Image Steganography Using Pixel Value Difference

Thesis submitted in partial fulfillment of the requirements for the award of degree of

Master of Technology
in
Computer Science and Applications

Submitted By
Vaibhav Arora
(Roll No. 601103028)

Under the supervision of
Singara Singh
Assistant Professor

SCHOOL OF MATHEMATICS AND COMPUTER APPLICATIONS
THAPAR UNIVERSITY
PATIALA – 147004

July 2013

Certificate

I hereby certify that the work which is being presented in the thesis entitled, **"Image Steganography Using Pixel Value Difference"**, in partial fulfillment of the requirements for the award of degree of Master of Technology in Computer Science and Applications submitted in School of Mathematics and Computer Applications(SMCA), Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Singara Singh** and refers other researcher's work which are duly listed in the reference section.

The matter presented in this thesis has not been submitted for award of any other degree of this or any other University.

Vaibhav.

(Vaibhav Arora)

This is to certify that the above statement made by the candidate is correct and true to the best of my knowledge.

15.7.13

(Singara Singh)
Assistant Professor
SMCA

Countersigned by:

(Dr. Rajesh Kumar)
Head
SMCA
Thapar University
Patiala

(Dr. S. K. Mohapatra)
Dean (Academic Affairs)
Thapar University
Patiala

Acknowledgement

First of all I would like to thank the Almighty, who has always guided me to work on the right path of the life.

This work would not have been possible without the encouragement and able guidance of my supervisor **Singara Singh**. I thank my supervisor for their time, patience, discussions and valuable comments. Their enthusiasm and optimism made this experience both rewarding and enjoyable.

I am equally grateful to **Dr. Rajesh Kumar**, Associate Professor and Head, SMCA, for motivation and inspiration that triggered me for the thesis work.

I will be failing in my duty if I don't express my gratitude to **Dr. S. K. Mohapatra,** Senior Professor and Dean of Academic Affairs, Thapar University, for making provisions of infrastructure such as library facilities, computer labs equipped with net facilities, immensely useful for the learners to equip themselves with the latest in the field.

I am also thankful to the entire faculty and staff members of SMCA for their direct-indirect help, cooperation, love and affection, which made my stay at Thapar University memorable.

Last but not least, I would like to thank my parents for their wonderful love and encouragement, without their blessings none of this would have been possible. I would also like to thank my close friends for their constant support.

Vaibhav Arora

(601103028)

ABSTRACT

In this work, steganography algorithm for images is proposed. There are five chapters in this thesis. In Chapter 1, introduction related to the image steganography is discussed. In Chapter 2, literature review of steganography using pixel value difference using spatial domain and transform domain is discussed. In Chapter 3, a steganography algorithm for images is proposed. The maximum *PSNR* is 37.554 dB and maximum hiding capacity is 32,883 bits. In Chapter 4, steganalysis of the proposed algorithm is performed. Two steganalysis named as Histogram steganalysis and Chi-square steganalysis are performed on stego images to show the effectiveness of the proposed algorithm. Conclusion and future work are discussed in Chapter 5.

LIST OF TABLES

LIST OF FIGURES

ABBREVIATIONS

BMP	Bitmap
BC	Before Christ
DCT	Discrete Cosine Transform
GIF	Graphic Interchange Format
JPEG	Joint Photographic Expert Group
LSB	Least Significant Bit
MSB	Most Significant Bit
OPAP	Optimal Pixel Adjustment Process
PICT	Picture
PSNR	Peak Signal to Noise Ratio
PVD	Pixel Value Difference
TIFF	Tagged Image File Format
TPVD	Tri-way Pixel Value Difference
SDS	Single Digit Sum

TABLE OF CONTENTS

CHAPTER 1 INTRODUCTION

1.1Introduction

Information hiding techniques have gained importance in many applications areas. Previously there used to be a thought that communication can be secured by encrypting the traffic. But it is not adequate if considered alone. For example, encrypted communication between a defence employee and embassy of some other nation has obvious implications. So the study of communications security includes not just encryption but also information hiding. Applications of information hiding are:

- Main application is in copyrights protection. The audio and video works once available in digital form can be used to make copies. This leads to large unauthorized copying which is great threat to the music, film, book publishing companies. Digital watermarks and fingerprints are used to identify copyright violators and prosecute them.
- Law enforcement and counter intelligence agencies concentrate on detecting the presence of the messages and trace them.
- Criminals also place great interest in unobtrusive communication.
- Some governments have decided not to allow the civilians to use cryptography and to put limit on line speech. This made people to look for some methods which would allow some sort of anonymous communication.

1.2 Steganography

Steganography is the art of secret communication. Steganography is a new method of making the communication more secured. These techniques gained importance in a many diverse areas. Steganography forms an important sub discipline of information hiding. Cryptography scrambles messages so that they cannot be understood. Steganography is

used to hide the very presence of message. A message in cipher text will arouse some suspicion at the recipient. But a file hidden in cover medium using some steganographic tool, remains invisible. Although cryptography and steganography are too different methodologies, we can borrow some techniques from former. Data transmissions are fundamental and became a essential these days. Whether you contribute to your secret messages with your close contacts, or company secrets with your partners, you must defend your secret information from hackers, your chief, or your competitors. However we exist in an insecure world where unnecessary people can access your personal information (like e-mails or personal documents) and often use it in opposition to you. The word steganography is plagiaristic from the Greek words “stegos” meaning “cover” and “grafia” meaning “writing”. Cryptography and Steganography are cousins in the spy skill relatives. Cryptography scrambles a message by using certain cryptographic algorithms for converting the secret data into incomprehensible form. On the former hand, Steganography hides the message so that it cannot be seen. A message in cipher text might arouse suspicion on the part of the recipient while an ”invisible” message created with steganographic methods will not. Somebody appealing in secret communication can forever apply a cryptographic algorithm to the data before embedding it to accomplish additional security.

Steganography has three main goals:

i) Security – is the secreted data distinguishable by either a person otherwise a computer, as well as

ii) Capacity – how much data can be hidden in a given cover file.

iii) Robustness- the resilience of your hidden data to image/audio manipulation.

These three goals are often in conflict. The additional data you hide, the more likely it is to be found, and i.e. it has less security and vice versa. A third goal, robustness, is what separates steganography from watermarking (a 2nd sub-discipline of information hiding). Robustness is the resilience of your hidden data to image/audio manipulation such as contrast, brightness, cropping, stretching, analog-to-digital-to-analog conversion, etc. There is a large commercial interest in watermarking for digital rights management. Since there is also a trade-off between robustness and capacity, steganographic programs often do not attempt to be robust, and the techniques presented here are no exception.

Another form of data hiding in digital images is Watermarking. Digital watermarking is the process of embedding auxiliary information into a digital cover signal with the aim of providing authentication information. A watermark is called robust with respect to a class of transformations if the embedded information can reliably be detected from the marked signal even if degraded by any transformation within that class. Typical image degradations are *JPEG* compression, rotation, cropping, additive noise and quantization.

Watermarking is a data hiding technique that protects digital documents, files, or images against removal of copyright information. Even if someone knows that a watermark is exist (i.e. visible watermarking) in a given object, it should be impossible to remove the watermark from the watermarked object without causing a distortion or destroying the original (watermarked) object. This aspect or feature of watermarking is known as "robustness". According to the kind of embedded information, two techniques of document marking can be distinguished:-watermarking and fingerprinting. Watermarking is the process of embedding a specific copyright mark into digital documents in the same way. On the other hand, in order to detect any break of licensing agreement, a serial number is embedded in every copy of this digital document. This process is known as "fingerprinting". Even if these markings are detected, it should be practically impossible to remove them.

The key difference between steganography and watermarking is the absence (in steganography) of an active adversary mainly because usually no value is associated with the act of removing the information hidden in the host content. Nevertheless, steganography may need to be robust against accidental or common distortion like compressions or color adjustment. Image steganography is defined as hiding a secret message within an image in such a way that others cannot discern the presence or contents of the hidden message. For example, a message might be hidden within an image by changing the Least Significant Bits (*LSB*) to be the message bits. By embedding a secret message into a carrier image, a stego-image is obtained. It is important that the stego-image does not contain any easily detectable artifacts due to message embedding that could be detected by electronic surveillance. One could utilise those artifacts to detect images that contain secret messages. Once this is achieved, the steganographic tool becomes useless.

