LJAST
Libya Journal Applied For Science and Technology
مجلة ليبيا للعلوم التطبيقية والتقنية

Volume 6. Issue 1
June 2019

LJAST
مجلة ليبيا للعلوم التطبيقية والتقنية

Analysis and Experimental Study of EMD and GEMD Steganographic Methods

OM-ESSAD M. LAMILES [1], HAJER A. ALASWED[1]
Dept. of Electronic Engineering
College of Industrial Technology(CIT)
Misurata, LIBYA
E-mail : Omu_alam2001@yahoo.com

Abstract

This paper proposed to analyze and experimentally study two steganographic methods: Exploiting Modification Direction (EMD) and Generalized Exploiting Modification Direction (GEMD). The implementation of these methods is explained in details such as the secret image used, data structures, and the optimal cover images number calculation. The main idea in EMD is that a separate n-pixel group of a cover image is used for embedding each next digit of (2n+1) -ary k-digit number and only one pixel in the n-pixel group could be modified by ±1. In GEMD, L-bit blocks from the input stream are embedded in the next n-pixel group, and at least one-pixel value in each group could be changed by ±1. In the implementation, some quality metrics like PSNR, MSE, and BPP are discussed. According to our analysis, PSNR of EMD is greater than that of GEMD. For MSE, EMD has less MSE than GEMD. On the other hand, GEMD is better than EMD in embedding capacity. GEMD is also better than EMD in memory and time consumption.

Keywords—Peak Signal to Noise Ratio (PSNR), Mean Square Error (MSE), Embedding Capacity, Bit Per Pixel (BPP), memory and time consumption, (2n+1)-ary number

I. Introduction

Steganography is a technique used to protect messages from unauthorized access, by embedding data into other media forms such as text, image, video, sound, etc., where the hidden data likely will not be detected [1].

There are two main directions in Steganography for hiding secret data, in the spatial domain and in frequency domain [1].

The last direction uses digital cosine transformations that is more time consuming compared with the spatial domain methods but provides more security. In Steganography, image file is the most common media form used, because the human visual system is not sensitive to small variation in colors. Furthermore, they could be easily used as cover media without any doubt as they are commonly used on the Internet [2].

We consider here spatial domain methods. There are many steganographic schemes based on direct replacement like Least Significant Bit (LSB) [3] [4] or based on indirect replacement such as Exploiting Modification Direction (EMD) [7] [13] [14], and Generalized Exploiting Modification

Direction (GEMD) [5] [6] [7] schemes; the latter ones will be discussed in this paper in details which similar to frequency domain methods provide greater security by the use of data transformations but in the spatial domain.

Spatial Domain Techniques

Steganographic algorithms are quite so many; each one has its own security and complexity, since the main aim for all of them is to embed large amount of secret data with less effects on the cover file, it means more embedding capacity Bit Per Pixel (BPP) with good image quality. One of the most common techniques is the LSB replacement method, where it is simple, fast, and has good stage image quality [2]. In this method the binary secret image is divided into blocks having L bits, and then embedding each L- block in L LSB's of each pixel of the cover image, where $1 \leq L \leq 8$. In general, this method can achieve a good image quality when $L \leq 3$, but for $4 \leq L \leq 8$, the image quality severely decreased [8].

To improve LSB replacement, many steganographic methods were proposed. In 2001 Wang, & Lin proposed a method that uses an optimal LSB replacement and genetic algorithm [12], where the genetic algorithm is presented to solve the problem of hiding data in the L LSBs of the cover image when L is large in order to improve the image quality and embedding capacity.

In 2002 Yu- Chee proposed a secure data hiding scheme for binary image [10], that uses a binary cover image to embed as many as log2(mn+1)bits of secret message into $m \times n$ block of binary cover image by changing at most two bits in the block , so this method has good image quality and embedding capacity.

In 2003 Wu & Tsai proposed a new method called Pixel Value Differencing (PVD) [13]. In this method, the cover image is divided into non-overlapping blocks of two adjacent pixels. A difference value is calculated from the values of the two pixels in each block. All possible difference values are classified into a number of ranges. The difference values then is replaced by a new value to embed the value of a sub-stream of the secret message. The number of bits which can be embedded in a pixel pair is decided by the width of the range that the difference value belongs to [13]. This method provided a better way to embed larger amount of secret data.

