**Hiding Data in Images Using Mode 16 Method (M16M)**

***Abstract****—Steganography is a process that involves hiding a message in an appropriate carrier like image or audio. The carrier can be sent to a receiver without anyone except the authenticated receiver only knows existence of the information. Considerable amount of work has been carried out by different researchers on steganography. In this work the authors propose a novel Steganographic method for hiding information within the spatial domain of the gray scale image. The proposed approach works by selecting the embedding pixels using some mathematical function and then finds the 8 neighborhood of the each selected pixel and map each two bit of the secret message in each of the neighbor pixel according to its intensity value in a specified*

*manner. Before embedding a checking has been done to find out whether the selected pixel or its neighbor lies at the boundary of the image or not. This solution is independent of the nature of the data to be hidden and produces a stego image with minimum degradation.*

Keywords: Cover Image, Mode 16 Method (M16M), Stego Image

**1. Introduction**

STEGANOGRAPHY is the art and science of hiding information by embedding messages within other, seemingly harmless messages. Steganography means “covered writing” in Greek. As the goal of steganography is to hide the presence of a message and to create a covert channel, it can be seen as the complement of cryptography, whose goal is to hide the content of a message. Another form of information hiding is digital watermarking, which is the process that embeds data called a watermark, tag or label into a multimedia object such that watermark can be detected or extracted later to make an assertion about the object. The object may be an image, audio, video or text only. A famous illustration of steganography is Simmons’ Prisoners’ Problem [15].An assumption can be made based on this model is that if both the sender and receiver share some common secret information then the corresponding steganography protocol is known as then the

secret key steganography where as pure steganography means that there is none prior information shared by sender and receiver. If the public key of the receiver is known to the sender, the steganographic protocol is called public key steganography [2], [3] and [9].For a more thorough knowledge of steganography methodology the reader may see [13], [16].Some steganographicmodel with high security features has been presented in [4], [5] and [6].Almost all digital file formats can be used for steganography, but the image and audio files are more suitable because of their high degree of redundancy [16]. Fig. 1 below shows the different categories of steganography techniques.



Fig. 1: Types of Steganography

A block diagram of a generic image steganographic system is given in Fig. 2. A message is embedded in a digital image (cover image) through an embedding algorithm, with the help of a secret key. The resulting stego image is transmitted over a channel to the receiver where it is processed by the extraction algorithm using the same key. During transmission the stego image, it can be monitored by unauthenticated viewers who will only notice the transmission of an image without discovering the existence of the hidden message. In this work a specific image based steganographic method for gray level image has proposed. In this method instead of embedding the secret message into the cover image a mapping technique has been incorporated to generate the stego image. This method is capable of extracting the secret message without the presence of the cover image.

****

Fig. 2: Generic form of Image Steganography

the secret message into the cover image a mapping technique has been incorporated to generate the stego image. This method is capable of extracting the secret message without the presence of the cover image. This paper has been organized as following sections: Section II describes some related works, Section III deals

with proposed method. Algorithms are discussed in Section IV and Experimental results are shown in Section V. Section VI contains the analysis of the results and Section VII draws the conclusion.

**2. RelatedWorks**

**2.1 Data Hiding by LSB**

Various techniques about data hiding have been proposed in literatures. One of the common techniques is based on manipulating the least-significant-bit (LSB) [7], [8] and [12], [14]planes by directly replacing the LSBs of the cover-image

with the message bits. LSB methods typically achieve high capacity but unfortunately LSB insertion is vulnerable to slight image manipulation such as cropping and compression.

2.2 Data Hiding by PVD

The pixel-value differencing (PVD) method proposed by Wu and Tsai [17] can successfully provide both high embedding capacity and outstanding imperceptibility for the stegoimage.

The pixel-value differencing (PVD) method segments the cover image into non overlapping blocks containing two connecting pixels and modifies the pixel difference in each

block (pair) for data embedding. A larger difference in the original pixel values allows a greater modification. In the extraction phase, the original range table is necessary. It is used to partition the stego-image by the same method as

used to the cover image. Based on PVD method, various approaches have also been proposed. Among them Chang et al. [11]. proposes a new method using tri-way pixel-value

differencing which is better than original PVD method with respect to the embedding capacity and PSNR.