Images are composed of picture elements, i.e. pixels. There are three major classes of images:

i) Black and white – each pixel is composed of a single bit and is either a zero or a one, representing either white or black.

ii) Greyscale – each pixel is composed of 8 bits (in rare cases, 16 bits) which defines the shade of grey of the pixel, from zero (black) to 255 (white).

iii) Full color – also called 24-bit color as there are 3 primary colors (red, green, blue), each of which is defined by 8 bits. There are over 16 million possible colors.

There exist many other representations, but these three, by far, are the most common. For the steganography techniques presented here, we will either use grayscale or 24-bit color. Considering 8-bit grayscale, the Most Significant Bit (*MSB*) contributes 1/2 the information, while the Least Significant Bit contributes $1/256^{th}$ of the information. So, changing that *LSB* only affects $1/256^{th}$ of the intensity and humans simply cannot perceive a difference. In fact, it is difficult to perceive a difference in $1/16^{th}$ of an intensity change, so we can alter the 4 *LSBs* with little or no perceptible difference.

As mentioned, steganography deals with hiding of information in some cover source. On the other hand, Steganalysis is the art and science of detecting messages hidden using steganography, this is analogous to cryptanalysis applied to cryptography. The goal of steganalysis is to identify suspected packages, determine whether or not they have a payload encoded into them, and, if possible, recover that payload.

Hence, the major challenges of effective steganography are:-

i) Security of Hidden Communication: In order to avoid raising the suspicions of eavesdroppers, while evading the meticulous screening of algorithmic detection, the hidden contents must be invisible both perceptually and statistically.

ii) Capacity: Unlike watermarking, which needs to embed only a small amount of copyright information, steganography aims at hidden communication and therefore usually requires sufficient embedding capacity. Requirements for higher payload and secure communication are often contradictory. Depending on the specific application scenarios, a trade off has to be sought. Higher capacity means naïve steganography as shown in Figure 1.1.

iii) Robustness: The resilience of your hidden data to image/audio manipulation.

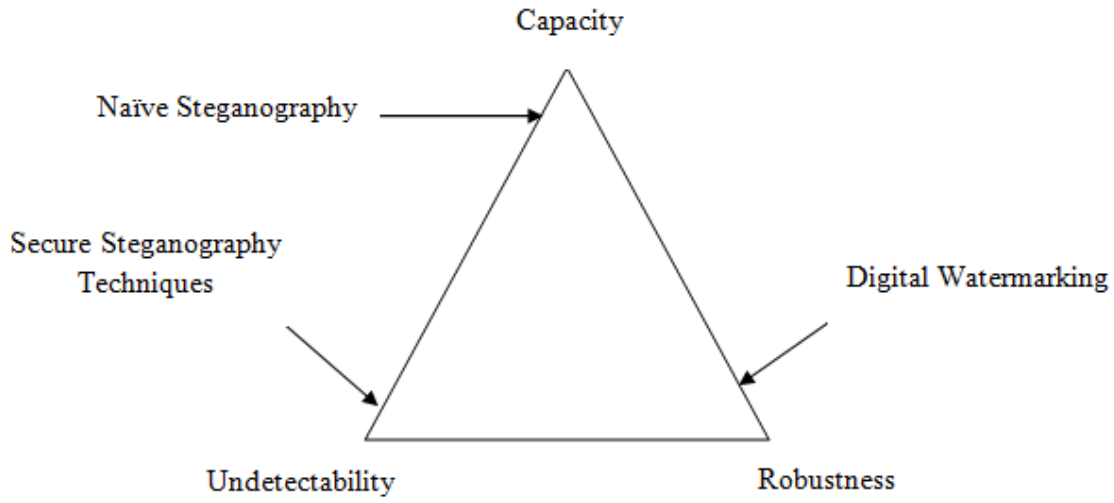


Figure 1.1 Trade off between embedding capacity, undetectability and robustness in data hiding.

1.3 History of Steganography

Steganography appeared before cryptography. In 474 Before Christ (*BC*), Greek historian Herodotus detailed how countrymen exchanged what appeared to be blank wax tablets. Underneath the wax, wood bases were scratched with secret messages. While exiled in Persia, Demeratus discovered that Greece was about to be invaded and wanted to convey a message of warning. However, the risk of exposure was great for him, so he concealed his message by writing directly on the wood and then covering it with wax. The seemingly blank tablets were then transported to Sparta where the message was literally uncovered and his allies forewarned.

Another example is that of the ancient Greek Histiaeus, who wished to inform his allies when to revolt against the enemy. To do so, he shaved the head of a trusted servant and then tattooed a message on his scalp. After allowing time for the slave's hair to grow back, he was sent through enemy territory to the allies. To the observer, the slave appeared to be a harmless traveller passing through the area. However, upon arrival, the slave reported to the leader of the allies and indicated that his head should be shaved,

thereby revealing the message. Recent times have yielded more advanced techniques, such as the use of invisible ink, where messages are written using substances that subsequently disappear. The hidden message is later revealed using heat or certain chemical reactions. Other methods may employ routine correspondence, such as the application of pinpricks in the vicinity of particular letters to spell out a secret message. Advances in photography produced microfilm that was used to transmit messages via carrier pigeon. Further developments in this area improved film and lenses that provided the ability to reduce the size of secret messages to a printed period. The Germans in World War II used this technique, known as the microdot.

With today's communications moving increasingly to electronic means, digital multimedia signals (typically audio, video, or still imagery) are being used as vehicles for steganographic communication.

Another method used in Greece was to select messengers and shave their head. They would then write a message on their head. Once the message had been written the hair was allowed to grow back. After the hair grew back the messenger was sent to deliver the message, the recipient would shave off the messenger's hair to see the secret. Now that some 2000 years have passed, people have turned to digital technologies to help them hide sensitive data. Radio and *TV* messages, from World War II to today, can be used to hide coded or hidden messages. Some government sources suspect that Osama bin Laden's prerecorded videos that are re-played on *TV* stations around the world contain hidden messages [3].

Steganography methods have two different algorithms. One is embedding the messages. Other is for extraction of the embedded messages. For hiding data in images, there is need of two files: cover image and hide file. The file to be hidden can be plain text, image data, audio data or video file. The image acquired after hiding the message in the cover image is called as the stego image. The method will be secure if stego image must not show any detectable artifacts due to message embedding. This implies that the stego image should have the same statistical properties as cover image. Information can be hidden in the images in many ways. We may find out the areas which will not draw much attention-those areas which have great deal of color variation. We may also choose to put

the message bits in random fashion throughput the image [2]. Common approaches may be classified as:

i) Least significant bit substitution

This is most widely used method for hiding images. Here the *LSB* of the color value pixel is replaced by a message bit. This method loses its effect if the image is covered from lossless format to lossy format (*JPEG*) and then image is built back into lossless format. Here we lose the message bits as the *LSBs* get modified.
Let us consider a 24-bit cover image and assume we want to embed message in *LSBs* of each of the bytes. A 1024×768 image has the capacity of hiding 2,359,296 bits (294,912 bytes). But in this case we need to choose the images more carefully .We should take care that only 10 percent of the total available capacity of the image is used for hiding in order to avoid any suspicion. These pixels may be randomly selected.

ii) Masking and filtering

These techniques are usually restricted to 24-bit and gray scale images. Here information may be hidden in a way similar to the paper watermarking. In steganography the object of communication is the cover. Masking techniques embed information in significant areas so that hidden message becomes more integral to the cover image.

iii) Transformation based embedding

LSB manipulation is an easy and quick method of steganography. But it is vulnerable to the small changes resulting from image processing or lossy compression. So alternatively we can look for the *JPEG* images. *JPEG* standard uses discrete cosine transforms to achieve compression. *DCT* is lossy compression transform. This is because the cosine values cannot be calculated exactly every time. There are always rounding errors when calculated using limited precision numbers. In addition to the *DCT*, we may use Fourier

transforms and wavelet transforms. Manipulating the *DCT* coefficients in a proper manner can help us in hiding message in *JPEG* images.

