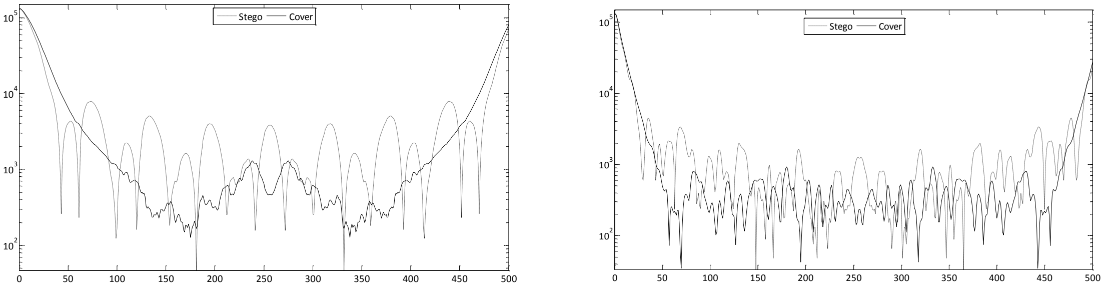
MPD (Elaine) MPD (Car)



MPD (Bridge) MPD (Aerial)

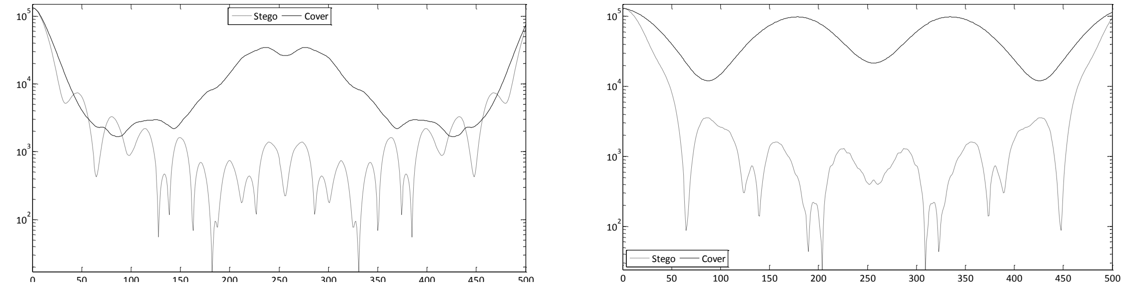


MPD (Boat) MPD (Lena)

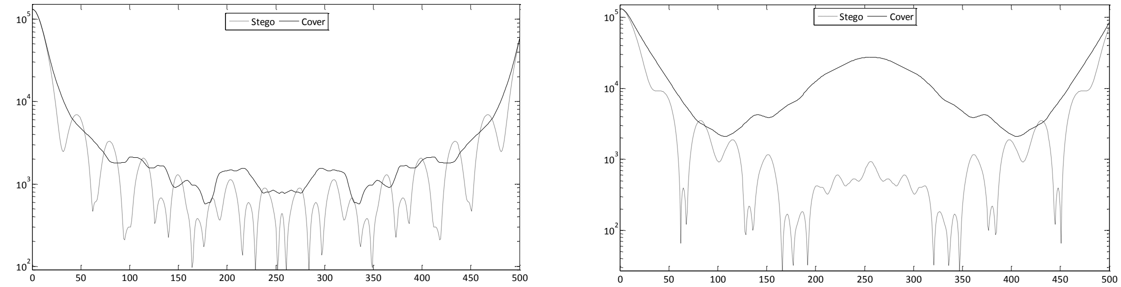


MPD (Peppers) MPD (Baboon)

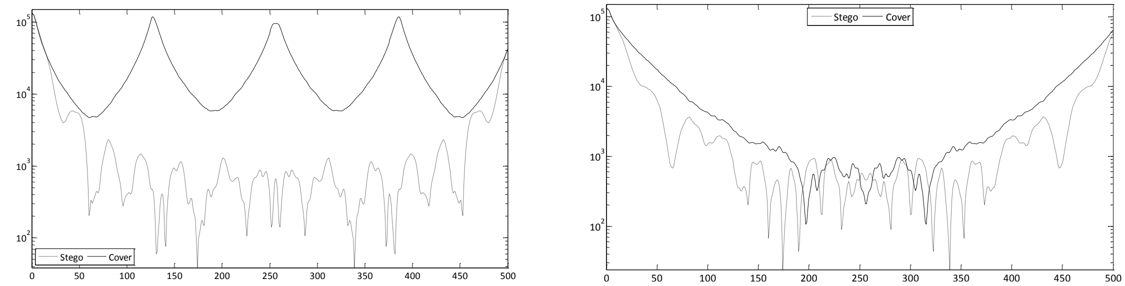
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



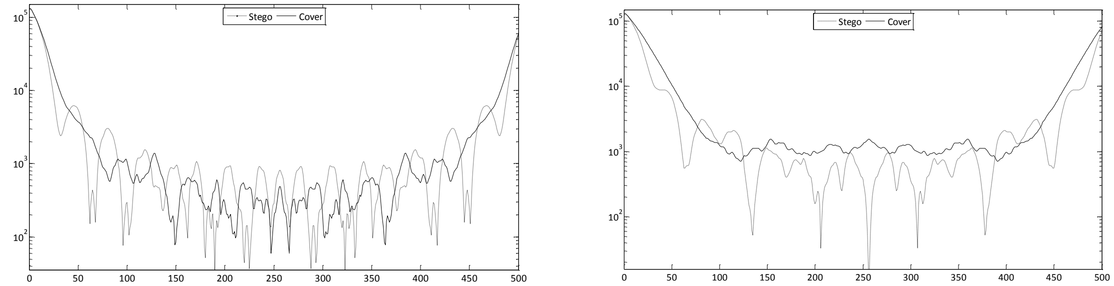
PVD+LSB (Tank) PVD+LSB (Plane)



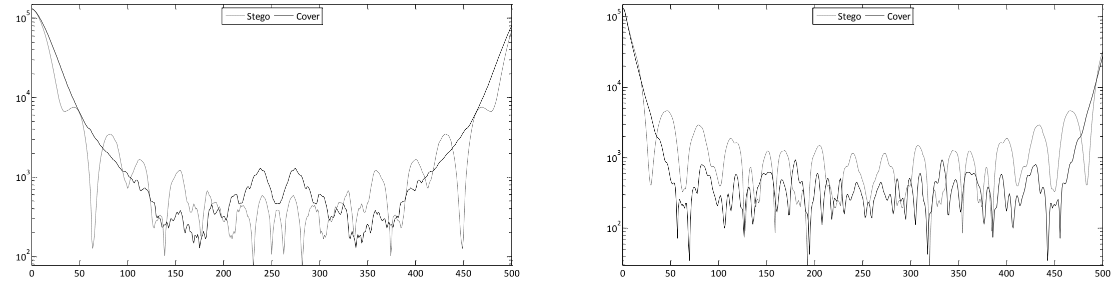
PVD+LSB (Elaine) PVD+LSB (Car)



PVD+LSB (Bridge) PVD+LSB (Aerial)

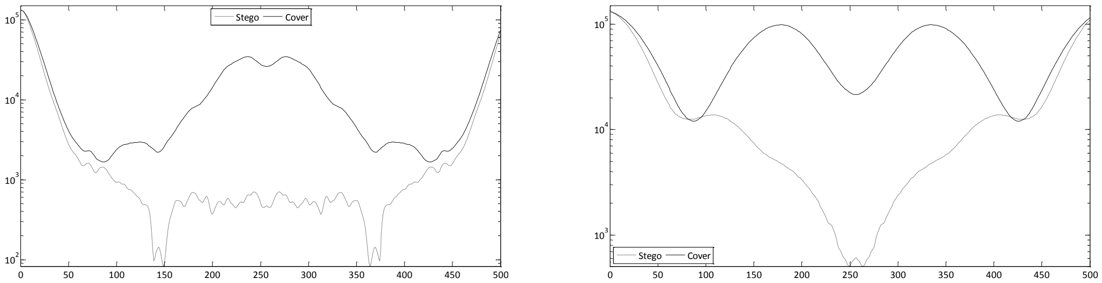


PVD+LSB (Boat) PVD+LSB (Lena)



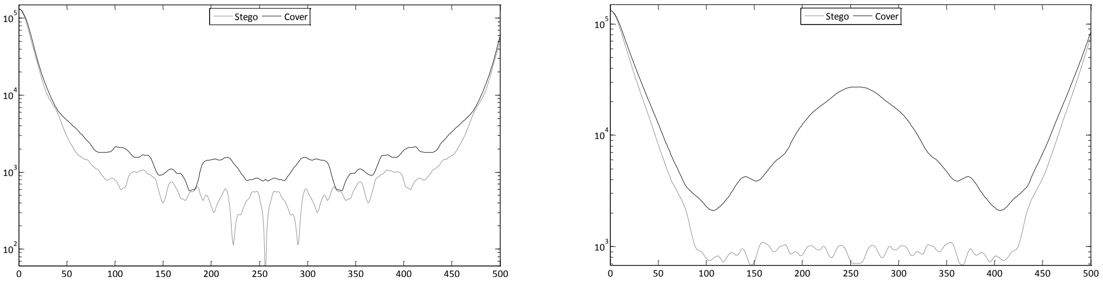
PVD+LSB (Peppers) PVD+LSB (Baboon)

Figure 24 (Cont.). Fourier transform of stego-images for different methods.