**2.3 Data Hiding by GLM**

In 2004, Potdar et al.[10] proposes GLM (Gray level

modification) technique which is used to map data by modifying the gray level of the image pixels. Gray level modification Steganography is a technique to map data (not embed or hide it) by modifying the gray level values of the image pixels. GLM technique uses the concept of odd and even numbers to map data within an image. It is a one-toone mapping between the binary data and the selected pixels in an image. From a given image a set of pixels are selected based on a mathematical function. The gray level values of those pixels are examined and compared with the bit stream

that is to be mapped in the image. Fig. 3: Data Embedding Process in GLM Fig. 4: Data Extraction Process in GLM

**2.4 Data Hiding by the method proposed by Ahmad et all.**

In this work [1] a novel Steganographic method for

hiding information within the spatial domain of the grayscale image has been proposed. The proposed approach works by dividing the cover into blocks of equal sizes and then embeds the message in the edge of the block depending on the number of ones in left four bits of the pixel.

**3. Proposed Method**

In this section the authors propose a new method for information hiding within the spatial domain of any gray scale image. The input messages can be in any digital form, and are often treated as a bit stream. Embedding pixels are selected based on some mathematical function which depends on the pixel intensity value of the seed pixel and

its 8 neighbors are selected in counter clockwise direction. Before embedding a checking has been done to find out whether the randomly selected pixel or its neighbor lies at the boundary of the image or not. Data embedding is done by mapping each four bit of the secret message in each of the

neighbour pixel based on the remainder of the intensity value when divided by 16. In Fig.5 Embedding Technique has been shown. Extraction process starts by selecting the same pixels required during embedding. At the receiver

side other different reverse operation has been carried out to get back the original information.

**3.1 Embedding Technique**

We have found out an embedding technique that is difficult to interpret yet very easy to implement.

The technique goes like this:-

1. Select a seed pixel and get the neighbouring pixels.

2. For each neighbouring pixel, take the intensity and get the remainder after modulus division it by 16.**Let the remainder be R**.

3. Subtract **R** from the intensity and hence the resultant intensity value is divisible by 16.

4. Add different values to the resultant intensity according to the different 4 bit binary message to be embedded according to the table:-

**MSG VALUE TO BE ADDED** 0000 0

0001 1

0010 2

0011 3

0100 4

0101 5

0110 6

0111 7

1000 8

1001 9

1010 10

1011 11

1100 12

1101 13

1110 14

1111 15

6.The modified intensity is replaced in the original image matrix.

7.At the end, get the stego matrix embedded with the message is obtained.

**4. Algorithms**

**4.1 Data Embedding Method**

Let C be the original 8 bit gray scale image of size N x N i.e. C = (Pij | 0 ≤ i < N, 0 ≤ j < N, Pij ∈ 0, 1, . . . , 255). Let MSG be the n bit secret message represented as MSG =(mk | 0 ≤ k < n,mk ∈ 0, 1).A seed pixel Prc can be selected with row (r) and column (c). Next step is to find the 8 neighbors Pr′c′ of the pixel Prc such that r′ = r + l , c′ = c + l ,−1 ≤ l ≤ 1. The embedding process will be finished when all the bits of every byte of secret message are mapped or embedded. Algorithm of the embedding method are described as :