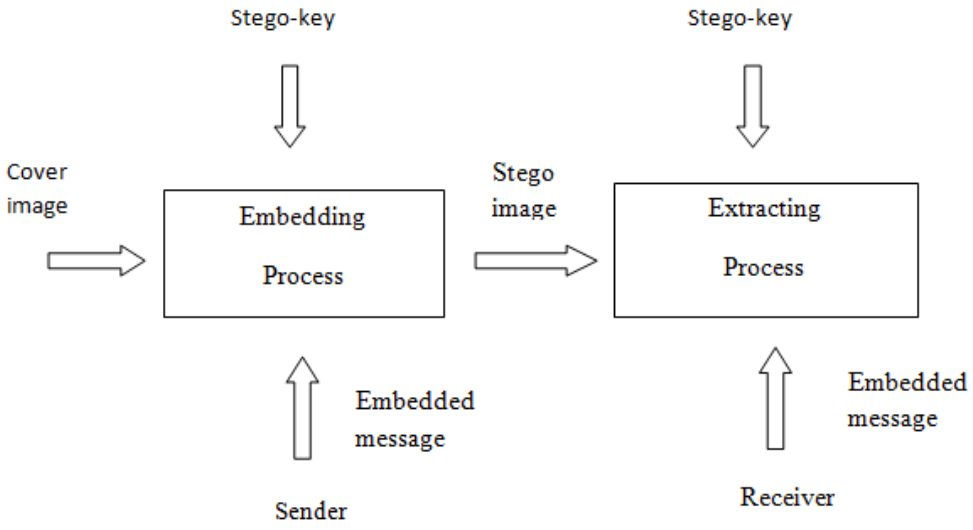


Figure 1.2 A generalized steganographic framework.

Figure 1.2 shows a general steganographic system. To embed secret data, a key is required and an embedding process hides the data into a cover image to generate a stego image. On the receiver side, the extraction process takes the stego image and inverse of the embedding algorithm is applied to extract the hidden message. Also the same stego key is required in this extraction process. This system can be explained using the 'prisoners problem' (Figure 1.3) where Alice and Bob are two inmates who wish to communicate in order to hatch an escape plan. However communication between them is examined by the warden, Wendy. To send the secret message to Bob, Alice embeds the secret message 'm' into the cover object 'c', to obtain the stego object 's'. The stego object is then sent through the public channel. In a pure steganographic framework, the technique for embedding the message is unknown to Wendy and shared as a secret between Alice and Bob. In private key steganography Alice and Bob share a secret key which is used to embed the message. The secret key, for example, can be a password used to seed a pseudo-random number generator to select pixel locations in an image

cover-object for embedding the secret message. Wendy has no knowledge about the secret key that Alice and Bob share, although she is aware of the algorithm that they could be employing for embedding messages. In public key steganography, Alice and Bob have private-public key pairs and know each other's public key. In this thesis we confine ourselves to private key steganography only.

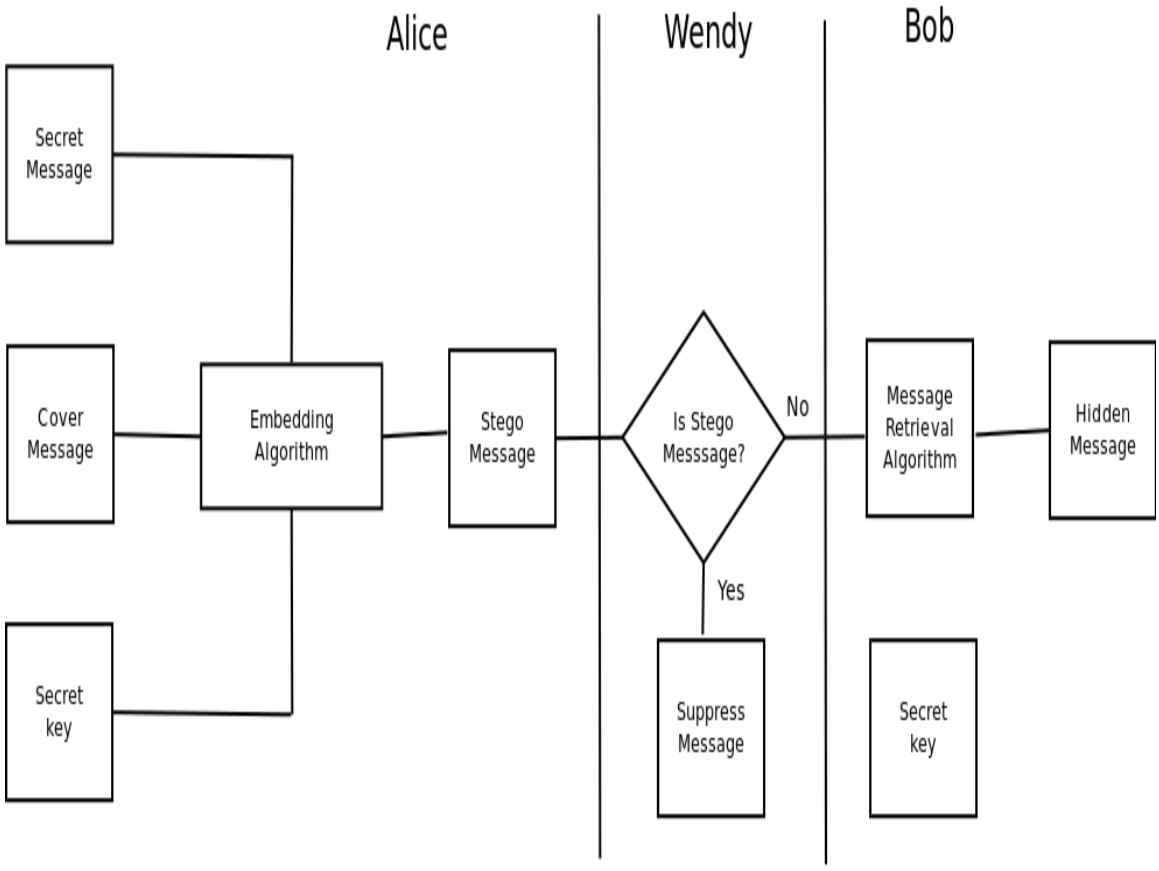


Figure 1.3 Framework for Private Key Passive Warden Steganography.

The driving force behind steganography is that a data can't be intercepted unless its existence is discovered. Steganography techniques focus on making changes to the host file in a manner that is both invisible and undetectable. An embedding algorithm said to be invisible if humans can't visually perceive whether an image contains information or it does not. In psycho-visual experiments, invisibility is commonly judged by presenting a

large number of images and asking the person to determine which images contain hidden data. A success ratio of 50% indicates that the person cannot determine which images contain hidden data. The detectability of an embedding algorithm relates to whether a computer can detect these changes and confirm the existence of embedded data.

CHAPTER 2 LITERATURE SURVEY

2.1 Introduction

In this chapter, the literature review of steganography techniques using pixel value differences is discussed. Section 2.2 includes brief description of the existing steganographic techniques. In Section 2.3 literature review related to image steganography techniques based on pixel value differences is discussed in brief.

2.2 Types of Steganography

The steganographic algorithms can broadly be classified into two categories.
(i) Spatial Domain Steganography Techniques
(ii) Transform Domain Steganography Techniques
Each of these techniques is discussed in detail in following subsections.

2.2.1 Spatial Domain Steganography Techniques

These techniques make use of the pixel gray level and their color standards directly for encoding the message bits. These techniques are several of the simplest schemes in terms of embedding and extraction complexity. The chief disadvantage of these methods is amount of preservative noise that creeps in the image which unswervingly affects the Peak Signal to Noise Ratio and the statistical properties of the image. Furthermore these embedding algorithms are appropriate mainly to lossless image compression schemes like tagged image file format (*TIFF*) images. For lossy compression scheme similar to *JPEG*, several of the message bits get lost for the period of the compression step.

The most ordinary algorithm belong to this category of techniques is the Least Significant Bit substitute technique in which the least significant bit of the binary sign of the pixel gray level is used to signify the message bit. This sort of embedding leads to count of a noise of 0:5p on average in the pixels of the image where p is the embedding charge in bits/pixel. This sort of embedding also leads to an irregularity and a combination in the pixel gray values (0,1); (2,3); (4,5) ;.. . . (254,255). This irregularity is exploited in the attacks developed for this method as explained further in section 2.2. To beat this uninvited irregularity, the assessment of changing the least significant bit is randomized i.e. if the message bit does not equivalent the pixel bit, then pixel bit is either decreased or increased by 1.

This method is commonly famous as *LSB* Matching. It can be experiential that even this sort of embedding adds a noise of 0:5p on average. To additional reduce the noise, [2] have recommended the use of a binary function of two cover pixels to embed the data bits. A pair of pixels as a unit is embeds, where the *LSB* of the first pixel carry one bit of information, and a gathering of the two pixel values carries a dissimilar bit of information. It has been made known that embedding in this manner reduces the embedding noise introduced in the cover signal.

In [4], a manifold base number system has been engaged for embedding data bits. At the same time as embedding, the human visualization sensitivity has been taken concern of. The difference value for a chunk of pixels is used to calculate the number base to be used for embedding. A comparable sort of algorithm based on human visualization sensitivity has been proposed by [5] by the name of Pixel Value Differencing. This method is based on adding more amount of data bits in the high variance regions of the image for example close to “the edges” by taking into consideration the difference values of two neighbouring pixels. This method has been enhanced further by clubbing it with least significant bit embedding in [6].