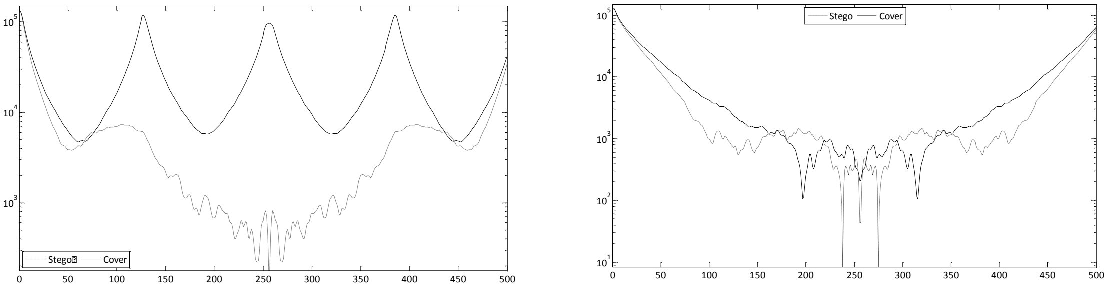
Side-Match (Tank)

Side-Match (Plane)



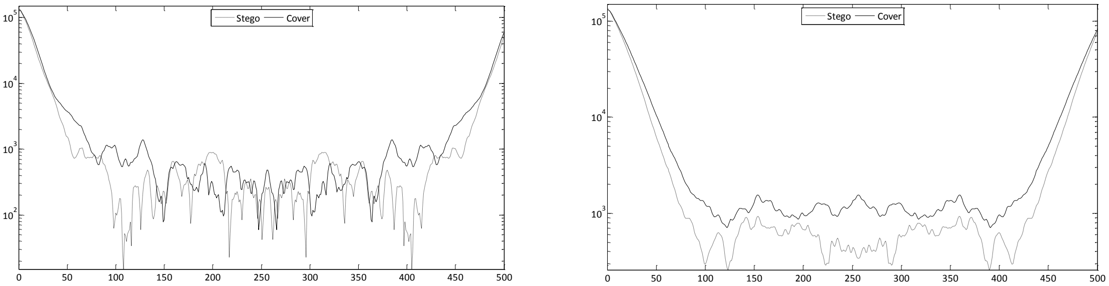
Side-Match (Elaine)

Side-Match (Car)



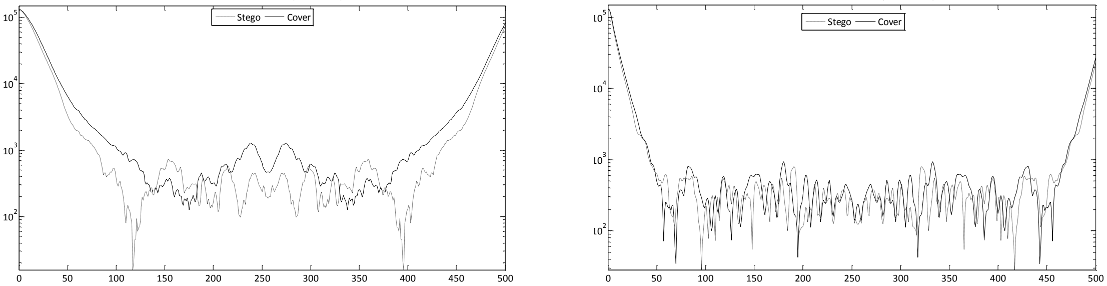
Side-Match (Bridge)

Side-Match (Aerial)



Side-Match (Boat)

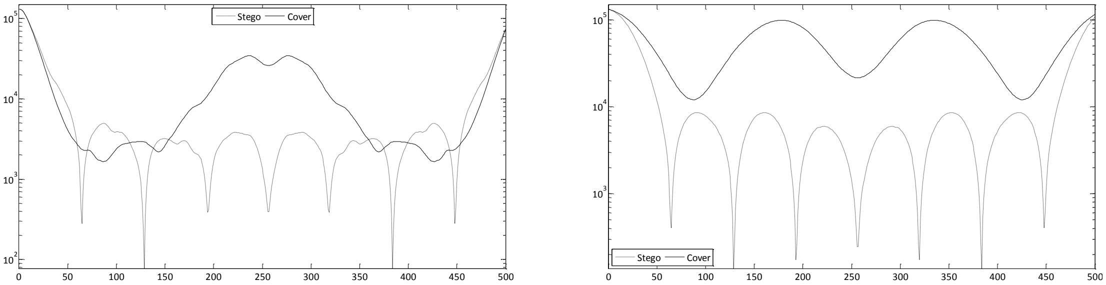
Side-Match (Lena)



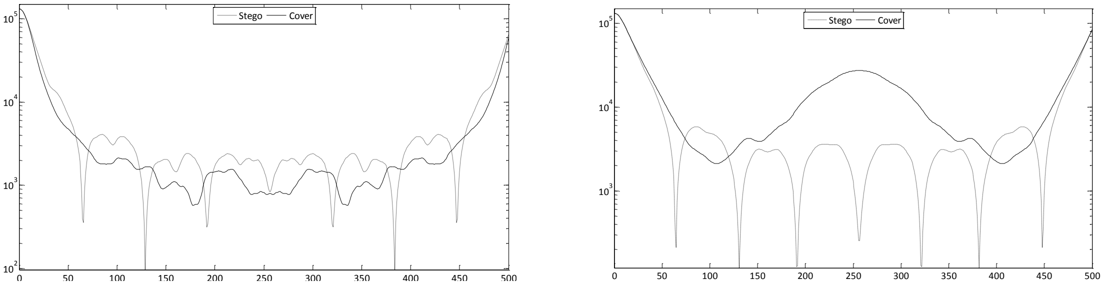
Side-Match (Peppers)

Side-Match (Baboon)

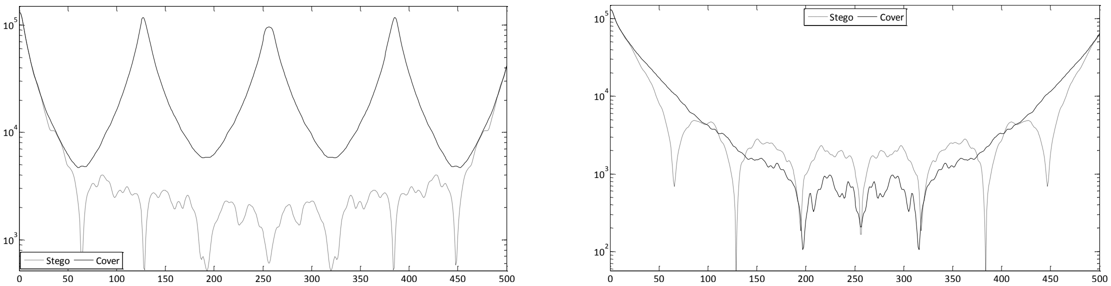
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



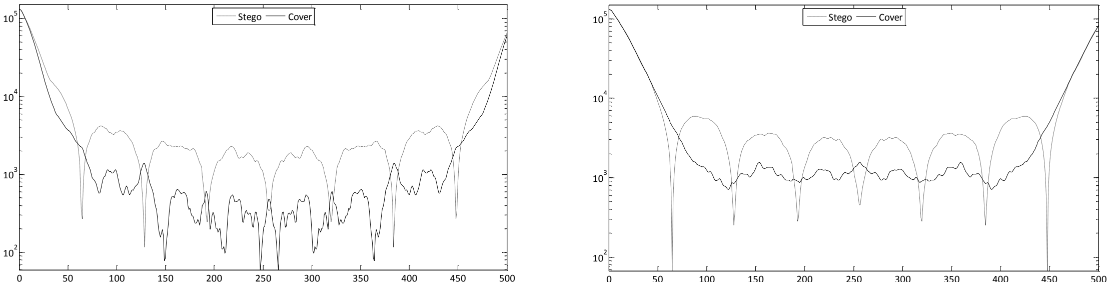
Tri-way (Tank) Tri-way (Plane)



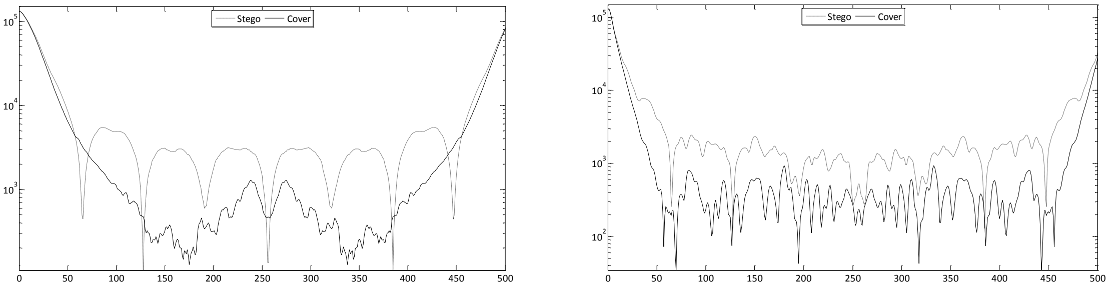
Tri-way (Elaine) Tri-way (Car)