**EMBEDDING ALGORITHM(4 Bit):-**

* Input: Cover Image Matrix(C),Message
* MSG=Message converted to binary from decimal
* Msize=size of binary message
* Imsize=Size of the image matrix
* Determine all the Seed Pixels required for the message before embedding starts
* We start with the first seed pixel on the image
* Count=1
* msg0=0
* msg1=1
* While(Count<=Msize)
* **Begin While Loop**
* For loop for embedding message around the seed pixel
* Get the First/Next surrounding pixel location about the seed pixel.
* V is the intensity of the neighbouring pixel.
* R is the remainder after dividing V by 16.
* Subtract R from V to get V’.V’ is divisible by 16.
* cvr=V’
* msgx1=Get next message bit
* Count=Count+1
* msgx2= Get next message bit
* Count=Count+1
* msgx3= Get next message bit
* Count=Count+1
* msgx4= Get next message bit
* Count=Count+1
* If(msgx1==0&&msgx2==msg0&&msgx3==msg0&&msgx4==msg0)
* cvr=cvr;
* elseif(msgx1==msg0&&msgx2==msg0&&msgx3==msg0&&msgx4==msg1)
* cvr=cvr+1;
* elseif(msgx1==msg0&&msgx2==msg0&&msgx3==msg1&&msgx4==msg0)
* cvr=cvr+2;
* elseif(msgx1==msg0&&msgx2==msg0&&msgx3==msg1&&msgx4==msg1)
* cvr=cvr+3;
* elseif(msgx1==msg0&&msgx2==msg1&&msgx3==msg0&&msgx4==msg0)
* cvr=cvr+4;
* elseif(msgx1==msg0&&msgx2==msg1&&msgx3==msg0&&msgx4==msg1)
* cvr=cvr+5;
* elseif(msgx1==msg0&&msgx2==msg1&&msgx3==msg1&&msgx4==msg0)
* cvr=cvr+6;
* elseif(msgx1==msg0&&msgx2==msg1&&msgx3==msg1&&msgx4==msg1)
* cvr=cvr+7;
* elseif(msgx1==msg1&&msgx2==msg0&&msgx3==msg0&&msgx4==msg0)
* cvr=cvr+8;
* elseif(msgx1==msg1&&msgx2==msg0&&msgx3==msg0&&msgx4==msg1)
* cvr=cvr+9;
* elseif(msgx1==msg1&&msgx2==msg0&&msgx3==msg1&&msgx4==msg0)
* cvr=cvr+10;
* elseif(msgx1==msg1&&msgx2==msg0&&msgx3==msg1&&msgx4==msg1)
* cvr=cvr+11;
* elseif(msgx1==msg1&&msgx2==msg1&&msgx3==msg0&&msgx4==msg0)
* cvr=cvr+12;
* elseif(msgx1==msg1&&msgx2==msg1&&msgx3==msg0&&msgx4==msg1)
* cvr=cvr+13;
* elseif(msgx1==msg1&&msgx2==msg1&&msgx3==msg1&&msgx4==msg0)
* cvr=cvr+14;
* elseif(msgx1==msg1&&msgx2==msg1&&msgx3==msg1&&msgx4==msg1)
* cvr=cvr+15;
* **End For loop**
* Get the next pixel value from the list of seed pixels
* **End While Loop**
* Return Stego image matrix(S).

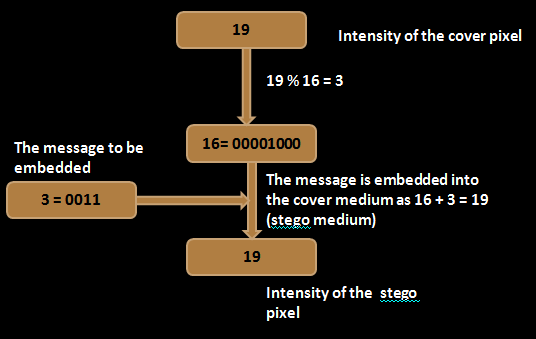


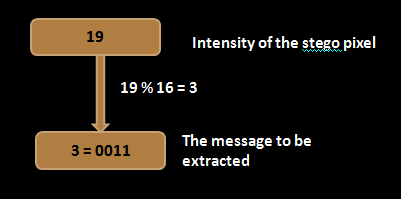
Fig. 6: A snapshot of data embedding process

**4.2 Data Extraction Method**

The process of extraction proceeds by selecting those same pixel with their neighbors. The extracting process will be finished when all the bits of every bytes of secret message are extracted. Algorithm of the extraction method are described as :