LSB substitute technique has been comprehensive to multiple bit planes too. Recently [3] has claimed that *LSB* replacement linking more than one least significant bit planes is fewer noticeable than single bit plane *LSB* replacement. Therefore the use of various bit planes for embedding has been buoyant. However the direct use of 3 or additional bit planes leads to calculation of considerable amount of noise in the cover image. [7] and

[8] have specified in depth study of the noise added by the *LSB* embedding in 3 bit planes. In addition, a latest algorithm which uses an arrangement of Single Digit Sum Function and Matrix Encoding has been wished-for. It has been exposed systematically that the noise added by the planned algorithm in a pixel of the image is 0:75 *p* as compared to 0:875 *p* added by 3 plane *LSB* embedding where *p* is the embedding rate.

2.2.2 Transform Domain Steganography Techniques

Generally these techniques encode message bits in the transform domain coefficients of the image. Content embedding task in the transform domain is extensively used for vigorous watermarking .Comparable techniques can also recognize large-capacity embedding for steganography. Discrete cosine Transform, discrete wavelet transform, and discrete Fourier transform integrated in Candidate Transform. Through being embedded in the transform domain, the hidden data resides in more strong areas, extend across the whole image, and provides better conflict against signal processing. For example, we can achieve a block *DCT* and, depending on payload and healthiness requirements, choose one otherwise more components in each block to form a fresh data cluster that, in turn, is pseudo arbitrarily twisted and undergoes a second-layer transformation. Adjustment is then agreed out on the double transform domain coefficients using a variety of schemes. Also these techniques have high embedding and extraction complexity. The transform domain embedding techniques are generally more applicable to the "Watermarking" aspect of data hiding due to its robustness property. A lot of steganographic techniques in this domain have been encouraged from their watermarking counterparts.

F5 [19] uses the Discrete Cosine Transform coefficients of an image for embedding data bits. F5 embeds data in the *DCT* coefficients by rounding the quantized coefficients to the nearest data bit. It also uses Matrix Encoding for reducing the embedded noise in the signal. F5 is one the most popular embedding schemes in *DCT* domain steganography, though it has been successfully broken in [19].

The transform domain embedding does not necessarily mean generating the transform coefficients on blocks of size 8×8 as done in *JPEG* compression techniques. It is possible to design techniques which take the transforms on the whole image. Other block based *JPEG* domain and pixel value difference based embedding algorithms have been proposed in [11] and [21] respectively.

2.3 Literature Review

Wu *et al.* [20] proposed the pixel-value differencing (*PVD*) and *LSB* replacement method. In order to improve the capacity of the hidden secret data and to provide an imperceptible stego-image quality, a novel steganographic method based on least-significant-bit (LSB) replacement and pixel-value differencing (*PVD*) method is presented. First, a different value from two consecutive pixels by utilising the *PVD* method is obtained. A small difference value can be located on a smooth area and the large one is located on an edged area. In the smooth areas, the secret data is hidden into the cover image by LSB method

In *DHPVD*, insertion is made by choosing 2×2 image mask from the host image in row major order. The dimension of the hidden image is extracted first and embedded into the host image. Three bits of each byte from the dimension of the hidden image is embedded per byte of the source image into the rightmost three bits from *LSB*. Though the size of the secret image can be about a half of the host image with degraded image quality of up to 42 dB (PSNR), their method restricts that the secret data should be in the form of natural images. The method of Wu *et al*., [20] performs digit number transformation and hides resulting digits by adding them to pixels of the host images. Stego [20], one of the several widely available software packages, simply encodes data in the least significant bit (*LSB*) of the host signal. The information about the number of embedded *LSBs* for each pixel is basically self-contained in its intensity value via an LSB-mapping function and requires no extra bits for later data extraction. Their method achieves the highest hiding efficiency under the constraint of human visual perception.

Author used three types of data: i) random bit stream, ii) image data bit stream and iii) *JPEG* bit stream, for embedding into two popular images “Lena” and “Jet”. The

embedded data are found to be within 31% - 45% volume of the host image, with the *PSNR* ranging from 40 dB - 33 dB.

Extensive analysis has been made on various images using *DHPVD* technique. Simulation is done in Linux environment C language, RAM capacity 2GB, Hard disk capacity is 160 GB. This section represents the results, discussion and comparative study between *DHPVD*.

Sur *et al*., [7] proposed a new special domain encoding method to reduce the amount of noise added during embedding. Their proposed based scheme uses Single digit sum encoding to improve the embedding efficiency. Their method is especially suitable when more than two bit planes are used for embedding. Embedding is done in multiple *LSB* planes using single digit sum encoding (*SDS*). They noted that the amount of noise added due to steganographic embedding depends upon the number of changes made in the cover signal. This is an important consideration in the design of embedding algorithms, since the noise added effects the statistical properties of a medium. For a given medium, the steganographic algorithm which makes fewer embedding changes or adds less additive noise will be less detectable as compared to an algorithm which makes relatively more changes or adds higher additives.

Changa *et al*., [10] proposed to enlarge the capacity of the hidden secret information and to provide an imperceptible stego-image for human vision, a novel steganographic approach using tri-way pixel-value differencing (*TPVD*) .To upgrade the hiding capacity of original *PVD* method referring to only one direction, three different directional edges are considered and effectively adopted to design the scheme of tri-way pixel-value differencing. In addition, to reduce the quality distortion of stego-image brought from setting larger embedding capacity, an optimal approach of selecting the reference point and adaptive rules are presented. A minimum peak signal-to noise ratio (*PSNR*) value of 38 dB is adopted as the quality requirement for the stego-images in the experiments. A piecewise mapping function according to human visual sensitivity of contrast is used so that adaptively can be achieved without extra bits for overhead The leading information for data decoding is few no more than 3 bytes. Experiments show that a large amount of bit streams (near& 30%-45% of the host image) can be embedded without sever degradation of the image quality (33-40 dB, depending on the volume of embedded bits).

They also presented to improve the capacity of the hidden secret data and to provide an imperceptible stego-image quality, a novel steganographic method based on least-significant-bit replacement and pixel-value differencing method. First, a different value from two consecutive pixels by utilising the *PVD* method is obtained. A small difference value can be located on a smooth area and the large one is located on an edged area. In the smooth areas, the secret data is hidden into the cover image by *LSB* method while using the *PVD* method in the edged areas. Because the range width is variable, and the area in which the secret data is concealed by *LSB* or *PVD* method are hard to guess, the security level is the same as that of a single using the *PVD* method of the proposed method. Compared with the *PVD* method being used alone, the proposed method can hide much larger information and maintains a good visual quality of stego-image.

They expanded the *LSB* matching revisited image steganography and propose an edge adaptive scheme which can select the embedding regions according to the size of secret message and the difference between two consecutive pixels in the cover image. For lower embedding rates, only sharper edge regions are used while keeping the other smoother regions as they are. When the embedding rate increases, more edge regions can be released adaptively for data hiding by adjusting just a few parameters. The experimental results evaluated on 6000 natural images with three specific and four universal steganalytic algorithms show that the new scheme can enhance the security significantly compared with typical *LSB*-based approaches as well as their edge adaptive ones, such as pixel-value-differencing-based approaches, while preserving higher visual quality of stego images at the same time.

LSB matching (*LSBM*) employs a minor modification to *LSB* replacement. If the secret bit does not match the *LSB* of the cover image, then it is randomly added to the corresponding pixel value. Statistically, the probability of increasing or decreasing for each modified pixel value is the same and so the obvious asymmetry artifacts introduced by *LSB* replacement can be easily avoided. Therefore, the common approaches used to detect *LSB* replacement are totally ineffective at detecting the *LSBM*. Up to now, several steganalytic algorithms have been proposed to analyze the *LSBM* scheme.

Zhang *et al*. [24] discussed Image Steganography using Pixel-Value Differencing which is used to increase the capacity of the hidden secret information and to provide a stego-

image imperceptible for human vision; a novel steganographic approach based on pixel-value differencing is presented. Their approach uses the largest difference value between the other three pixels close to the target pixel to estimate how many secret bits will be embedded into the pixel. The theoretical estimation and experimental results demonstrate that the proposed steganography scheme can provide a superior embedding. Besides, the embedded confidential information can be extracted from stego-images without the assistance of original images. In this experiment, a random generated bit-stream is embedded into three classic host-images namely Lena, Baboon and Airplane. The secret binary data is generated by pseudo-random numbers. Several experiments were performed to evaluate in this method. Test images applied to the experiment are gray-scale images with size of 256 × 256.