In 2005 Wu et al proposed a method based on LSB replacement and PVD methods [14]. First, a difference value from two adjacent pixels by PVD method is obtained, where 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. This method provided double embedding capacity of PVD method with a good stage image quality PSNR.

In 2006 Millikanian proposed a modification to LSB method that uses a pair of pixels from the cover image as a group [4], where the secret bits are carried in LSB's of two pixels. Therefore, this method has the same payload as LSB replacement method, but with fewer changes to the cover image pixels. So, the performance of this method is better than LSB replacement, and the direction of modification

to the cover pixels is exploited for data hiding, but there exist two different modification-directions corresponding to a same pair of secret bits to be embedded, meaning that the exploitation is incomplete [10].

Also, in 2006 Zhang and Wang proposed a new method called Exploiting Modification Direction (EMD) [15]. The main idea of the EMD method is to use a separate n-pixel group of a cover image to embed the next digit of (2n+1)-ray k-digit number representing the next L-bit block from the secret image input and only one pixel in the group can be changed by ±1. Therefore, this method has very good image quality and better embedding capacity, but embedding capacity decreases as increasing n.

To improve EMD method Lee et al. proposed Improved EMD (IEMD) method in 2007 [9]. This method uses two pixels from the cover image as group and 8-ary extraction function. It has greater embedding capacity than EMD, but it uses only two pixels in a group and cannot use more.

To enhance the hiding capacity of EMD and IEMD methods, a novel information concealing method based on Exploiting Modification Direction was proposed in 2011 [16]. This method embeds 2x secret digits in the 5-ary notational systems into each group of (2x + 1) cover pixels, where x is a positive integer. Thus, the proposed method can provide better hiding capacity.

In 2013 Kuo and Wang provided GEMD method [5], where it uses n-pixels from the cover image to embed n+1 bit, and at least one-pixel value in each group could be changed by ±1. Also, in this method there is no need for transformation, GEMD maintained good image quality and good embedding capacity, and also it can adjust the n-pixel size.

Frequency Domain Techniques

Frequency domain uses the transform coefficients to embed secret data. Moreover, frequency domain techniques are very robust against attacks. In frequency domain the cover image is transformed into the frequency domain coefficients before embedding secret messages in it, where the main techniques used are: Discrete Cosine Transform (DCT) [11], and Discrete Wavelet Transform (DWT), in Discrete Cosine Transform [17].

DCT method is used extensively with video and image compression e.g. JPEG compression, since for each color component the JPEG image format uses a discrete cosine transform to transform successive 8 × 8-pixel blocks of the image into 64 DCT coefficients each [11].

In DWT method [17], the cover image is divided into four sub-images such as approximation coefficients (CA), horizontal detail coefficients (CH), vertical detail coefficients (CV) and diagonal detail coefficients (CD). Similarly, the secret image is decomposed into four sub-images. These sub-images are divided into non-overlapping blocks. The blocks of approximation coefficients of cover image are subtracted from approximation coefficient of secret image. The differences of these coefficients are called error blocks. The replacement of an error block is being done with the best matched CH block [17].

Though spatial domain methods are more difficult and slower than spatial domain methods, yet they provide more security [1]. In this work two spatial domain methods EMD and GEMD will be discussed in details which similar to frequency domain methods provide greater security by the use of data transformations but in the spatial domain. In addition, in [7] Kuo and Wang provided a comparison between EMD and GEMD methods over different values of n-pixel group using the

metrics Peak Signal to Noise Raito (PSNR), Mean Square Error (MSE), and embedding capacity Bit Per Pixel (BPP). Since they considered the embedding capacity for GEMD is better than EMD, but the comparison in this case is not valid because the parameter n has different meaning for EMD and GEMD. For EMD n is the number of pixels required for one digit among k digits in one block, but for GEMD it is the total number of pixels required for one block, where in EMD total number of pixels required for one block is ink not just n as in GEMD. So, we need to analyze and experimentally study these methods as they exploited the modification of direction with bit differences in embedding and extracting processing, also we compare the performance of them using the averages of the metrics PSNR, MSE, and BPP in addition to the memory and time consumption.

A major characteristic of the EMD method is that it uses a separate n-pixel group of a cover image to embed the next digit of (2n+1)-ray k-digit number representing the next L-bit block from the secret image input and only one pixel in the group can be changed by ±1. In GEMD scheme the next L-bit block, L=n+1, is hidden in the next n-pixel group, and more than one-pixel value in a group may be changed by ±1, so the image quality for it may be lower than that for EMD.