**Receiving Algorithm(4 Bit):**

* Input: Stego Image Matrix(S),Length of the embedded message(Msize)
* Msize=Msize/4
* Imsize=Size of the image Matrix
* Binmsg=’ ‘ //Initializing
* Determine all the Seed Pixels required for the message before decoding starts.
* We start with the first seed pixel on the image.
* count=0
* While(count<=Msize)
* **Begin While Loop**
* For loop to extract message around a seed pixel
* Get the First/Next surrounding pixel location about a fixed seed pixel
* V is the intensity of the neighbouring pixel
* R is the remainder after dividing V by 16.
* If(R==0)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==1)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==2)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==3)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==4)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0')
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==5)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==6)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==7)
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==8)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==9)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==10)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0')
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==11)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==12)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==13)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* elseif(R==14)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('0');
* count=count+1;
* elseif(R==15)
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* binmsg1(count)=char('1');
* count=count+1;
* **End for loop**
* Take the next seed pixel value
* **End While loop**
* Group the binmsg into groups of eight bits and find the corresponding message.



**Fig. 7: A snapshot of data extracting process**

One important point needs to be kept in mind that a specific order for selecting the neighbour pixels has to be maintained for embedding / mapping process and also for the process of extraction otherwise it would not be possible for retrieve the data in proper sequence. This sequence has been shown in Figure 8.



**4.3 Pixel Selection Method**

Random Pixel Generation for embedding message bits is dependent on the intensity value of the previous pixel selected.It includes a decision factor (dp) which is dependent on intensity with a fixed way of calculating the next pixel.The algorithm for selection of pixel for embedding is described below:

* **Input: C , previous pixel position (x,y),pixel intensity value (v).**
* **Consider dp (Decision Parameter)=1 if (intensity ≤80),**

**dp=2 if (intensity ≥ 80 & ≤ 160) ,**

**dp=3 if (intensity > 160 & ≤ 255).**

* **t = x + 2 + dp**
* **if (t ≥ N)m = 2, n = y + 2 + dp**
* **else m = x + 2 + dp, n = y**
* **Return m and n.**
* **End.**

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **122** | **45** | **69** | **132** | **256** | **145** | **56** | **79** | **112** |
| **156** | **125** | **169** | **123** | **79** | **78** | **12** | **186** | **123** |
| **224** | **212** | **145** | **125** | **147** | **86** | **45** | **110** | **123** |
| **119** | **248** | **46** | **112** | **48** | **23** | **79** | **45** | **90** |
| **119** | **79** | **116** | **189** | **53** | **63** | **130** | **90** | **141** |
| **56** | **71** | **26** | **83** | **43** | **75** | **93** | **67** | **116** |
| **90** | **112** | **179** | **212** | **201** | **38** | **99** | **119** | **157** |
| **83** | **53** | **89** | **115** | **63** | **78** | **90** | **76** | **255** |
| **131** | **141** | **176** | **159** | **126** | **146** | **255** | **73** | **86** |

Fig. 9: Snapshot of Selected Pixel for embedding.

**5 .Experimental Results**

In this section the authors present the experimental result of the proposed method based on two benchmarks techniques to evaluate the hiding performance. First one is the capacity of hiding data and another one is the imperceptibility of the stego image, also called the quality of stego image. The quality of stego-image should be acceptable by human eyes. The authors also present a comparative study of the proposed methods with the existing methods like PVD,GLM and the methods proposed by Ahmad T et al.by computing embedding capacity, mean square error (MSE) and peak signal-to noise ratio (PSNR).The authors also compute the normalized cross correlation coefficient for computing the similarity measure between the cover image and stego image. In this section experimental result of stego image are shown based on two well known images: Lena and Pepper. In Fig. 11 a segment of Lena as cover image has been shown. Fig 12 shows the same segment of Lena as stego image after embedding the message **"I am an Indian"** on that segment.