Xin *et al*. [21] gives to improve the embedding capacity and provide an imperceptible visual quality, a novel steganographic method based on four-pixel differencing and modified least significant bit substitution is presented. The average difference value of a four-pixel block is exploited to classify the block as a smooth area or an edge area. Secret data are hidden into each pixel by the k-bit modified *LSB* substitution method, where k is decided by the level which the average difference value falls into. Readjustment will be executed to guarantee the same level that the average difference value belongs to before and after embedding, and to minimize the perceptual distortion. By proving that the readjusting procedure works, a theoretical proof is given to justify our method succeeded in embedding and extracting. Their experimental results have shown that the proposed method not only has an acceptable image quality but also provides a large embedding capacity.

Mandal *et al*. [13] proposed pixel value differencing based image authentication/data hiding technique through steganographic approach in spatial domain, termed as *DHPVD*. The numbers of secret bits of the hidden image that are embedded in consecutive two pixels of the source image depend on the difference between the pixels. More number of secret bits is inserted to the edge areas than smooth areas. 2 × 2 non overlapping mask is chosen from the source image in row major order. This scheme takes the difference between the two consecutive pixels. The difference may fall into any one of four levels such as lower, middle1, middle2 and higher. Then depending upon the difference level

variable numbers of secret bits is embedded using a hash function in the consecutive two pixels in the non overlapping 2×2 mask. In addition to embedding the contents of the hidden image, dimension of the hidden image has also been embedded. A bit handling is used to minimize the difference between the source and embedded pixels.

CHAPTER3 PROPOSED ALGORITHM

3.1 Introduction

Steganography algorithms embed the secrete information into different types of natural cover data like sound, images. The resulting altered data is referred to as stego-data and it must be perceptually indistinguishable from its natural cover. Steganography includes the costume of digital information within computer files/images. Normally, a steganographic message will appear to be something else, may be picture, video, sound file even the radio communication. Security may be achieved by hiding data within the image. Data hiding in the image has become an important technique for image authentication. So this chapter explains the proposed algorithm which is discussed in detail.

3.2 Proposed Algorithm

Insertion using *PVD* is made by choosing 2×2 image mask from the host image in row major order. The dimension of the hidden image is extracted first and set into the host image. Three bits of each byte from the dimension of the hidden image is embedded per byte of the source image into the rightmost three bits from *LSB.*

The hidden image pixels are embedded as follows. Two diagonally pixels from the 2×2 mask are taken to determine the difference (d). The difference value (d) may lies in any one of the four levels:

- If d<= 8, then difference lies within lower level.
- If 8<d<= 16, then difference lies in middle level 1.
- If 16<d<= 24, then difference lies in middle level 2.
- If d>24, then difference lies in higher level.

Three, four, five and six secret bits are embedded in the lower level, middle level 1, middle level 2 and higher level difference respectively. Difference expansion is a method

for reversibly hiding data in pixel images. The basic idea of difference expansion is to utilize the high correlation of the cover data. Given a pair of adjacent elements of highly correlated cover data (denoted by x1 and x2, which are both integers), an integer transform is defined to calculate their difference (d) and integer-mean (m), which is shown in (i) The transform is strictly invertible and (ii) is its inverse transform.

$$d = x1 - x2, m = \frac{floor((x1 + x2)}{2} \quad \ldots 3.1$$

$$x1 = m + floor\left(\frac{(d+1)}{2}\right),\ x2 = m - floor\left(\frac{d}{2}\right) \quad \ldots . 3.2$$

Based on the difference d in which level it falls into, the number of secret bits of the authenticating image are embedded into the cover image. To hide data bits, x1 and x2 are first transformed into d and m using equation 3.1. The high correlation of the cover data means that two elements x1 and x2 are generally very close (i.e., their difference d could be very small in most cases). So it is possible to embed secret bits onto d while keeping the induced distortions acceptable by bit handling procedure. Suppose the expanded difference carrying hidden bits is denoted as d1, the embedded elements x11 and x22 can be calculated using equation 3.2.

Phases involved in Algorithm: This Algorithm involves two phases (i) Insertion phase (ii) Extraction phase. Steps involved in these two phases are discussed below.

(i) Steps for insertion phase

Input: Host image of size p×q, hidden image of size m×n.

Output: Embedded image of size p×q.

Step 1: Acquire the dimension of the hidden image m×n.

Step 2: For each hidden message / image, examine source image non overlapping block of size 2×2 in row major order. Extract hidden message/ image dimension bits one by one. Substitute the hidden message/image bit in the rightmost 3 bits within the block, three bits in each byte.

Step 3: Read one character/ pixel of the hidden message/ image.

Step 4: Read source image block of size 2×2 in row major order and determine difference (d) and mean (m) using equation 3.1.

Step 5: Find out the level in which d belongs to; embed k secret bits onto the difference, based on the level and produced modified difference d1.

Step 6: A bit handling method is performed on d1 if required. The new value of embedded pixels (x11 and x22) is determined from d1 and m using equation 3.2**.**

Step 7: If the difference among the source (x1) and embedded pixel (x11) i.e. (x11-x1) is bigger than the difference among the source (x1) and embedded pixel (x22) that is (x22-x1) then x11 and x22 will be swapped to keep intact the visibility of the embedded image.

Step 8: Repeat step 2 to step 7 while all 2×2 matrix are not covered from m×n.

(ii)Steps for Extraction phase

Input: Embedded image of size p×q.

Output: Host image of size p×q, hidden image of size m×n.

Step 1: Read the image block of size 2×2 in row major order for each block, extract the hidden image dimension from the rightmost 3 bits of each byte replace hidden message/ image bit position in the block by '1'. For extraction of each 8 (eight) bits one character/ one image pixel is constructed.

Step 2: Read the image block of size 2×2 in row major order for each block, determine the difference (dd) and mean (mm) of two consecutive pixels of the non overlapping blocks.

Step 3: Determine the level to which dd belongs to. Extract hidden bits from the rightmost k bits of difference where the value of k depends on the level. For each 8 (eight)/24 (twenty four) bits extraction construct one character/ one image pixel.

Step 4: Repeat step 2 and 3 to complete decoding as per size of the hidden image m×n.