Tri-way (Bridge) Tri-way (Aerial)

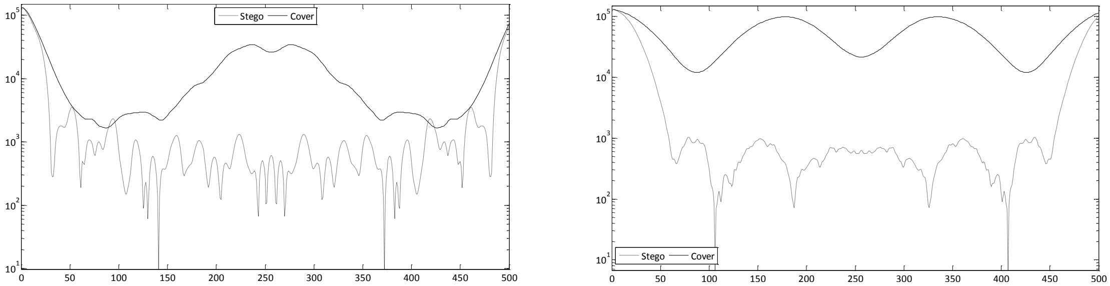


Tri-way (Boat) Tri-way (Lena)

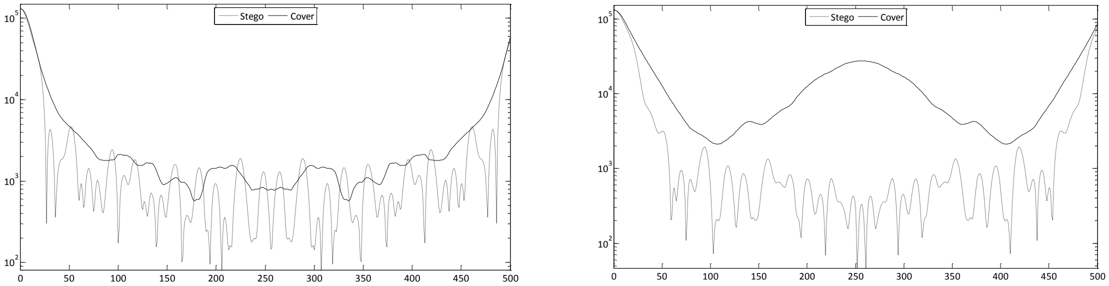


Tri-way (Peppers) Tri-way (Baboon)

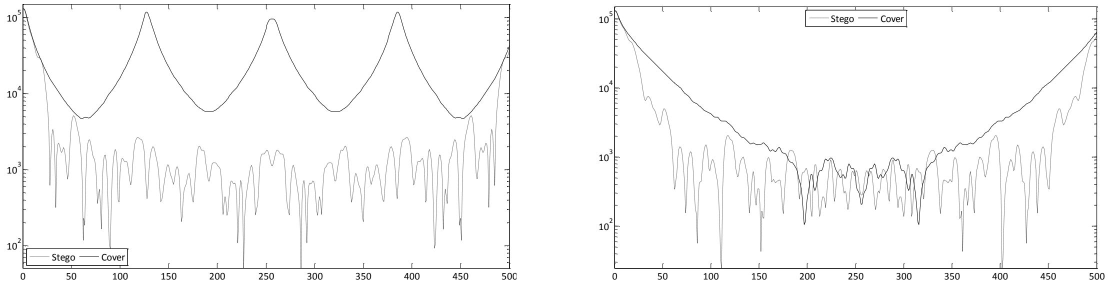
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



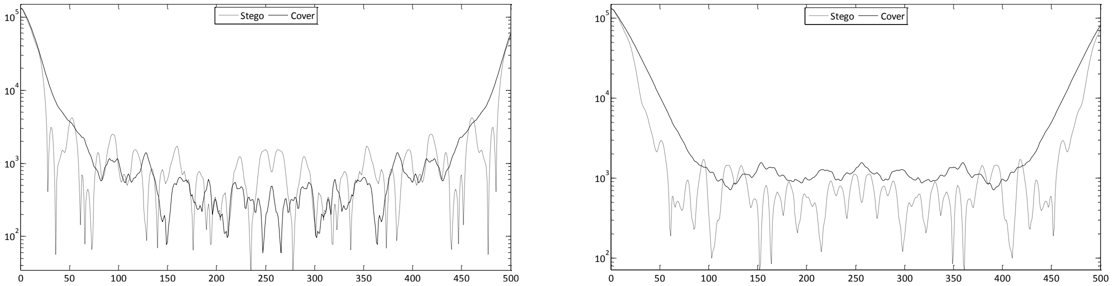
MOPVD (Tank) MOPVD (Plane)



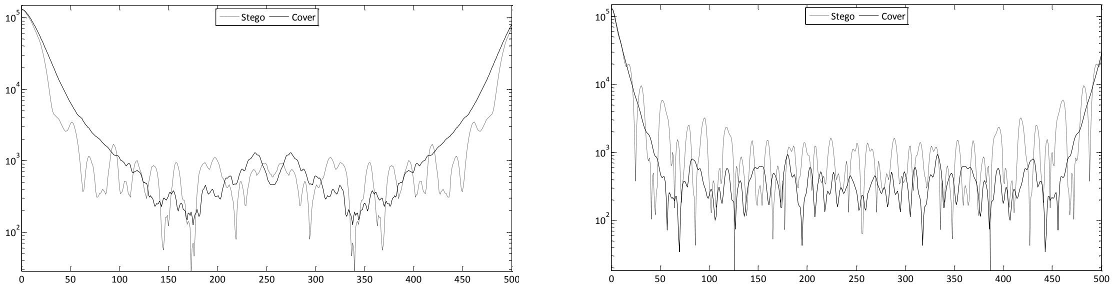
MOPVD (Elaine) MOPVD (Car)



MOPVD (Bridge) MOPVD (Aerial)

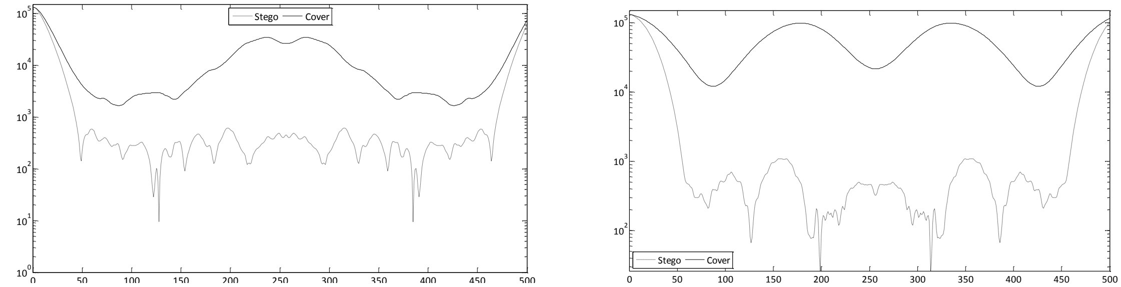


MOPVD (Boat) MOPVD (Lena)

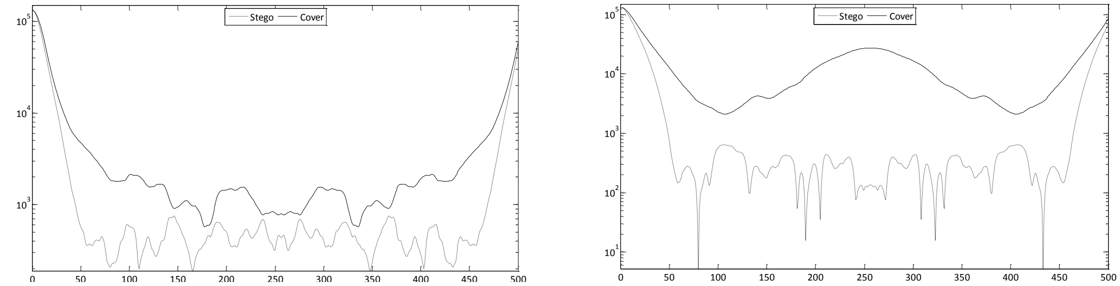


MOPVD (Peppers) MOPVD (Baboon)

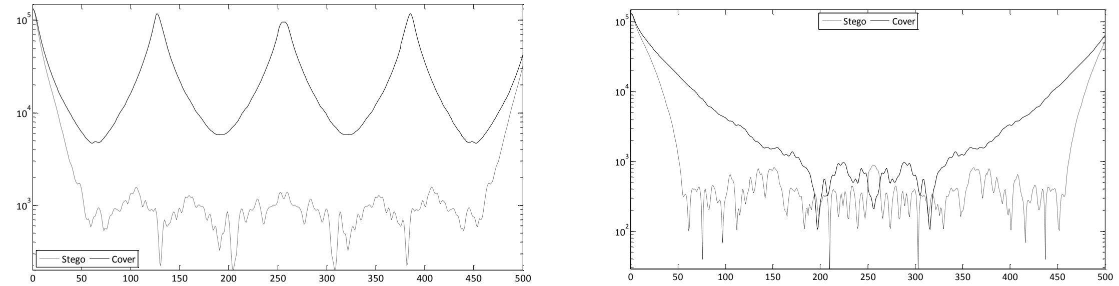
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