A comparison of the embedding capacity has been illustrated in figure 10

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| IMAGE | Image  Size | PVD | GLM | AHMAD  ET ALL | M16M |
| LENA | 128X128 | \*\* | 2048 | 2493 | 4736 |
| 256X256 | \*\* | 8192 | 10007 | 19684 |
| 512X512 | 50960 | 32768 | 40017 | 89920 |
| PEPPER | 128X128 | \*\* | 2048 | 2443 | 5676 |
| 256X256 | \*\* | 8192 | 9767 | 22956 |
| 512X512 | 50685 | 32768 | 39034 | 93660 |

Fig. 10: Comparision of embedding capacity

\*\* For PVD method all the images used are of size

512x512

**5.1 Peak Signal to Noise Ratio (PSNR)**

PSNR measures the quality of the image by comparing the original image or cover image with the stego-image, i.e. it measures the percentage of the stego data to the image percentage. The PSNR is used to evaluate the quality of the stego-image after embedding the secret message in the cover.

Assume a cover image C(i,j) that contains N by N pixels and a stego image S(i,j) where S is generated by embedding / mapping the message bit stream. Mean squared error (MSE) of the stego image as follows:



The PSNR is computed using the following formulae:

**PSNR = 10 log10 2552/ MSE db.**

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **3** | **12** | **122** | **34** | **123** | **38** | **64** | **57** | **56** | **89** |
| **12** | **23** | **74** | **34** | **53** | **75** | **49** | **54** | **67** | **54** |
| **75** | **87** | **91** | **94** | **97** | **97** | **94** | **95** | **97** | **97** |
| **96** | **86** | **93** | **98** | **97** | **96** | **97** | **96** | **93** | **95** |
| **18** | **18** | **23** | **18** | **18** | **15** | **16** | **18** | **13** | **17** |
| **69** | **70** | **54** | **45** | **29** | **17** | **20** | **19** | **14** | **12** |
| **90** | **88** | **87** | **88** | **85** | **88** | **88** | **86** | **85** | **87** |
| **78** | **78** | **79** | **82** | **78** | **74** | **72** | **61** | **64** | **66** |
| **83** | **84** | **89** | **81** | **81** | **78** | **67** | **55** | **38** | **27** |
| **91** | **91** | **95** | **90** | **87** | **87** | **90** | **89** | **90** | **93** |

Fig. 11: A Segment of Cover Image with selected pixel

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 12 | 6 | 127 | 33 | 112 | 32 | 64 | 57 | 56 | 89 |
| 6 | 23 | 68 | 38 | 53 | 70 | 49 | 54 | 67 | 54 |
| 64 | 82 | 89 | 89 | 102 | 100 | 94 | 95 | 97 | 97 |
| 96 | 86 | 93 | 98 | 97 | 96 | 97 | 96 | 93 | 95 |
| 18 | 18 | 23 | 18 | 18 | 15 | 16 | 18 | 13 | 17 |
| 69 | 70 | 54 | 45 | 29 | 17 | 20 | 19 | 14 | 12 |
| 89 | 86 | 94 | 88 | 85 | 88 | 88 | 86 | 85 | 87 |
| 68 | 78 | 71 | 82 | 78 | 74 | 72 | 61 | 64 | 66 |
| 80 | 82 | 89 | 81 | 81 | 78 | 67 | 55 | 38 | 27 |
| 91 | 91 | 95 | 90 | 87 | 87 | 90 | 89 | 90 | 93 |

Fig. 12: A Segment of Stego Image with selected pixel

**5.2 Similarity Measure**

For comparing the similarity between cover image and the stego image, the normalized cross correlation coefficient (r) has been computed. In statistics, correlation indicates the strength and direction of a linear relationship between two

random variables. The correlation coefficient ***ƥxy*** between two random variables X and Y with expected values **μx** and **μy** and standard deviations **σx** and **σy** is defined as :



where E is the expected value operator and cov means covariance. The value of correlation is 1 in the case of an increasing linear relationship, -1 in the case of a decreasing



Fig. 13: A) Cover Image B) Stego Image of Lena after embedding "I am an Indian and I feel proud to an Indian."