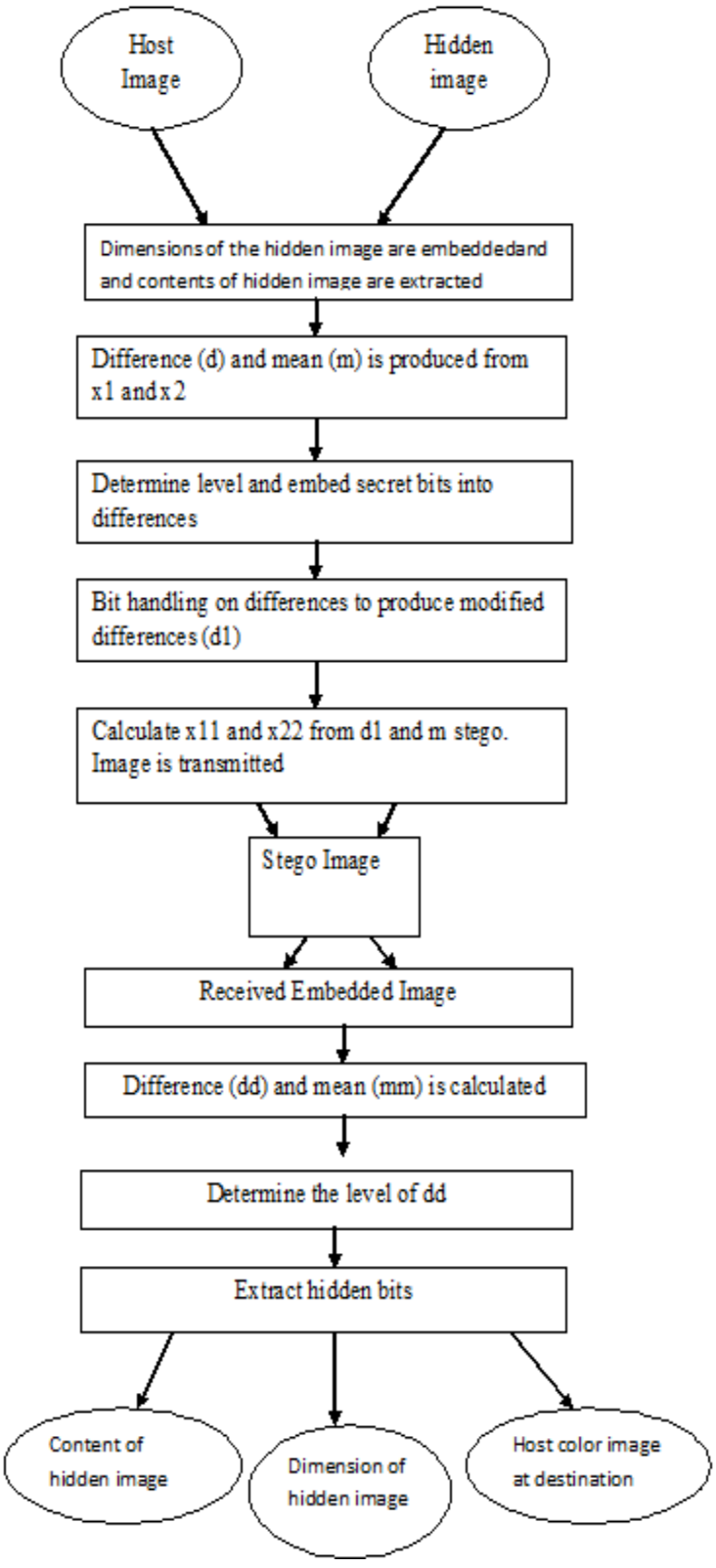


Figure 3.1 Flow chart for data hiding using *PVD*.

Flow chart of proposed algorithm is shown in figure3.1. Host image and hidden image as input and after whole process of data hiding perform over cover image we get output as content of hidden image, dimension of hidden image and host color image at destination.

3.3 Numerical Example

To explain how this algorithm will work, let's take an example. Consider pixel value bits of Cameraman image to be inserted into each mask of Zelda image. B bits of the Cameraman image is inserted into the Zelda image (source image). Insertion is done through difference value of two diagonally pixels and mean as shown in figure 3.1.

10110001 10100101 10010110

Figure 3.2(a) Pixel bits of Cameraman image

190	133
192	180

Source image block1

148	140
152	156

Source image block2

Figure 3.2(b) Source image block Zelda

00001010
00111011

Difference image block1

00001000
00001100

Difference image block2

Figure 3.2(c) Difference image block Zelda

00001101	00001101
00111000	00000100
Embedded Difference image block1	**Embedded Difference image block2**

Figure 3.2(d) Embedded Difference image Zelda

185	152
163	146
Mean image block1	**Mean image block2**

Figure 3.2(e) Mean calculated between pixel

100 086	083 070
110 121	067 079
Stego image block1	**Stego image block2**

Figure 3.2(f) Stego Zelda image

This is the numerical explanation to hide bits into appropriate pixel. Figure 3.2(f) shows the final result of 2×2 matrix or block after hiding secret bits into cover part.

Figure 3.3(a) Cover Zelda image **Figure 3.3(b) Stego Zelda image**

Table 3.1 PSNR between Cover Images and Stego Images, Capacity.

COVER IMAGE	**HIDDEN IMAGE**	**PSNR(db)**	**CAPACITY (bytes)**
Lena	Boat	34.5631	31110
Peppers	Sailboat	26.7162	32646
Zelda	Cameraman	34.2370	32883
Cameraman	Peppers	30.8890	31959
Living room	Baboon	25.3360	31109
Boat	Zelda	37.5542	31724

Sailboat	Woman darkhair	28.3821	32840
Pirate	Walkbridge	37.4543	32861
Woman darkhair	Boat	32.6898	32645
Jet plane	Lake	29.4570	31115
Walkbridge	Baboon	31.3373	32454
Boat	Lake	34.8562	31358
Lake	Pirate	37.4521	32881
Airplane	Zelda	32.5846	31268
Cameraman	Sailboat	34.4522	31880

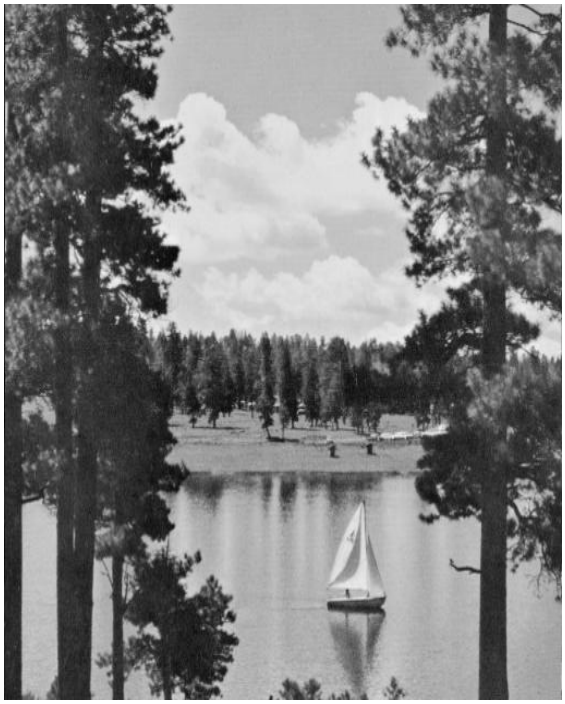
Table 3.1 shows the *PSNR* values for each embedding image against the source image. *PSNR* computes the peak signal to noise ratio.A 512×512 cover image has the capacity of hiding 32,883 bits and the PSNR value of cover image with respect to Stego image are also shown in table 3.1. It is seen that the maximum value of *PSNR* is 37.5542 and minimum value of *PSNR* is 29.4570. If x and y is cover and stego image then *PSNR* is calculated as:

$$PSNR = 10 \times \log_{10} \frac{(2^z-1)^2}{MSE} \quad \ldots(3.3)$$

Where z is the bit depth of the image and MSE is defined as,

$$MSE = \sum_{m=1}^{M} \sum_{n=1}^{N} \frac{(A_{mn}-B_{mn})^2}{M \times N} \quad \ldots(3.4)$$

(a)



(b)

(c)

(d)

(e)

(f)

Figure 3.4 (a) Original Boat Image (b) Original Lake Image (c) Stego Boat Image after hiding Lake Image (d) Original Zelda Image (e) Original Cameraman Image (f) Stego Zelda Image after hiding Lake Image.

3.4 Conclusion

In this work, image steganography algorithm based on pixel value differences is proposed. The proposed algorithm is implemented in MATLAB. Fifteen different cover images like Lena, Barbara, Peppers, and Cameraman *etc.* are considered in this work. Some of these cover images are shown in Figure 3.4(a), Figure 3.4(b) and Figure 3.4(c). The maximum capacity of hidden bits is 32,883 and maximum *PSNR* is 37.5542 dB.

CHAPTER 4 STEGANALYSIS

4.1 STEGANALYSIS

Steganalysis developed till date can be classified into visual and statistical steganalysis. The statistical Steganalysis can further be classified as:

(i) Histogram steganalysis

(ii) Chi-square steganalysis

Each of these classes of analysis is covered in detail in the next two subsections along with several examples of each category.

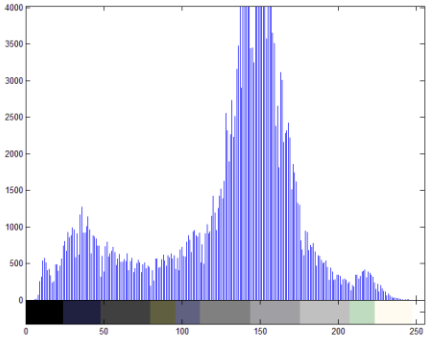
These Steganalysis are designed keeping a particular steganographic algorithm in mind. These Steganalysis are based on the image features which get modified by a particular kind of steganographic embedding. A particular steganographic algorithm imposes a specific kind of behaviour on the image features. This specific kind of behaviour of the image statistics is exploited by the histogram steganalysis.

4.1.1 Histogram Steganalysis

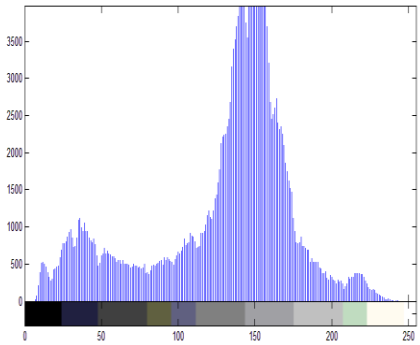
One of the proposed methods for detecting steganography is histogram analysis. Histogram is used to visualize the changes made to the image histogram due to embedding. Image histogram is a graphical representation of the distribution of colors or gray scales in an image. It has been applied to detect embedding by methods based on least-significant bit (e.g. *LSB* replacement and *LSB* matching) [18], [19], [20]. Although in general visual artifacts are not noticeable by human eyes in the stego-image, changes in the histogram can be easily observed. 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 low 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.