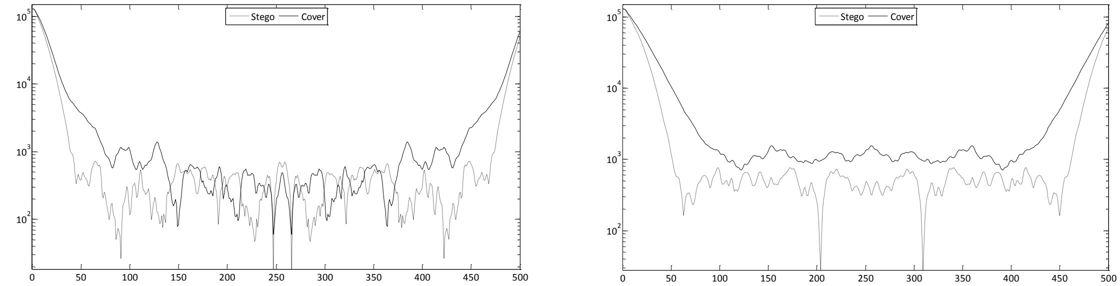
CMOPVD (Tank) CMOPVD (Plane)



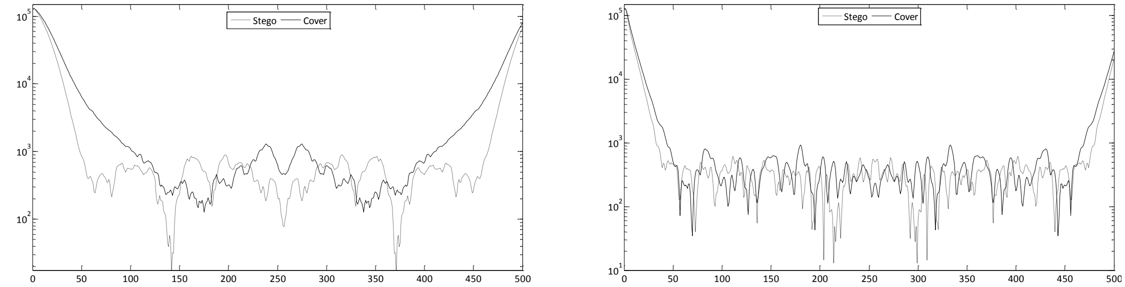
CMOPVD (Elaine) CMOPVD (Car)



CMOPVD (Bridge) CMOPVD (Aerial)

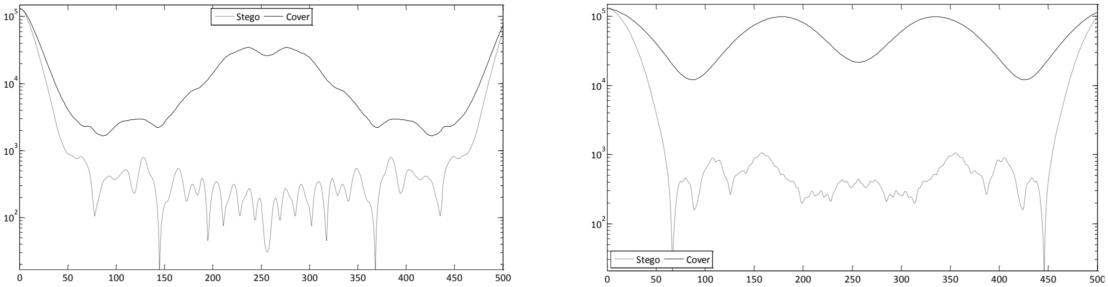


CMOPVD (Boat) CMOPVD (Lena)

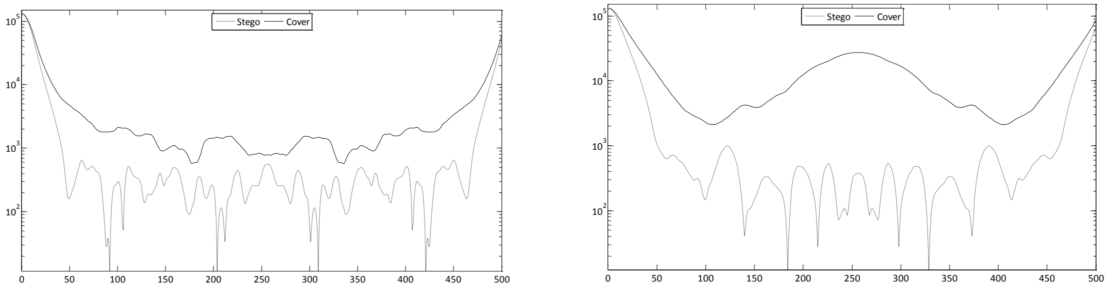


CMOPVD (Peppers) CMOPVD (Baboon)

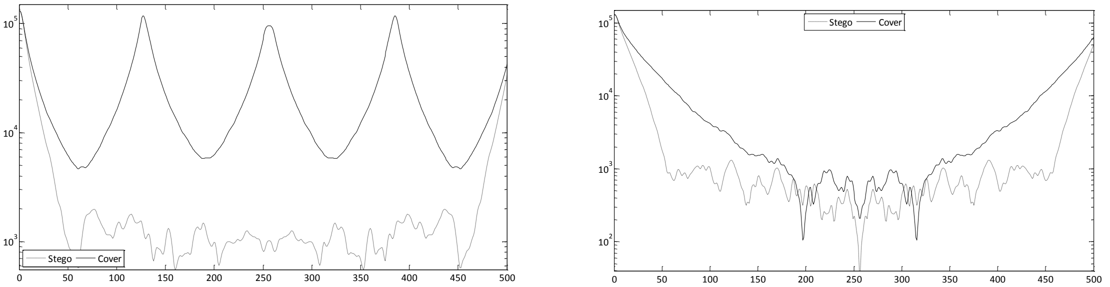
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



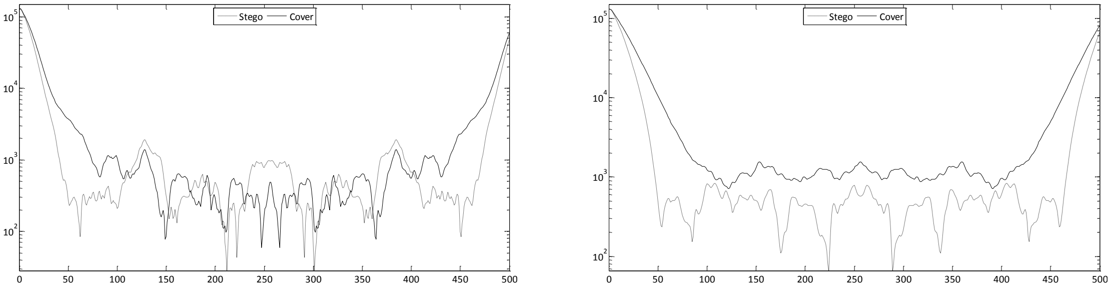
E-MOPVD (Tank) E-MOPVD (Plane)



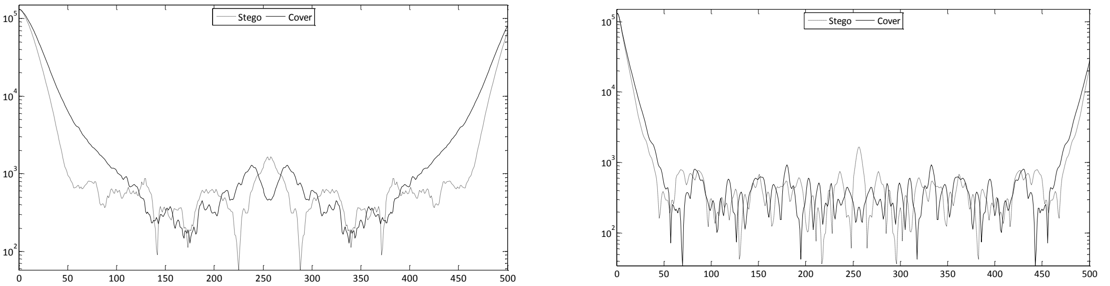
E-MOPVD (Elaine) E-MOPVD (Car)



E-MOPVD (Bridge) E-MOPVD (Aerial)