EMD Embedding Algorithm

Begin

Inputs: cover image, CI (M, N); M is the number of rows; N is the number of columns; integer, n >0, pixel group size; integer, L >1, input binary stream block size; binary secret message, S.

Output: stage image, SI (M, N).

Step 1. Get next binary secret message block having L bits, and convert it to (2n+ 1)-ary k-digit number, where k is defined from the next relations

$$2L \leq (2n+1)\, k$$

$$L \leq \log_2 (2n+1)\, k$$

$$L \leq \lfloor k . \log_2 (2n+1) \rfloor$$

$$k = \left\lceil \frac{L}{\log_2 (2n+1)} \right\rceil \quad (1)$$

Step 2. For each digit s_i, i=1, …., k,

Begin

Get next pixel group from cover image, CI, X= (x_1, x_2..., xn), and calculate

$$t = ef(x) = \sum_{i=1}^{n} x_i \,.i \bmod (2n+1) \quad \text{..... (2)}$$

Calculate

$$d = (s_i - t) \bmod (2n+1) \quad \text{..... (3)}$$

Set

$$X' = X \quad \text{..... (3.1)}$$

If d= 0, nothing is made.

If $d \leq n$, increase the d^{th} pixel in the pixel group by 1:

$$x'_d = x'_d + 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.2)$$

Otherwise, decrease $((2n+1)-d)^{th}$ pixel in the pixel group by 1:

$$x'_{(2n+1-d)} = x'_{(2n+1-d)} - 1 \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (3.3)$$

End of step2.

Step 3. Go to Step 1 until the secret message is embedded.

Step 4. End. Data structure for EMD Embedding procedure is illustrated in Figure 1.

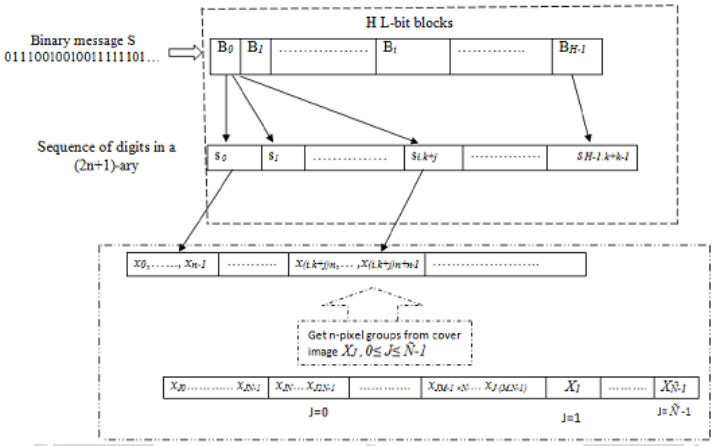

Figure 1. Data structure for EMD Embedding procedure

EMD Extraction Algorithm

Begin

Inputs: stage image, *SI (M, N)*; *M* is the number of rows; *N* is the number of columns; *integer, n>0*-pixel group size.

Outputs: binary secret message, *S*.

Step 1: Set *S*= {};*//empty set.*

Step 2. Obtain the next *n*-pixel block X'= ($x'_1, x'_2, \ldots\ldots x'n$) from stage image, *SI*.

Step 3. Calculate

$$s = ef(x'1,\ldots,x'n) = \sum_{i=1}^{n} x'_i \,.i \ mod\ (2n+1) \quad \ldots\ldots\ldots (4)$$

Step 4. Transform *s* into *L*-bit binary block and append it to the secret data stream, *S*. Go to **Step 2**.

Step 5. End. Data structure for EMD extraction procedure is illustrated in Figure 2.

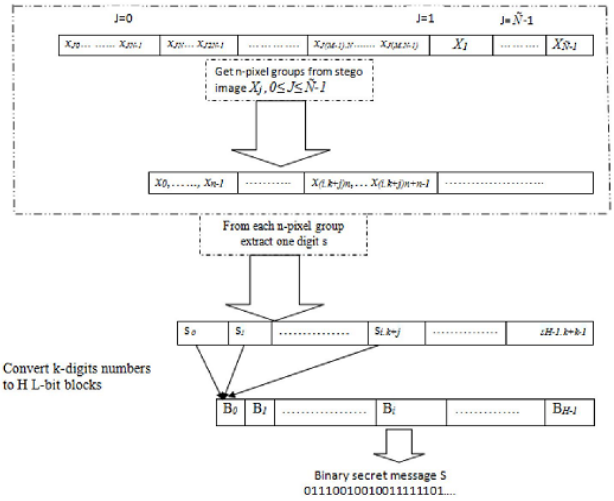

Figure 2.EMD Data structure for extraction procedure

GEMD Embedding Algorithm

Begin

Inputs: Cover image, *CI (M, N)*; *M* is the number of rows; *N* is the number of columns; integer, *n*>0, defining bit block and pixel group size; binary secret message, *S*.