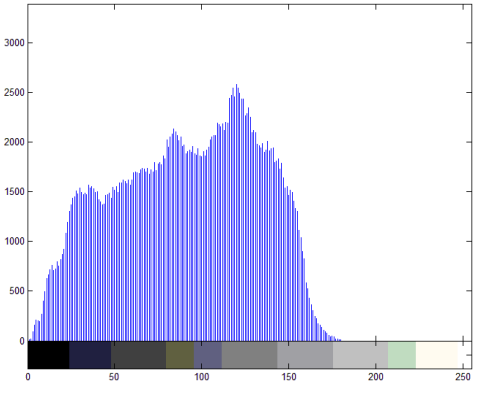
The histogram analysis method exploits the asymmetry introduced by *LSB* replacement. The main idea is to look for statistical artifacts of embedding in the histogram of a given image. Figure 4.1(a) represents histogram of original boat image and Figure 4.2(b) represents histogram of stego boat image which is quiet similar to original boat image histogram. So it is difficult to detect stego part i.e. either image is stego image or original image by checking histogram of cover and stego image. Similarly, Figure 4.2(a) represents histogram of original zelda image and Figure 4.2(b) represents histogram of stego zelda image which is quiet similar to original image histogram. Histogram analysis shows how difficult to detect hidden data if histogram of cover and stego image is same.



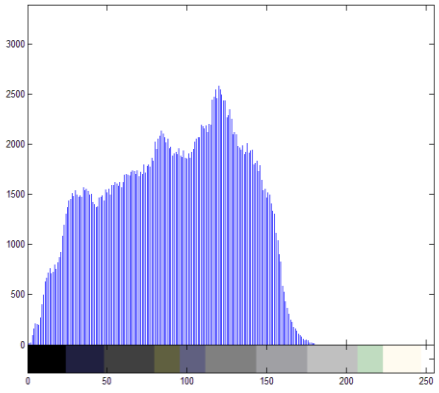
(a)



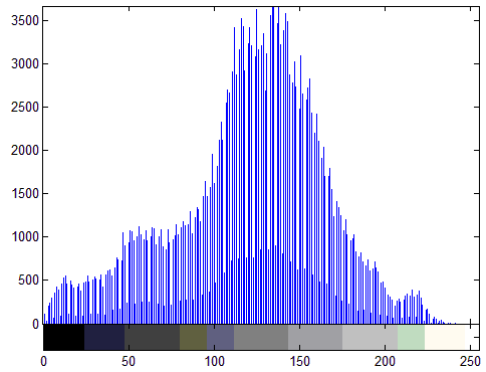
(b)



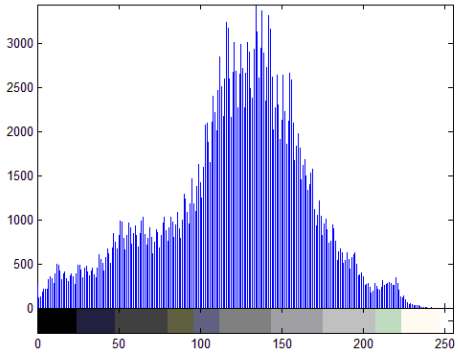
(c)



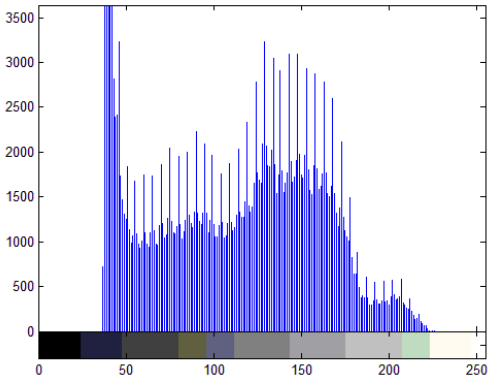
(d)


3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

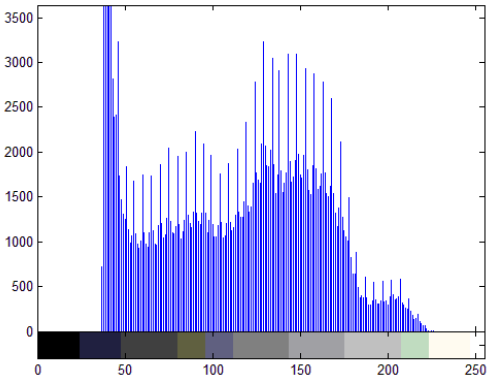
(e)


3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

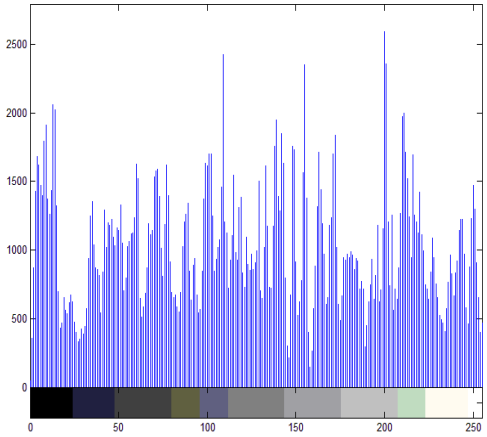
(f)


3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

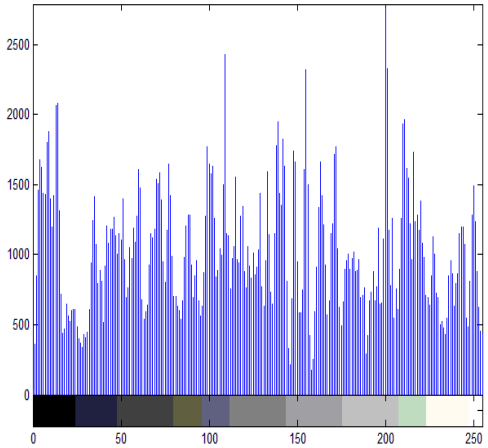
(g)


3500
3000
2500
2000
1500
1000
500
0
0
50
100
150
200
250

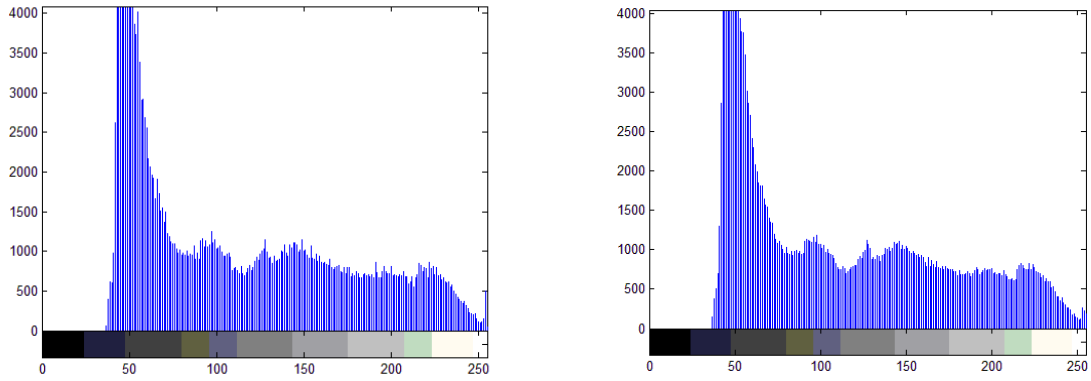
(h)


2500
2000
1500
1000
500
0
0
50
100
150
200
250

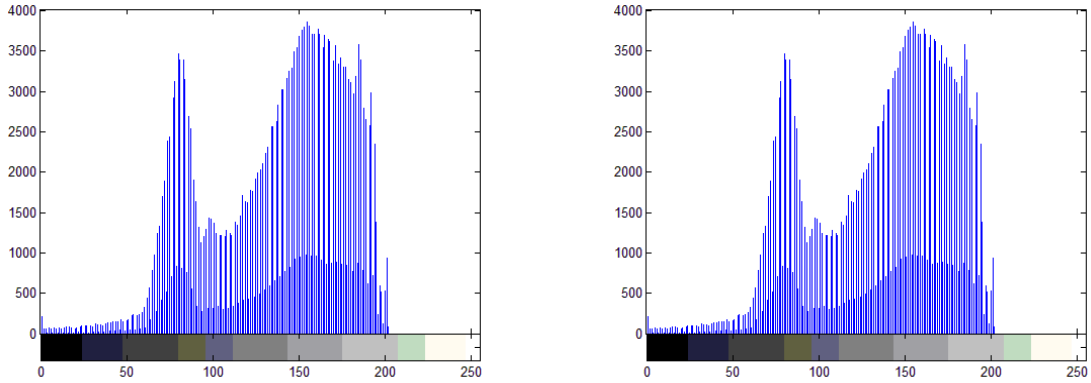
(i)


2500
2000
1500
1000
500
0
0
50
100
150
200
250

(j)



(k) (l)



(m) (n)

Figure 4.1 (a)Original Boat Image histogram (b)Stego Boat Image histogram (c)Original Zelda Image histogram (d)Stego Zelda Image histogram (e)Original Livingroom Image histogram (f)Stego Livingroom Image histogram (g)Original Pirate Image histogram (h)Stego Pirate Image histogram (i)Original Peppers Image histogram (j)Stego Peppers Image histogram (k)Original Woman_darkhair Image histogram (l)Stego Woman_darkhair Image histogram (m) Original Woman_blonde Image histogram.