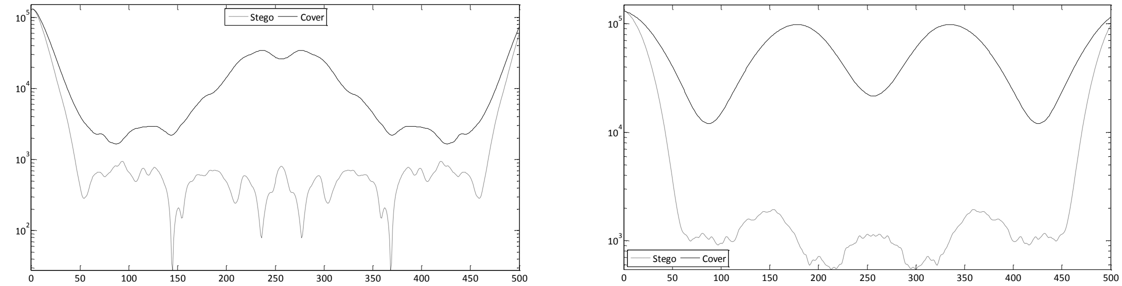


E-MOPVD (Boat) E-MOPVD (Lena)

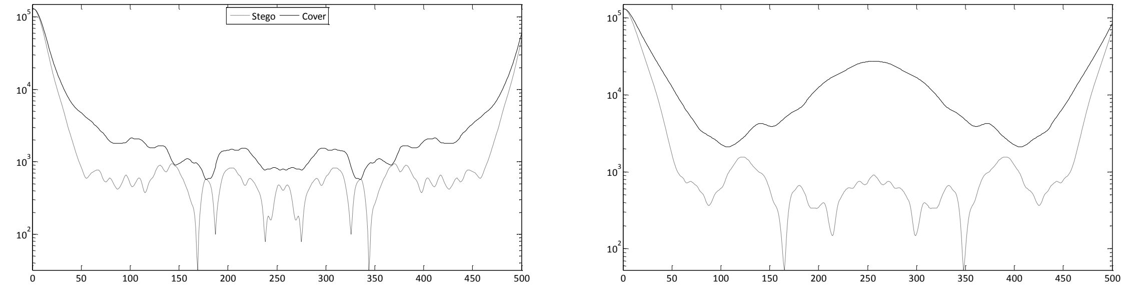


E-MOPVD (Peppers) E-MOPVD (Baboon)

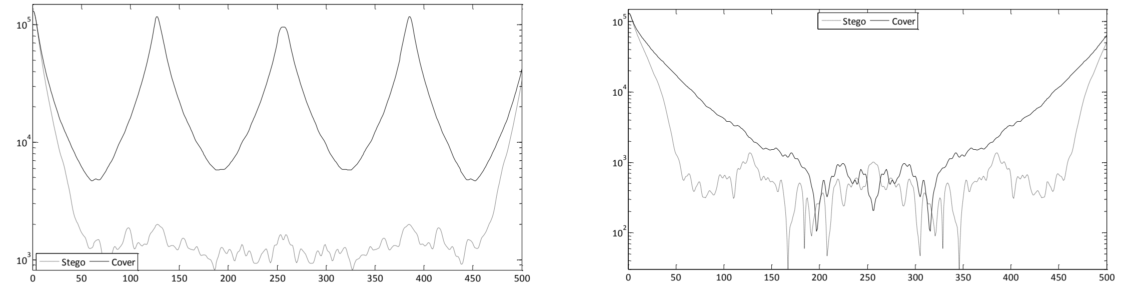
Figure 24 (Cont.). Fourier transform of stego-images for different methods.



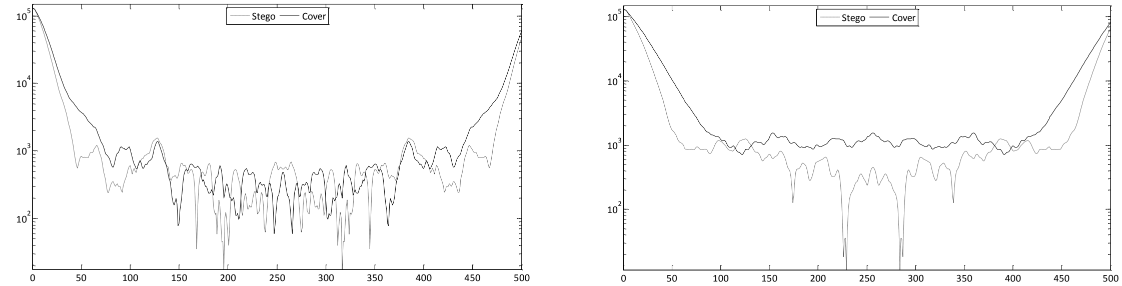
E-CMOPVD (Tank) E-CMOPVD (Plane)



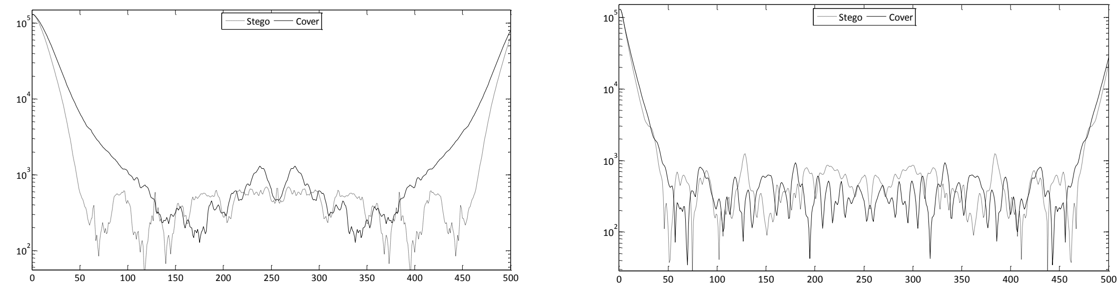
E-CMOPVD (Elaine) E-CMOPVD (Car)



E-CMOPVD (Bridge) E-CMOPVD (Aerial)



E-CMOPVD (Boat) E-CMOPVD (Lena)



E-CMOPVD (Peppers) E-CMOPVD (Baboon)

Figure 24 (Cont.). Fourier transform of stego-images for different methods.

CHAPTER 6

CONCLUSIONS AND FUTURE WORK

Data transmission in open access environments, such the Internet, has raised several security issues. Steganography has been one of the effective methodologies for concealing the existence of secret data by hiding it in a cover medium or carrier. In this work, we reviewed and discussed several existing spatial domain approaches for digital image steganography based on pixel-value differencing including PVD method and six other related methods. We conducted several experiments to evaluate their performance in terms of embedding capacity, quality, and security. We used PSNR, MPSNR, WPSNR and SSIM as performance measures for image quality. For security, we considered their resistance to a number of steganalytic attacks including image histogram, pixel-pair difference histogram and fast Fourier transform spectrum. Based on our analysis, observations and understanding of how pixel-value differencing methods operate, we then proposed a new steganographic system composed mainly of three components that can be used separately or combined together to have more flexibility. These components are rotation based on chaotic maps, modulus overlapping of pixel-value differencing and fuzzy edge detection mechanism. Each component in this system has its own specific strength.

The rotation component increases the security of pixel-value differencing steganography by randomly changing the order of computing the pixel-pair difference. It divides the cover image into 2×2 blocks which are then rotated clockwise or counter-clockwise based on the sequence generated by a chaotic map. Because of the stochastic nature of the sequence generated by the chaotic map that depends only on the initial condition and the control parameter, the prediction of the rotations will be a challenge. This will make the extraction of the embedded message without knowing the chaotic map parameters a difficult chore. Also it improves the histogram of the pixel-pair differences which has been found to be a good steganalytic tool for detecting the existence of data embedded by the PVD method.

The second component, which is the modulus overlapping of pixel-value differencing, increases the embedding capacity. It modifies the remainder of the pixel-pair blocks of the cover image to be equal to the secret data that will be embedded in this pixel pair. Then, this component uses the second pixel of the first block as the first pixel of the second block. This leads to utilizing each pixel individually for embedding the secret data.

The third part of our system depends on the fuzzy edge detection algorithm. Unlike the PVD method which identifies only vertical edges, the fuzzy edge detection algorithm detects edges in various directions and generates an edge image. Since the number of edge pixels is much less than the number of the smooth pixels, the proposed method hides edge information in the stego-image in addition to the secret message. Consequently, the receiver only requires the stego-image to extract the message.

The experimental results show that the proposed system increases the embedding capacity, and the security while preserving a good quality for the stego-image with more than 30dB WPSNR. Some methods such as the proposed MOPVD has increased the average capacity by a factor of 2 more than the original PVD, around 8% over PVD+LSB, more than 47% over OPVD (with a slight degradation in the average PSNR). Moreover, comparing with the HP method, the proposed E-MOPVD has around 23.27% greater average capacity and around 9.21% greater average PSNR than the HP method under the same conditions. We have also found that the security of the proposed system is excellent against the histogram attacks comparing to almost all surveyed methods. For instance, using chaotic rotation in the proposed system significantly reduced the unusual steps in the pixel-pair difference histogram.

Future Work

As future work, we are planning to enhance the proposed steganographic system further. We plan to make the number of embedded bits more adaptive. This can be achieved by calculating the edge strength percentage to identify the number of secret bits which can be embedded in edge pixels. Storing information about edges in the first columns of the image still affects the quality and should be examined further to develop more effective approaches.