Output: stage image, *SI (M, N)*.

Step 1. Get next *n*-pixel group $X= (x_1, x_2..., xn)$ from cover image *CI*

Step 2. Get next binary secret message, S, block having (*n*+1) bits with decimal value *s*.

Step 3. Compute $ef\ (x_1, x_2..., xn)$ with the pixel groups:

$$t = ef(x1, . . ., xn) = \sum_{i=1}^{n} x_i .(2^i - 1) mod\ 2^{n+1} \quad \ldots\ldots\ldots \quad (5)$$

Step 4. Compute the difference *d*

$$d= (s - t)\ mod\ 2^{n+1} \ldots\ldots\ldots\ldots\ldots\ldots\ldots\ldots \quad (6)$$

Step 5. If $d=2^n$ then $R = 1$;

else if ($d< 2^n$) then $R = 2$; else $R = 3$;

Step 6. Switch (*R*)

Case 1: Let $x'_n = x_n + 1$, $x'_1 = x_1 + 1$.

$x'_i = xi$, **I**=*2..., n-1*

Case 2: let $d = (d_n\ d_{n-1}\ d_{n-2} \ldots\ d_1\ d_0\ )_2$

for *i* =*n* down to 1 do

Begin

if ($d_i = 0$ and $d_{i-1} = 1$) then $x'_i = x_i + 1$;

else if ($d_i = 1$ and $di_{-1} = 0$) then x'i = xi - 1;

else *x'i* = *xi*

End.

Case 3: Let $d'=2^{n+1}$ -d.

Let $d'= (d_n\ d_n\text{-}1\ d_{n-2}\ldots\ d_1\ d_0\ )2$

for i =*n* down to 1 do

Begin

if (d_i = 0 and d_{i-1} = 1) then $x'_i = x_i - 1$

else if (di = 1 and di-1 = 0) then $x'_i = xi + 1$;

else x'i = *xi*

End.

Step 7. Go to **Step1** until secret the message is embedded.

Step 8. End. Data structure for GEMD Embedding procedure is illustrated in Figure 3.

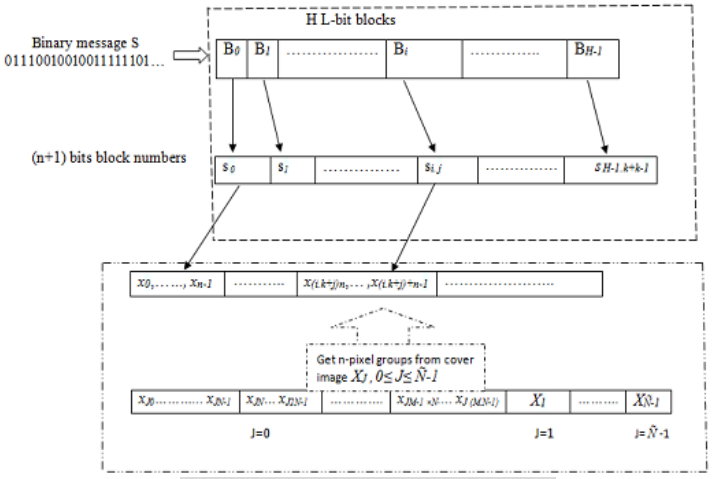

Figure 3. Data structure for GEMD Embedding procedure

GEMD Extraction Algorithm

Begin

Inputs: stage image, *SI (M, N); M* is the number of rows; *N* is the number of columns*;* integer, *n>0,* defining binary block and pixel group size.

