Performance Comparison of Proposed Embedding Method and Pixel Mapping Method

Su Mon Thu[#1], Su Wai Phyo[*2]

[#]*Information Technology Department, Mandalay Technological University*
The Republic of the Union of Myanmar
[1]susu052@gmail.com
[2]suwaiphyo@gmail.com

Abstract **- Data hiding is one of the challenging issues in the field of network security. Unlike cryptography, steganography is used to hide the existence of secret message by embedding the message behind any cover object like image, text, audio and video files. In many research areas, various information hiding techniques are developed in order to meet their security requirements. This paper proposes a new embedding method by optimizing pixel mapping method (PMM) in order to obtain larger embedding capacity and produce stego image quality with minimum degradation. To prove the better performance of the proposed embedding method, the comparison analysis between proposed embedding method and PMM will be performed according to the values of Peak Signal to Noise Ratio (PSNR) and embedding capacity.**

Keywords **- Data hiding, Steganography, Pixel mapping method (PMM), Embedding capacity, PSNR.**

I. INTRODUCTION

With the exponential growth of the Internet usage, demand for effective information security techniques is increasing day by day. Digital image steganography is one of those techniques that are used for effective secret communications. In this technique, secret communication is achieved by embedding a message into a cover image and generating a stego image that carries a hidden text message [1].

There are different methods for data hiding in image steganography such as least significant bit method, pixel-value difference method, histogram modification and pixel mapping method that use spatial domain. Many works have been done in this area and many methods have been developed and reported [2], [3].

Among them, PMM embedding algorithm is popular method because it is employed in various useful applications such as commercial, medical imaging and military communication systems and so on where the information security is essential. It has medium embedding capacity when comparing other image steganographic methods.

In order to obtain better embedding capacity with less distortion of image quality, this work proposes a new embedding method based on PMM. To do the comparative study, the analysis of proposed embedding method and PMM is performed in term of embedding capacity and PSNR.

The rest of the paper is organized as follows: some related works are described in section II. Pixel mapping method (PMM) is presented in section III. Section IV and V represent the proposed system architecture and implementation results. The performance consideration of this system is shown in section VI. Finally, section VII draws the conclusion.

II. RELATED WORKS

Stegnaographic methods were developed to hide secret information behind gray scale images from the data communication and information security areas. So, the benefits and weakness of these techniques are clearly studied according to their analytical results and conclusion.

In 2011, Souvik Bhattacharyya and his fellows [4] proposed a new image based steganographic method "Pixel Mapping Method (PMM) Based Bit Plane Complexity Segmentation (BPCS) Steganography". The combination of PMM and BPCS produces a robust image based steganography method which is independent of the information to be hidden and obtain a stego image. Due to their analytical results, the embedding capacity is less than PMM and other methods although their system produces robust stegno image quality.

In 2013, Prince Kumar Panjabi and Parvinder Singh [5] proposed an enhanced technique "An Enhanced Data Hiding Approach Using Pixel Mapping Method (PMM) with Optimal Pixel Substitution Approach " that provides a better Peak Signal to Noise Ratio (PSNR) between cover image and stego image with good embedding capacity. Their approach is based on four modules – mapping rules, set classifier method, pixel selection method, and minimum differencing function to hide data within an image. Finally, they presented that their integrated proposed approach not only provides larger embedding capacity but also produces an acceptable stego image quality that can be seen by human eyes.

According to the literature, it is interested how to optimize PMM to achieve in both: (1) better embedding capacity and (2) less distortion of stego image quality. Therefore, this work is intended to provide an efficient embedding method based on single PMM without integrating other embedding methods.

III. PIXEL MAPPING METHOD (PMM)

PMM is a 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 selected embedding pixels or its neighbors lie at the boundary of the image or not. Data embedding is done by mapping each four bits of the secret message in each of the neighbor pixel based on some features of that pixel. Fig. 1 shows the mapping information for embedding four bits per pixel [6].

Msg Bit Seq	2nd Set-Reset Bit	3rd Set-Reset Bit	Pixel Intensity Value	No of Ones (Bin)
0000	even	even	even	even
0001	even	even	even	odd
0010	even	even	odd	even
0011	even	even	odd	odd
0100	even	odd	even	even
0101	even	odd	even	odd
0110	even	odd	odd	even
0111	even	odd	odd	odd
1000	odd	even	even	even
1001	odd	even	even	odd
1010	odd	even	odd	even
1011	odd	even	odd	odd
1100	odd	odd	even	even
1101	odd	odd	even	odd
1110	odd	odd	odd	even
1111	odd	odd	odd	odd

Fig. 1 Mapping technique for data embedding

Extraction process starts again by selecting the same pixels required during embedding. At the receiver side, other different reverse operations have been carried out to get back the original information [6].