Fig. 14: A) Cover Image B) Stego Image of Pepper after embedding "I am an Indian and I feel proud to an Indian.

linear relationship, and some value in between in all other cases, indicating the degree of linear dependence between the variables. Cross correlation is a standard method of estimating

the degree to which two series are correlated. Consider two series x(i) and y(i) where i=0,1,2,. . . ,N-1. The cross correlation r at delay d is defined as



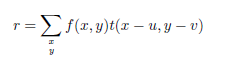
where mx and my are the means of the corresponding series. The cross-correlation is used for template matching which is motivated through the following formula

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Images** |  | **Data Size** | | | | | |
| **100** | **500** | **1000** | **2000** | **4000** | **5000** |
| **Lena**  **512X512** | **PSNR** | **63.3** | **57.5** | **54.5** | **51.3** | **47.8** | **46.8** |
| **MSE** | **0.03** | **0.11** | **0.22** | **0.47** | **1.07** | **1.34** |
| **Correlation** | **1.00** | **1.00** | **0.99** | **0.99** | **0.99** | **0.99** |
| **Lena**  **256X256** | **PSNR** | **57.0** | **50.5** | **47.3** | **44.3** | **41.5** | **40.5** |
| **MSE** | **0.12** | **0.56** | **1.20** | **2.33** | **4.52** | **5.71** |
| **Correlation** | **1.00** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** |
| **Lena**  **128X128** | **PSNR** | **51.3** | **44.5** | **41.4** | **38.4** | **35.4** | **N.A.** |
| **MSE** | **0.47** | **2.29** | **4.67** | **9.21** | **18.4** | **N.A** |
| **Correlation** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** | **N.A** |
| **Pepper**  **512X512** | **PSNR** | **62.6** | **56.2** | **53.2** | **50.1** | **47.1** | **46.3** |
| **MSE** | **0.03** | **0.15** | **0.30** | **0.63** | **1.25** | **1.52** |
| **Correlation** | **1.00** | **1.00** | **0.99** | **0.99** | **0.99** | **0.99** |
| **Pepper**  **256X256** | **PSNR** | **57.4** | **50.5** | **47.2** | **44.3** | **41.4** | **40.3** |
| **MSE** | **0.11** | **0.64** | **1.22** | **2.38** | **4.68** | **5.94** |
| **Correlation** | **1.00** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** |
| **Pepper**  **128X128** | **PSNR** | **50.9** | **44.3** | **41.5** | **38.4** | **35.6** | **34.6** |
| **MSE** | **0.52** | **2.40** | **4.56** | **9.27** | **17.7** | **22.4** |
| **Correlation** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** | **0.99** |

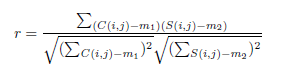
**Fig .15:A)Comparison of the Experimental Results**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Image | Image Size | PVD | GLM | AHMAD ET ALL | **M16M** |
| Lena | 128X128 | 36.20 | 30.50 | 44.30 | **34.7956** |
| 256X256 | 35.00 | 35.20 | 46.80 | **34.6156** |
| 512X512 | 41.79 | 35.50 | 55.00 | **34.0496** |
| Pepper | 128X128 | 38.70 | 38.00 | 43.60 | **34.0796** |
| 256X26 | 35.00 | 37.20 | 47.50 | **33.9842** |
| 512X512 | 40.97 | 34.00 | 57.50 | **33.9044** |

**Fig. 15: B) Comparison of PSNR**



where f is the image and the sum is over x, y under the window containing the feature t positioned at u, v. Similarity measure of two images can be done with the help of normalized cross correlation generated from the above concept using the following formula:



Here C is the cover image, S is the stego image,m1 is the mean pixel value of the cover image and m2 is the mean pixel value of stego image.