Outputs: binary secret message, *S.*

Step 1. Set *S*= {}*;//empty set*

Step 2. Get next *n*-pixel group, *x= (x1, x2..., xn),* from stage image, *SI.*

Step 3. Calculate

$$s = ef(x'1, x'2, .., x'n) = \sum_{i=1}^{n} x'_i \cdot (2^i - 1) mod\ 2^{n+1} \quad (11)$$

Step 3. Append *s* as (*n*+1)-bit binary block to binary output secret data stream, *S.*

Step 4. If *SI* has not processed blocks, go to step2.

Step 5. End. Data structure for GEMD extraction procedure is illustrated in Figure 4.

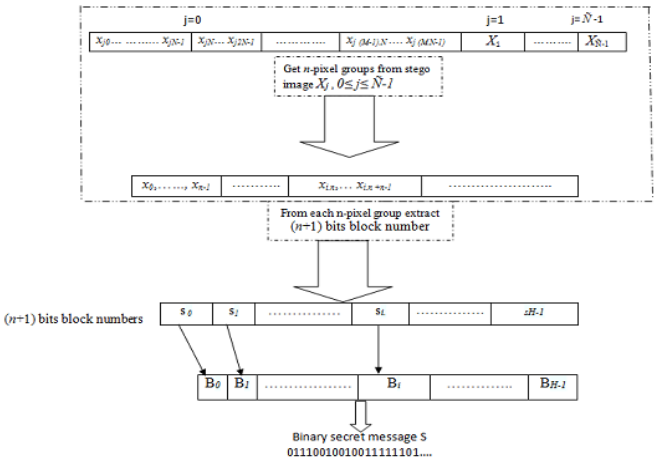

Figure 4. Data structure for GEMD extraction procedure GEMD

Known Experiments on EMD and GEMD

EMD and GEMD was evaluated using the following quality metrics.

1.Mean Square Error (MSE) is defined as mean squares differences between the original cover image and image after embedding [7]:

$$MSE = \frac{1}{M \times N} \sum_{r=1}^{M} \sum_{c=1}^{N} (CI(r,c) - SI(r,c))^2 \quad (7)$$

where M is the number of rows and N is the number of columns of the cover and stage images.CI

(r,c) is the original image pixel value and SI(r,c) is stage image pixel value .

2. Signal Peak to Noise Ratio (PSNR) is calculated as follows

$$PSNR = 10log_{10} \frac{255 \times 255}{MSE} \text{ dB} \quad (8)$$

where 255 is the maximum value of pixels for grey scale images.

3.Embedding capacity Bit Per Pixel (BPP) is defined as the number of secret bits embedded in each pixel of cover file. For EMD, log2(2*n*+1) bits that represent a (2*n*+1)-ray digit embedded in n pixels, while in GEMD (*n*+1)- bit values are embedded in *n*-pixel group [14]. BPP is calculated for EMD and GEMD as follows [7]:

$$\text{BppEMD} = \frac{log_2(2n+1)}{n} \quad (9)$$

Where number of bits embedded = $\log_2(2n+1)$

$$\text{BppGEMD} = \frac{n+1}{n} \quad (10)$$

In the experiments conducted in this paper, we tried to find the best values of EMD and GEMD parameters that achieved the best results with minimum number of cover images, and the comparison between both methods will be taken as the average over the metrics PSNR, MSE, BPP, time and memory consumption.

Experimental Results

Gray scale secret images and cover images of size 512×512 used in the experiments are shown in Figure5 and Figure 6. respectively

Figure 5: secret images (1) Balloon; (2) Tiffany; (3) Boat; (4) Pepper

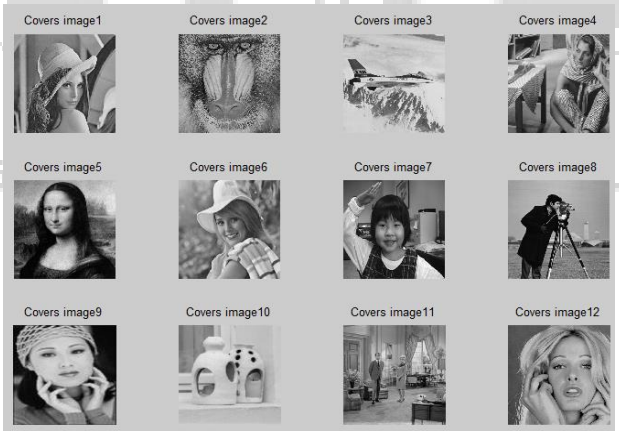

Figure 6: cover images used in EMD and GEMD simulation

EMD Simulation

EMD optimal parameters used in the simulations are given in Table 1.