APPENDIX 1: FUZZY TEMPLATE BASED EDGE DETECTOR

In this appendix, we explain the details of the Fuzzy Template Based (FTB) edge detector. This method has been proposed by Chaira and Ray [51], [52] [73] to identify edge directions using a set of 16 fuzzy templates; each template is 3×3 matrix representing the edge profile in one direction. Figure 25 shows the adopted 16 fuzzy templates in FTB where the values of the parameters *a* and *b* are arbitrarily chosen to ensure good edge detection; the inventors of the method suggested $a = 0.3$ and $b = 0.8$ [73]. To detect edges, the image is initially normalized, *i.e.* each pixel is divided by the maximum gray level value of the image; thus each pixel value becomes a real value between 0 and 1. Then, each template is located at each pixel position in the image and a fuzzy similarity measure is calculated between the template elements and the image pixels where the template is located. Assume two elements are denoted a_{ij} and b_{ij} (where a_{ij} represents an image pixel and b_{ij} represents the corresponding element in a template *r*), a divergence measure is calculated as follows:

$$Div_r\left(a_{ij},b_{ij}\right) = 2 - \left[1-\mu_I(a_{ij})+\mu_r(b_{ij})\right]e^{\mu_I(a_{ij})-\mu_r(b_{ij})} - \left[1-\mu_r(b_{ij})+\mu_I(a_{ij})\right]e^{\mu_r(b_{ij})-\mu_I(a_{ij})} \tag{A.1}$$

where $\mu_I\left(a_{ij}\right)$ and $\mu_r\left(b_{ij}\right)$ represent the membership values of the $(i,j)^{\text{th}}$ pixel of the normalized image *I* and the corresponding element in the fuzzy template *r*. Using the following max-min measure, a fuzzy value is calculated for the position (i,j) in the image :

$$Div\left(i,j\right) = \max_{r\in\{m_1,\dots,m_{16}\}}\left[\min_{i,j}\left(Div_r(a_{ij},b_{ij})\right)\right] \tag{A.2}$$

If *Div* (*i*,*j*) is greater than a certain threshold, this pixel (*i*,*j*) is an edge pixel and is assigned 1 in the edge image otherwise it becomes 0. At the end the morphological thinning approach implemented in Matlab image processing toolbox is applied to the binary edge image [72]. The flowchart of the FTB procedure for edge detection is shown in Figure 26.

$$\underset{m_1}{\begin{bmatrix} a & a & a \\ 0 & 0 & 0 \\ b & b & b \end{bmatrix}} \quad \underset{m_2}{\begin{bmatrix} a & a & b \\ a & b & 0 \\ b & 0 & 0 \end{bmatrix}} \quad \underset{m_3}{\begin{bmatrix} b & b & b \\ 0 & 0 & 0 \\ a & a & a \end{bmatrix}} \quad \underset{m_4}{\begin{bmatrix} b & a & a \\ 0 & b & a \\ 0 & 0 & b \end{bmatrix}} \quad \underset{m_5}{\begin{bmatrix} b & a & 0 \\ b & a & 0 \\ b & a & 0 \end{bmatrix}} \quad \underset{m_6}{\begin{bmatrix} a & 0 & b \\ a & 0 & b \\ a & 0 & b \end{bmatrix}} \quad \underset{m_7}{\begin{bmatrix} 0 & 0 & 0 \\ b & b & b \\ a & a & a \end{bmatrix}} \quad \underset{m_8}{\begin{bmatrix} 0 & b & a \\ 0 & b & a \\ 0 & b & a \end{bmatrix}}$$

$$\underset{m_9}{\begin{bmatrix} a & a & a \\ b & b & b \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{m_{10}}{\begin{bmatrix} a & b & 0 \\ a & b & 0 \\ a & b & 0 \end{bmatrix}} \quad \underset{m_{11}}{\begin{bmatrix} 0 & 0 & 0 \\ a & a & a \\ b & b & b \end{bmatrix}} \quad \underset{m_{12}}{\begin{bmatrix} 0 & a & b \\ 0 & a & b \\ 0 & a & b \end{bmatrix}} \quad \underset{m_{13}}{\begin{bmatrix} b & b & b \\ a & a & a \\ 0 & 0 & 0 \end{bmatrix}} \quad \underset{m_{14}}{\begin{bmatrix} b & 0 & a \\ b & 0 & a \\ b & 0 & a \end{bmatrix}} \quad \underset{m_{15}}{\begin{bmatrix} b & 0 & 0 \\ a & b & 0 \\ a & a & b \end{bmatrix}} \quad \underset{m_{16}}{\begin{bmatrix} 0 & 0 & b \\ 0 & b & a \\ b & a & a \end{bmatrix}}$$

Figure 25. The FTB sixteen fuzzy templates of size 3×3.

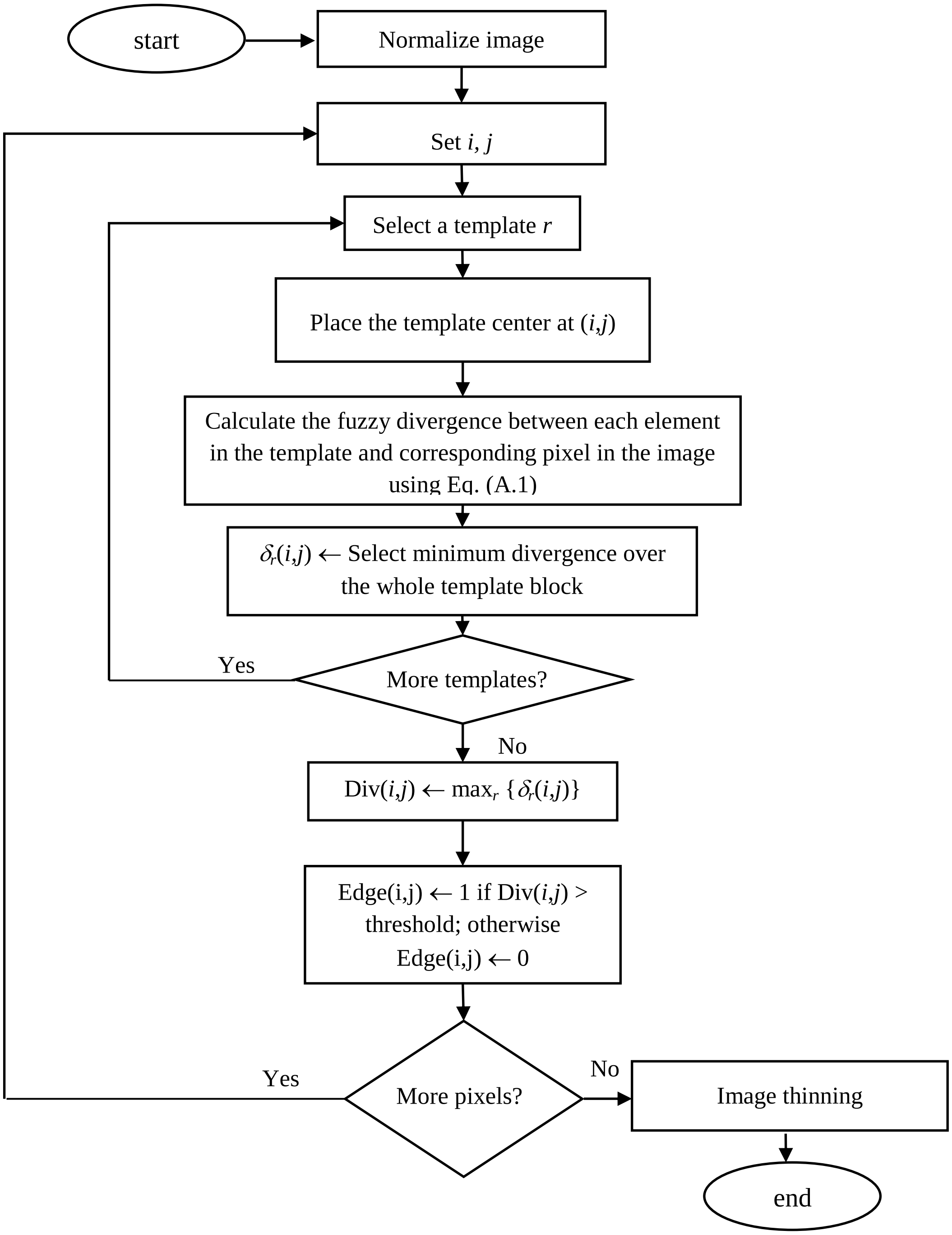

start
Normalize image
Set i, j
Select a template r
Place the template center at (i,j)
Calculate the fuzzy divergence between each element in the template and corresponding pixel in the image using Eq. (A.1)
$\delta_r(i,j) \leftarrow$ Select minimum divergence over the whole template block
More templates?
Yes
No
$\text{Div}(i,j) \leftarrow \max_r \{\delta_r(i,j)\}$
Edge(i,j) ← 1 if $\text{Div}(i,j) >$ threshold; otherwise Edge(i,j) ← 0
More pixels?
Yes
No
Image thinning
end

Figure 26. Fuzzy Template Based (FTB) edge detector flowchart.