4.1.2 Chi-Square Steganalysis

The chi-square ($\chi 2$) test is another approach that can be used to determine whether the statistical properties of an image are changed due to altering the least significant bits (*LSBs*) of the image pixels. Unlike the message to be hidden, the *LSBs* of the image

pixels are not random; the backgrounds of the majority of images contain comparable *LSBs*. The embedding data will affect the histogram of grayscale frequencies in a particular way.

So, based on this artifact a statistical attack based on the Chi-Square Hypothesis Testing is developed to probabilistically suggest one of the following two hypotheses:

Null Hypothesis H0: The given image contains steganographic embedding

Alternative HypothesisH1: The given image does not contain steganographic embedding

The decision to accept or reject the Null Hypothesis H0 is made on basis of the observed confidence value p.

Table 4.1 Chi-square comparison between cover and stego image

Cover Image	**Chi-square ($\chi 2$) Steganalysis in case of cover image(in 10^6)**	**Chi-square($\chi 2$) Steganalysis in case of stego image(in 10^6)**	**Difference(in 10^6)**
Pirate	4.297000	4.254700	0.004230
Peppers	8.784500	8.733700	0.050800
Zelda	3.811800	3.797300	0.014500
Cameraman	6.593900	6.569900	0.024000
Living room	3.281300	3.225600	0.055700
Boat	3.152100	3.098300	0.053800
Sailboat	3.281900	3.224200	0.057700
Lake	4.666800	4.654200	0.012600

Woman darkhair	5.384000	5.378200	0.005800
Jet plane	6.893400	6.885800	0.007600
Walkbridge	8.568700	8.493400	0.075300

Table 4.1 shows the chi-square difference between the cover image and stego image. Minimum the difference between cover and stego image, more secure the image. It means less difference of chi square values not easily detect by victim i.e. more compatibility of hidden image with cover image.

CHAPTER 5 Conclusion and Future Work

In this thesis, a steganography algorithm using pixel value differences of neighbouring pixels for images is proposed. The maximum *PSNR* of proposed algorithm is 37.554 *d*B and maximum hidden capacity is 32,883 bits. Steganalysis of the proposed algorithm is performed. Two steganalysis tests- Histogram steganalysis and Chi-square steganalysis are performed on stego images to show the effectiveness of the proposed algorithm.

The following future direction can be

- It can be extended for video files and medical images *etc*.
- More neighbouring pixels can be considered to increase the capacity and quality of the stego images.

References

[1] Alattar A. M., "Reversible Watermark using the Difference Expansion of a Generalized Integer Transform", *IEEE* Transactions on Image Processing, 13 (8) 1147-1156, 2004.

[2] Chang, C. C. and Tseng, H. W., "A steganographic method for digital images using side match", Pattern Recognition Letters, 1431-1437, 25, 2004.

[3] Chen C., Shi Y.Q., Chen W., and Xuan G., "Statistical moments based universal steganalysis using JPEG-2D array and 2-D characteristic function", in Proc. Int. Conf. on Image Processing, Atlanta, GA, USA, pp. 105-108, 8-11 Oct., 2006.

[4] Crandall R., "Some Notes on Steganography", Posted on Steganography Mailing List, 1998.

[5] Fridrich J, Goljan M., and Du R., "Detecting LSB steganography in color, and gray-scale images," IEEE multimedia, Vol. 8, Issue 4, pp. 22-28, 2001.

[6] Fridrich J., "Feature-Based Steganalysis for JPEG Images and its Implications for Future Design of Steganographic Schemes", in Proc. 6th Int. Workshop on Information Hiding, Toronto, Canada, pp. 67-81, 23-25 May 2004.

[7] Goel P., Sur A., and Mukhopadhyay J., "A SDS based Steganographic scheme for reducing Embedding Noise", 15th International Conference on Advanced Computing and Communication, Guwahati, India, pp. 771-775, 18-21 Dec., 2007.

[8] Goel P., Sur A., and Mukhopadhyay J., "A Spatial Domain Steganographic Scheme for Reducing Embedding Noise", in Proc. 3rd International Symposium on Communications, Control and Signal Processing, St. Julians, Malta, 12, pp. 1024 – 1028, March, 2008.

[9] Johnson N. F., and Jajodia S., "Steganography: Seeing the Unseen", IEEE Computer, pp. 26-34, Feb. 1998.

[10] Changa Ko-Chin, Changa Chien-Ping, Huangb Ping S., and Tua Te-Ming, "A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing" Journal of Multimedia, Vol. 3, No. 2, June 2008

[11] Li B., He J., Huang J., and Shi Y. Q., "A survey on image steganography and steganalysis," Journal of Information Hiding and Multimedia Signal Processing, Vol. 2, No. 2, pp. 142–172, 2011.

[12] Li S. L., Leung K. C., Cheng L. M., and Chan C. K., "Data Hiding in Images by Adaptive LSB Substitution Based on the Pixel-Value Differencing," First International Conference on Innovative Computing, Information and Control, Vol. 3, pp. 58-61, 2006.

[13] Mandal J. K., Khamrui A., "A Data- Hiding Scheme for Digital Image using Pixel Value Differencing", in International Symposium on Electronic System Design, 2011.

[14] Manjunath B. S., Sarkar A., and Solanki K., "Further Study on YASS: SteganographyBased on Randomized Embedding to Resist Blind Steganalysis", in Proc. SPIE -Security, Steganography, and Watermarking of Multimedia Contents X, San Jose, California, Vol. 6819, pp. 681917-681917-11, Jan. 2008.

[15] Provos N., "Defending against statistical steganalysis", in Proc. 10th USENIX Security Symposium, Vol. 10, pp. 24-24, Washington DC, 2001.

[16] Suk-Ling Li, Lai-Chi Leung, L.M. Cheng, Chi-Kwong Chan, "Performance Evaluation of a Steganographic Method for Digital Images Using Side Match" , IS16-004, Aug 2006.

[17] Tsai W. H., Wu D. C., A steganographic method for images by pixel-value differencing, Pattern Recognition Letters, pp. 1613–1626, 2003.

[18] Wang C. M., Wu N. I., Tsai C. S., and Hwang M. S., "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.

[19] Westfeld A., "High capacity despite better steganalysis (F5 - a steganographic algorithm)", in Proc. 4th Int. Workshop on Information Hiding, Pittsburgh, PA, USA, pp. 289-302, 25-27 April 2001.

[20] Wu H. C, Wu N. I, Tsai C. S, Hwang M. S. "Image steganographic scheme based on pixel-value differencing and LSB replacement methods" IEE proceedings in Visual Image and Signal Processing, Vol. 152, No. 5, pp. 611-615, Oct 2005.

[21] Xin Liao, Qiao-yan Wen, Jie Zhang, "A steganographic method for digital images with four-pixel differencing and modified LSB substitution", Journal of Visual Communication and Image Representation, pp. 1-8, 22 August 2010.

[22] Yang C. H., Wang S. J., and Weng C. Y., "Analyses of pixel-value-differencing schemes with LSB replacement in steganography," in Proc. Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing, IIHMSP, 2007.

[23] Yang C. and Weng C. Y., "A steganographic method for digital images by multi-pixel differencing," in Proc. of International Computer Symposium, Taipei, Taiwan, ROC, 2006.

[24] Zhang Han-ling, Geng Guang-zhi, Xiong Cai-qiong, "Image Steganography using Pixel-Value Differencing", Second International Symposium on Electronic Commerce and Security, pp.1-4, 2009