Table 1: EMD parameters of the simulation

parameter	Number of pixels *n* for one digit			
	2	3	4	5
L bits	16	16	32	64
k digits	7	6	11	19
cover images $\tilde{N}$	7	10	11	12

Where *k* is calculated as (1) , and number of cover images $\tilde{N}$ is calculated as following:

$$\tilde{N} = |CI| = \left\lceil \frac{H \times k}{C} \right\rceil \quad (12)$$

Where the number of blocks H is defined as follows:

$$H = \left\lceil \frac{|S|}{L} \right\rceil \quad (13)$$

Table 2 shows the EMD average results using different values of *n*. As in some cases we have two averages, fully used and fully set, where the average fully used indicates to the averages of PSNR, MSE, memory and time consumption for the fully images used (without the last image that not fully embedded), while the average fully set refers to the averages for the fully images set (with the last image that not fully embedded).

Table 2: EMD average results

Metric	average	Number of pixels n for one digit			
		2	3	4	5
PSNR (dB)	Fully used	52.11	53.57	54.66	55.53
	Fully set		58.45		55.70
MSE	Fully used	0.40	0.28	0.22	0.18
	Fully set		0.25		0.17
Time (sec)	Fully used	7.81	6.38	4.66	3.97
	Fully set		5.76		3.82
Memory (MB)		481	486	490	496
Capacity (BPP)		1.16	0.93	0.79	0.69

GEMD Simulation

In GEMD, we take $L=n+1$ bits to embed in n pixels. GEMD parameters are given in Table 3.

Table 3: GEMD parameters of the simulation

parameter	Number of pixels n for one block L			
	2	3	4	5
L bits	3	4	5	6
cover image $\tilde{N}$	6	7	7	7

Table 4: GEMD average results

Metric	average	Number of pixels n for one digit			
		2	3	4	5
PSNR (dB)	Fully used	50.16	50.79	51.01	51.09
	Fully set	52.55	52.30	52.11	51.97
MSE	Fully used	0.62	0.54	0.51	0.50
	Fully set	0.42	0.41	0.41	0.42
Time (sec)	Fully used	4.61	3.77	2.92	2.02
	Fully set	4.26	3.31	2.47	1.97
Memory (MB)		491	493	496	498
Capacity (BPP)		1.50	1.33	1.25	1.20

Table 5:The EMD-versus-GEMD comparison results

method	Metric				
	PSNR (dB)	MSE	Time (Sec)	Memory (MB)	Capacity (BPP)
EMD	53.97	0.27	5.71	488	0.89
GEMD	50.77	0.54	5.36	485	1.32

From Table 5 we find that EMD stage image quality PSNR is better than 53 dB, since in embedding procedure only one pixel among n-pixel group is modified, while in GEMD it is nearly 51 dB, because more than one pixel in each n-pixel group could be modified. So in both sizes PSNR in EMD is better than GEMD by 0.06. For MSE comparison result, EMD has less error than GEMD by 0.5%, since fewer pixels are changed. On the other hand, GEMD is better in
embedding capacity, BPP, by 0.33%. GEMD has less memory and time consumption. GEMD is better in memory and time consumption by 0.006% and 0.06% respectively.

CONCLUSION

This thesis analyzes two steganographic methods; EMD and GEMD. The algorithms are explained in details such as the input, output, data structure, and the best values for their parameters which required a minimum number of cover images to maintain good image quality PSNR and minimum MSE, time and memory consumption. The results were obtained for four gray scale secret images, where the number of cover image required for a secret image is defined according to some parameters such as the number of bits in each block of secret image, and the number of pixels n in each group of the cover image. The experiments were conducted with four different values of n, as we tried to find the best value for the number of bits in each block L of the secret image and the maximum digit k in $(2n+1)$-ray in each case of n to achieve the best case of EMD and GEMD which taking less number of cover image. According to our analysis, EMD stage image quality PSNR is better than GEMD, since fewer pixels values are modified. On the other hand, GEMD has less memory and time consummation. For MSE comparison result EMD has less error than GEMD, because in EMD at most only one pixel is changed by ±1 in a group, while in GEMD more than one pixel in a group could be modified. But also, GEMD has greater embedding capacity. In addition to GEMD required less number of cover images

As a comparison between the results using different size of cover image, we find that as we use greater size then we need less number of cover images, and less time consumption. On the other hand, we need more memory consumption. For other metrics, PSNR, MSE, and embedding capacity BPP, we get the same results. However, the both methods have the same aim for hiding data, but one of them, EMD, is better in image quality, PSNR and MSE, while GEMD is better in memory and time consumption and also better embedding capacity.