APPENDIX 2: PUBLISHED PAPERS

1. El-Alfy, E.-S. M**.** and **Al-Sadi, A**. “A More Effective Steganographic Approach for Color Images by Combining Simple Methods,” The 7th International Computing Conference in Arabic, ICCA 2011, Riyadh, Saudi Arabia, May 2011. (in Arabic)
2. El-Alfy, E.-S. M. and **Al-Sadi, A**. “A Comparative Study of PVD-Based Schemes for Data Hiding in Digital Images,” The 9th ACS/IEEE International Conference on Computer Systems and Applications, (AICCSA 2011), Sharm El-Sheikh, Egypt, June 2011.
3. **Al-Sadi, A. A.** and El-Alfy, E.-S. M**.**, “An Adaptive Steganographic Method for Color Images Based on LSB Substitution and Pixel Value Differencing,” The International Conference on Advances in Computing and Communications (AC 2011), Kochi Kerala, India, July 2011.
4. El-Alfy, E.-S. M. and **Al-Sadi, A**. “Pixel-Value Differencing Steganography: Attacks and Improvements,” in Proceedings of the First Taibah University International Conference on Computing and Information Technology, (ICCIT2012), Al-Madinah Al-Munawwarah, Saudi Arabia, March 2012.
5. El-Alfy, E.-S. M. and **Al-Sadi, A**. “Pixel Improved Pixel Value Differencing Steganography Using Logistic Chaotic Maps,” in Proceedings of the 8th International Conference on Innovations in Information Technology, (IIT2012), Al Ain, UAE, March 2012.
6. El-Alfy, E.-S. M. and **Al-Sadi, A**. “High-Capacity Image Steganography Based on Overlapped Pixel Differences and Modulus Function,” in Proceedings of the Fourth International Conference on Networked Digital Technologies, (NDT2012), Dubai, UAE, April 2012.
7. **Al-Sadi A**. and El-Alfy, E.-S. M. “Security Improvement of PVD Steganographic Method against Histogram Attack,” Third Scientific Conference for Graduate and Undergraduate Students, Khobar, April/May 2012.
8. **Al-Sadi A**. and El-Alfy, E.-S. M. “High-Capacity Steganographic Method Based on Overlapped PVD,” Third Scientific Conference for Graduate and Undergraduate Students, Khobar, April/May 2012.

APPENDIX 3: LIST OF ABBREVIATIONS

CMOPVD	Chaotic MOPVD
DCT	Discrete Cosine Transform
E-CMOPVD	E-MOPVD with chaotic block rotation
E-MOPVD	A combination of FTB and modified MOPVD functions
FTB	Fuzzy Template Based (edge detector)
HP	High Payload Method
HVS	Human Visual System
LSB	Least Significant Bit
Modulus-PVD	PVD with Modulus Function
MOPVD	Modulus Overlapping Pixel-Value Differencing
MPD	Multi-Pixel Differencing
MPSNR	Maximum Peak Signal-to-Noise Ratio
OPVD	Overlapping Pixel-Value Differencing
PSNR	Peak Signal-to-Noise Ratio
PVD	Pixel-Value Differencing
PVD+LSB	A combination of the PVD and LSB replacement methods
SMVQ	Two-sided-match vector quantization
SSIM	Structural Similarity Index
TPVD	Tri-way PVD Method
WPSNR	Weighted Peak Signal-to-Noise Ratio

References

[1] D. C. Wu and W. H. Tsai, “A steganographic method for images by pixel-value differencing,” *Pattern Recognition Letters*, vol. 24, no. 9–10, pp. 1613–1626, 2003.

[2] A. A. Alshennawy and A. A. Aly, “Edge detection in digital images using fuzzy logic technique,” in *Proc. World Academy of Science, Engineering and Technology*, vol. 51, pp. 178–186, 2009.

[3] C. H. Yang and C. Y. Weng, “A steganographic method for digital images by multi-pixel differencing,” in *Proc. International Computer Symposium, Taipei*, 2006.

[4] C. H. Yang, S. J. Wang, and C. Y. Weng, “Analyses of pixel-value-differencing schemes with LSB replacement in stegonagraphy,” in *Proc. 3rd International Conference on Intelligent Information Hiding and Multimedia Signal Processing*, 2007.

[5] X. Zhang and S. Wang, “Vulnerability of pixel-value differencing steganography to histogram analysis and modification for enhanced security,” *Pattern Recognition Letters*, vol. 25, no. 3, pp. 331–339, 2004.

[6] F. A. Petitcolas, R. J. Anderson, and M. G. Kuhn, “Information hiding–a survey,” *Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062–1078, 1999.

[7] K. Rabah, “Steganography–the art of hiding data,” *Information Technology Journal*, vol. 3, no. 3, pp. 245–269, 2004.

[8] A. R. Madane and R. Khare, “Time domain steganography,” in *Proc. International Workshop on Machine Intelligence Research*, 2009.

[9] C. H. Huang, S. C. Chuang, and J. L. Wu, “Digital invisible ink and its applications in steganography,” in *Proc. 8th Workshop on Multimedia and Security*, 2006.

[10] F. Y. Shih, *Image Processing and Pattern Recognition: Fundamentals and Techniques*. Wiley-IEEE Press, 2010.

[11] H. Nyeem, W. Boles, and C. Boyd, “Developing a digital image watermarking model,” in *Proc. Digital Image Computing Techniques and Applications*, Queensland, Australia, 2011.

[12] H. Nyeem, W. Boles, and C. Boyd, “On the robustness and security of digital image watermarking,” in *Proc. International Conference on Informatics, Electronics & Vision*, 2012.

[13] J. Nazario, *Defense and Detection Strategies against Internet Worms*. Artech House, 2004.

[14] J. Mielikainen, “LSB matching revisited,” *IEEE Signal Processing Letters*, vol. 13, no. 5, pp. 285–287, 2006.

[15] X. Li, B. Yang, D. Cheng, and T. Zeng, “A generalization of LSB matching,” *IEEE Signal Processing Letters*, vol. 16, no. 2, pp. 69–72, 2009.

[16] R. Z. Wang, C. F. Lin, and J. C. Lin, “Image hiding by optimal LSB substitution and genetic algorithm,” *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, 2001.

[17] C. K. Chan and L. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[18] C. C. Chang, Y. H. Huang, H. Y. Tsai, and C. Qin, “Prediction-based reversible data hiding using the difference of neighboring pixels,” *International Journal of Electronics and Communications*, vol. 66, no. 9, pp. 758–766, Sep. 2012.

[19] J. Mandal, “A frequency domain steganography using Z transform,” in *Proc. International Workshop on Embedded Computing and Communication System*, 2011.

[20] N. Kafri and H. Y. Suleiman, “Bit-4 of frequency domain-DCT steganography technique,” in *Proc. 1st International Conference on Networked Digital Technologies*, 2009.

[21] A. Westfeld and A. Pfitzmann, “High capacity despite better steganalysis (F5–a steganographic algorithm),” in *Proc. 4th International Workshop on Information Hiding*, Pennsylvania, USA, 2001.

[22] N. Provos, “Defending against statistical steganalysis,” in *Proc. 10th USENIX Security Symposium*, 2001.

[23] I. J. Cox, *Digital Watermarking and Steganography*. Morgan Kaufmann, 2008.

[24] Q. Liu and A. H. Sung, “Feature mining and nuero-fuzzy inference system for steganalysis of LSB matching steganography in grayscale images,” in *Proc. 20th International joint Conference on Artificial Intelligence*, 2007.

[25] T. Pevný and J. Fridrich, “Merging Markov and DCT features for multi-class JPEG steganalysis,” in *Proc. Electronic Imaging, Security, Steganography, and Watermarking of Multimedia Contents IX*, San Jose, USA, 2007.

[26] E. Zheng, X. Ping, T. Zhang, and G. Xiong, “Steganalysis of LSB matching based on local variance histogram,” in *Proc. 17th IEEE International Conference on Image Processing*, Hong Kong, 2010.

[27] A. D. Ker, “Steganalysis of LSB matching in grayscale images,” *IEEE Signal Processing Letters*, vol. 12, no. 6, pp. 441– 444, Jun. 2005.

[28] D. Neeta, K. Snehal, and D. Jacobs, “Implementation of LSB steganography and its evaluation for various bits,” in *Proc. 1st International Conference on Digital Information Management*, India, 2006.

[29] A. Cheddad, J. Condell, K. Curran, and P. Mc Kevitt, “Digital image steganography: Survey and analysis of current methods,” *Signal Processing*, vol. 90, no. 3, pp. 727–752, 2010.

[30] V. Sabeti, S. Samavi, M. Mahdavi, and S. Shirani, “Steganalysis of pixel-value differencing steganographic method,” in *Proc. IEEE Pacific Rim Conference on Communications, Computers and Signal Processing*, Canada, 2007.

[31] V. Sabetia, S. Samavia, M. Mahdavia, and S. Shiranib, “Steganalysis and payload estimation of embedding in pixel differences using neural networks,” *Pattern Recognition*, vol. 43, no. 1, pp. 405–415, 2010.

[32] C. M. Wang, N. I. Wu, C. S. Tsai, and M. S. Hwang, “A high quality steganographic method with pixel-value differencing and modulus function,” *Journal of Systems and Software*, vol. 81, no. 1, pp. 150–158, 2008.