**6. Analysis of the Results**

In this article the authors proposed an efficient data hiding technique using image based steganography approach in a gray scale image. Comparison has been shown with some existing methods like PVD, GLM and the technique proposed by Ahmad T et al. From the experimental results in can be seen that the embedding capacity of the proposed method is better in most cases compared to the other method except the PVD technique and also the similarity measures proves that the proposed method is best among these four methods which ensures that cover image and the stego image is almost identical. Also as the message bits are not directly embedded at the pixels of the cover image, steganalysis may be able to find out the embedded bits but can not be able

to extract the original message bits. Besides PSNR value of the proposed method for various size of the image is much better than compared to other methods.

**7. Conclusion**

The work dealt with the techniques for steganography as related to gray scale image. A new and efficient steganographic method for embedding secret messages into images

without producing any major changes has been proposed. Although in this method it has been shown that each two bit of the secret message has been mapped in the pixels of the cover image,but this method can be extended to map n no of bits also by considering more no of features of the

embedding pixels.This method also capable of extracting the secret message without the cover image. This approach may be modified to work on color images also

**8. References**

[1] Ahmad T. Al-Taani. and Abdullah M. AL-Issa. A novel steganographic method for gray-level images. *International Journal of Computer, Information, and Systems Science, and Engineering*, 3, 2009.

[2] RJ Anderson. Stretching the limits of steganography. *Information Hiding, Springer Lecture Notes in Computer Science*, 1174:39–48, 1996.

[3] Ross J. Anderson. and Fabien A.P.Petitcolas. On the limits of

steganography. *IEEE Journal on Selected Areas in Communications (J-SAC), Special Issue on Copyright and Privacy Protection*, 16:474– 481, 1998.

[4] Souvik Bhattacharyya. and Gautam Sanyal. Study of secure steganography model. In *Proceedings of International Conference on AdvancedComputing*

*and Communication Technologies (ICACCT-2008)*,

Panipath,India, 2008.

[5] Souvik Bhattacharyya. and Gautam Sanyal. An image based steganography model for promoting global cyber security. In *Proceedings of International Conference on Systemics, Cybernetics and Informatics*, Hyderabad,India, 2009.

[6] Souvik Bhattacharyya. and Gautam Sanyal. Implementation and design of an image based steganographic model. In *Proceedings of IEEE International Advance Computing Conference*, Patiala ,India, 2009.

[7] J.Y. Hsiao. C.C. Chang. and C.-S. Chan. Finding optimal leastsignificant- bit substitution in image hiding by dynamic programming strategy. *Pattern Recognition*, 36:1583–1595, 2003.

[8] C.K. Chan. and L. M.Cheng. Hiding data in images by simple lsb substitution. *Pattern Recognition*, 37:469–474, 2004.

[9] Scott. Craver. On public-key steganography in the presence of an active warden. In *Proceedings of 2nd International Workshop on Information Hiding.*, pages 355–368, Portland,Oregon, USA, 1998.

[10] Potdar V.and Chang E. Gray level modification steganography for secret communication. In *IEEE International Conference on Industria lInformatics.*, pages 355–368, Berlin, Germany, 2004.

[11] P Huang. K.C. Chang., C.P Chang. and T.M Tu. A novel image steganography method using tri-way pixel value differencing. *Journal of Multimedia*, 3, 2008.

[12] Y. K. Lee. and L. H.Chen. High capacity image steganographic model. *IEE Proc.-Vision, Image and Signal Processing*, 147:288–294, 2000.

[13] N.F.Johnson. and S. Jajodia. Steganography: seeing the unseen. *IEEE Computer*, 16:26–34, 1998.

[14] C.F. Lin. R.Z. Wang. and J.C. Lin. Image hiding by optimal lsb

substitution and genetic algorithm. *Pattern Recognition*, 34:671–683, 2001.

[15] Gustavus J. Simmons. The prisoners’ problem and the subliminal channel. *Proceedings of CRYPTO.*, 83:51–67, 1984.

[16] JHP Eloff. T Mrkel. and MS Olivier. An overview of image

steganography. In *Proceedings of the fifth annual Information Security South Africa Conference.*, 2005.

[17] D.C. Wu. and W.H. Tsai. A steganographic method for images by pixel-value differencing. *Pattern Recognition Letters*, 24:1613–1626, 2003.