References

[1] Cheddad, A., Condell, J., Curran, K., & Kevitt, P. M. (2010, October). Digital Image Steganography: Survey and Analysis of Current Methods. *Signal Processing*, pp. 727-752. Vol 90. No.3.

[2] Devi, M., & Sharma, N. (2014, March). Improved Detection of Least Significant Bit SteganographyAlgorithms in Color and Gray Scale Images. *IEEE, Recent Advances in Engineering and Computational Sciences (RAECS)* , pp. 1-5. Vol 34. No.7.

[3] Hegde, R., & S, J. (2015, July). Design and Implementation of Image Steganography by Using LSB Replacement Algorithm and Pseudo Random Encoding Technique. *International Journal on Recent and Innovation Trends in Computing and Communication*, pp.4415 - 4420. Vol 3. No.7.

[4] Jarno, M. (2006, May). LSB Matching Revisited. IEEE, Signal Processing Letters, pp. 285- 287. Vol 13 No.5.

[5] Kuo, W.-C., & Wang, C.-C. (2013, October). Data Hiding Based on Generalised Exploiting Modification Direction Method. *The Imaging Science Journal*, pp.484-490. Vol 61. No.10.

[6] Kuo, W. C., Chen, Y. H., & Chuang, C.-T. (2014, April). High-Capacity Steganographic Method Based on Division Arithmetic and Generalized Exploiting Modification Direction. *Journal of Information Hiding and Multimedia Signal Processing*, pp. 213-222. Vol 5. No.2.

[7] Kuo, W. C., Wang, C. C., & Hou, H. C. (2015, August). Signed Digit Data Hiding Scheme. *Information Processing Letters*, pp. 15-26. Vol 5. No.2.

[8] Kieu, T. D. & Chang, C. C. (2011, April) A Steganographic Scheme by Fully Exploiting Modification Directions, Expert Systems with Applications, pp.10648-10657. Vol 38. No.8.

[9] Lee .C.F; Wang. Y & Chang. C (2007, August). A Steganographic Method with High Embedding Capacity by Improving Exploiting Modification Direction. *Proceedings of the Third International Conference on Intelligent Information Hiding and Multimedia Signal Processing(IIHMSP07)*, pp.497-500. Vol 5. No.2.

[10] Pan, H. K., Tseng, Y. C & Chen Y. Y. (2002, August). A Secure Data Hiding Scheme for Binary Images. *IEEE Trans. Commun.*, pp. 1227-1231. Vol. 50. No.8.

[11] Rita, C. & Deepika, B. (2014, September), An Improved DCT based Steganography Technique, International Journal of Computer Applications, pp. 46-49. Vol 102. No.14.

[12] Wang, R. Z; Lin, C. F.; & Lin, J. C (2001, May). Image Hiding by Optimal LSB Substitution and Genetic Algorithm. *Pattern Recognition*, pp.671-683. Vol 34. No 3.

[13] Wu, D. C., & Tsai, W. H. (2003, April). A Steganographic Method for Images by Pixel-Value Differencing. *Pattern Recognition Letters*, pp.1613-1626. Vol 24. No.9.

[14] Wu, H. C., Wu, N. I., Tsai, C. S., & Hwang, M. S. (2005, March). Image Steganographic Scheme Based on Pixel-Value Differencing and LSB Replacement Methods. *IEEE, image signal process*, pp. 611–615.Vol 152. No. 12.

[15] Zhang, X., & Wang, S. (2006, November). Efficient Steganographic Embedding by Exploiting Modification Direction. *IEEE Communication Letters*, pp. 781-783. Vol 12. No.7.

[16] Zhi, H. W., Kieu,T.D.,& Chin, C.C. (2010, January), A Novel Information Concealing Method Based on Exploiting Modification Direction, *Journal of Information Hiding and Multimedia Signal Processing*, pp.130-138, Vol 1. No.1.

[17] Vijay, K. & Dinesh, K. (2010, June), Performance Evaluation of DWT Based Steganography, *IEEE 2nd International Advance Computing Conference,* pp. 223-228. Vol 6. No.10.