[33] J. Fridrich, M. Goljan, and R. Du, “Reliable detection of LSB steganography in color and grayscale images,” in *Proc. ACM Workshop on Multimedia and Security*, 2001.

[34] J. C. Joo, H. Y. Lee, C. Bui, W. Y. Yoo, and H. K. Lee, “Steganalytic measures for the steganography using pixel-value differencing and modulus function,” in *Proc. the 9th Pacific Rim Conference on Multimedia: Advances in Multimedia Information Processing*, 2008.

[35] J. C. Joo, K. S. Kim, H. K. Lee, and H. Y. Lee, “Histogram estimation-scheme-based steganalysis defeating the steganography using pixel-value differencing and modulus function,” *Optical Engineering*, vol. 49, no. 7, 2010.

[36] J. C. Joo, H. Y. Lee, and H. K. Lee, “Improved steganographic method preserving pixel-value differencing histogram with modulus function,” *EURASIP Journal on Advances in Signal Processing*, 2010.

[37] H. C. Wu, N. I. Wu, C. S. Tsai, and M. S. Hwang, “Image steganographic scheme based on pixel-value differencing and LSB replacement methods,” *IEE Proceedings-Vision, Image and Signal Processing*, vol. 152, no. 5, pp. 611-615, 2005.

[38] R. Z. Wang, C. F. Lin, and J. C. Lin, “Image hiding by optimal LSB substitution and genetic algorithm,” *Pattern Recognition*, vol. 34, no. 3, pp. 671–683, 2001.

[39] C. K. Chan and L. M. Cheng, “Hiding data in images by simple LSB substitution,” *Pattern Recognition*, vol. 37, no. 3, pp. 469–474, 2004.

[40] C. C. Chang and H. W. Tseng, “A steganographic method for digital images using side match,” *Pattern Recognition Letters*, vol. 25, no. 12, pp. 1431–1437, 2004.

[41] T. Kim, “Side match and overlap match vector quantizers for images,” *IEEE Transactions on Image Processing*, vol. 1, no. 2, pp. 170–185, 1992.

[42] K. C. Chang, C. P. Chang, P. S. Huang, and T. M. Tu, “A novel image steganographic method using tri-way pixel-value differencing,” *Journal of multimedia*, vol. 3, no. 2, pp. 37–44, 2008.

[43] K. C. Chang, P. S. Huang, T. M. Tu, and C. P. Chang, “Adaptive image steganographic scheme based on Tri-way Pixel-Value Differencing,” in *Proc. IEEE International Conference on Systems, Man and Cybernetics*, 2007.

[44] N. Zaker and A. Hamzeh, “A novel steganalysis for TPVD steganographic method based on differences of pixel difference histogram,” *Multimedia Tools and Applications*, vol. 58, no. 1, pp. 147–166, May 2012.

[45] K. H. Jung, K. J. Ha, and K. Y. Yoo, “Image data hiding method based on multi-pixel differencing and LSB substitution methods,” in *Proc. International Conference on Convergence and Hybrid Information Technology*, 2008.

[46] C. C. Chang, J. C. Chuang, and Y. C. Hu, “Spatial domain image hiding scheme using pixel-values differencing,” *Fundamenta Informaticae*, vol. 70, no. 3, pp. 171–184, 2006.

[47] M. Sonka, V. Hlavac, and R. Boyle, *Image Processing, Analysis, and Machine Vision, Third Edition*. Thomson Engineering, 2007.

[48] M. Heath, S. Sarkar, T. Sanocki, and K. Bowyer, “Comparison of edge detectors: a methodology and initial study,” in *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 1996.

[49] M. Thakkar and H. Shah, “Edge detection techniques using fuzzy thresholding,” in *Proc. World Congress on Information and Communication Technologies*, 2011.

[50] P. Vidya, S. Veni, and K. Narayanankutty, “Performance analysis of edge detection methods on hexagonal sampling grid,” *International Journal of Electronic Engineering Research*, vol. 1, no. 4, pp. 313–328, 2009.

[51] T. Chaira, “Image segmentation and color retrieval–a fuzzy and intuitionistic fuzzy set theoretic approach,” Ph.D. Dissertation, Indian Institute of Technology, Kharagpur, India, 2004.

[52] T. Chaira and A. Ray, “A new measure using intuitionistic fuzzy set theory and its application to edge detection,” *Applied Soft Computing*, vol. 8, no. 2, pp. 919–927, 2008.

[53] R. Amirtharajan, B. Bose, S. Imadabathuni, J. Bosco, and B. Rayappan, “Security building at the line of control for image stego,” *International Journal of Computer Applications*, vol. 12, no. 5, pp. 46–53, 2010.

[54] N. M. AL-Aidroos, M. H. Mohamed, and M. A. Bamatraf, “Data hiding technique based on dynamic LSB,” in *Proc. International Arab Conference on Information Technology*, Riyadh, 2011.

[55] W. J. Chen, C. C. Chang, and T. Le, “High payload steganography mechanism using hybrid edge detector,” *Expert Systems with Applications*, vol. 37, no. 4, pp. 3292–3301, 2010.

[56] M. H. Goodarzi, A. Zaeim, and A. S. Shahabi, “Convergence between fuzzy logic and steganography for high payload data embedding and more security,” in *Proc. 6th International Conference on Telecommunication Systems, Services, and Applications*, 2011.

[57] E.-S. M. El-Alfy and A. A. Al-Sadi, “A comparative study of PVD-based schemes for data hiding in digital images,” in *Proc. 9th IEEE/ACS International Conference on Computer Systems and Applications*, 2011.

[58] H. C. Wei, P. C. Tsai, and J. S. Wang, “Three-sided side match finite-state vector quantization,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 10, no. 1, pp. 51–58, 2000.

[59] A. T. Al-Taani and A. M. AL-Issa, “A novel steganographic method for gray-level images,” *International Journal of Computer, Information, and Systems Science and Engineering*, vol. 3, pp. 5–10, 2009.

[60] N. Boccara, *Modeling Complex Systems*. Springer Verlag, 2004.

[61] Y. Zhang, F. Zuo, Z. Zhai, and C. Xiaobin, “A new image encryption algorithm based on multiple chaos system,” in *Proc. International Symposium on Electronic Commerce and Security*, 2008.

[62] K. Satish, T. Jayakar, C. Tobin, K. Madhavi, and K. Murali, “Chaos based spread spectrum image steganography,” *IEEE Transactions on Consumer Electronics,* vol. 50, no. 2, pp. 587–590, 2004.

[63] L. Yu, Y. Zhao, R. Ni, and T. Li, “Improved adaptive LSB steganography based on chaos and genetic algorithm,” *EURASIP Journal on Advances in Signal Processing*, 2010.

[64] http://www.mathworks.com/matlabcentral/fileexchange/3675-wpsnr/content/wpsnr.m.

[65] M. Miyahara, K. Kotani, and V. R. Algazi “Objective picture quality scale (PQS) for image coding,” *IEEE Transactions on Communications*, vol 46, no.9, pp. 1215-1226, 1998.

[66] Z. Wang, A. C. Bovik, H. R. Sheikh, and E. P. Simoncelli, “Image quality assessment: From error visibility to structural similarity,” *IEEE Transactions on Image Processing*, vol. 13, no. 4, pp. 600-612, 2004.

[67] Z. Wang and A. C. Bovik, “Mean squared error: Love it or leave it?,” *IEEE Signal Processing Magazine*, pp. 98-11, 2009.

[68] G. Schaefer and M. Stich, “UCID - an uncompressed colour image database,” in *Proc. SPIE Conference on Storage and Retrieval Methods and Applications for Multimedia*, San Jose, USA, 2004.

[69] UCID - Uncompressed colour image database. [Online]. Available: http://vision.cs.aston.ac.uk/datasets/UCID/ucid.html. [Accessed: 13-Dec-2011].

[70] S. Cho, B.-H. Cha, J. Wang, and C.-C. J. Kuo, “Performance study on block-based image steganalysis,” in *Proc. IEEE International Symposium on Circuits and Systems*, 2011.

[71] S. Cho, J. Wang, C.-C. J. Kuo, and B.-H. Cha, “Block-based image steganalysis for a multi-classifier,” in *Proc. IEEE International Conference on Multimedia and Expo*, 2010.

[72] Mathworks Matlab R2010a, http://www.mathworks.com/products/matlab/

[73] T. Chaira and A. Ray, *Fuzzy Image Processing and Applications with Matlab*. CRC Press, 2010.

VITAE

- Azzat Ahmed Ali Al-Sadi
- Born in Mukalla, Yemen, on 3rd November, 1979
- Nationality: Yemeni
- Received B.S Information Engineering from Baghdad University in 2005
- Worked in Hadhramout University of Science and Technology as Lecturer from Nov,2006 to Nov,2008
- Joined King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia as a Master Student in February, 2009.
- Completed M.S in Computer Networks from King Fahd University of Petroleum and Minerals, Dhahran, Saudi Arabia in May 2012.
- E-mail: azzat.sadi@gmail.com
- Phone: 009675353166
- Present Address: P.O. Box 6332, King Fahd University of Petroleum and Minerals, Dhahran-31261, Saudi Arabia
- Permanent Address: October Street, Mukalla, Hadhramout, Yemen