One important point needs to be kept in mind that a specific order for selecting the neighbor pixels has to be maintained for embedding / mapping process and also for the process of extraction otherwise it would not be possible to retrieve the data in proper sequence. This sequence has been shown in Fig. 2 [6].

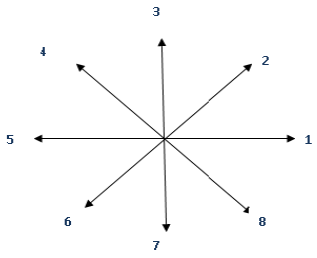

Fig. 2 Sequence of data embedding

A. *Pixel Selection Algorithm of PMM*

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 Factor) = 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

B. *Embedding Algorithm of PMM*

Let C be the original 8 bit gray scale image of size N × N i.e. C = (P_{ij} | 0 ≤ i < N, 0 ≤ j < N, P_{ij} € 0, 1, …, 255). Let MSG be the n bit secret message represented as MSG = (m_k | 0 ≤ k < n, m_k € 0, 1). A seed pixel P_{rc} can be selected with row (r) and column (c). Next step is to find the 8 neighbors $P_{r',c'}$ of the pixel P_{rc} such that r' = r + l, c' = c + l, -1≤ l ≤ 1. The embedding process will be finished when all the bits of every bytes of secret message are mapped or embedded. (Bincvr = Binary number of pixel intensity value (V))

Input : Cover Image(C), Message (MSG).
Find the first seed pixel P_{rc}.
count = 1.
while (count ≤ n)
begin (for embedding message in message surrounding a seed pixel).
m_k = Get next msg bit.
count = count + 1.
Mask the 5^{TH} bit from left with the m_k in 'Bincvr'
m_{k+1} = Get next msg bit.
count = count + 1.
Mask the 6^{TH} bit from left with the m_{k+1} in 'Bincvr'
cnt = Count number of ones of one of the $P_{r',c'}$ of intensity (V).
m_{k+2} = Get next msg bit.
count = count + 1.
m_{k+3} = Get next msg bit.
count = count + 1.
Bincvr = Binary of V.
If(m_{k+2} = 0 & m_{k+3} = 1)
Bincvr (zerothbit) = 0
If (cnt mod 2 = 0)
Bincvr (firstbit) = ¬ Bincvr (firstbit)
If (m_{k+2} = 1 & m_{k+3} = 0)
Bincvr (zerothbit) = 1
If (cnt ÷ 2 ≠ 0)
Bincvr (firstbit) = ¬ Bincvr (firstbit)
If (m_{k+2} = 0 & m_{k+3} = 0)
Bincvr (zerothbit) = 0
If (cnt mod 2 ≠ 0)
Bincvr (firstbit) = ¬ Bincvr (firstbit)
If (m_{k+2} = 1 & m_{k+3} = 1)
Bincvr (zerothbit) = 1
If(cnt mod 2 = 0)
Bincvr (firstbit) = ¬ Bincvr (firstbit)
End

Get the next neighbor pixel $P_{r'c'}$ for embedding based on previous $P_{r'c'}$ and repeat.

End

Return the stego image (S).

C. Extraction Algorithm of PMM

The process of extraction proceeds by selecting those same pixels 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 is described as follows. (Bincvr = Binary number of pixel intensity value (V))

Input : Stego image (S) , count.

count = count ÷ 2.

BinMsg = " ".

Find the first seed pixel P_{rc}.

I = 0.

while (count ≤ N)

begin (for extract message in message around a seed pixel).

Get the (First/Next) neighbor pixel $P_{r'c'}$.

cnt = Count number of ones of one of the $P_{r'c'}$ of intensity (V).

Bincvr = Binary of V.

Binmsg(i) = 3rd Bit of Bincvr from Right.

i = i+1.

Binmsg(i) = 2nd Bit of Bincvr from Right.

i = i+1.

Binmsg(i) =ZerothBit of Bincvr.

i = i + 1.

If (cnt mod 2 = 0) (i.e. it is even) Binmsg(i) = 0 Else Binmsg(i) = 1

Binmsg(i) = Enters according to One of ones in the intensity (1 for odd : 0 for even).

i = i + 1.

count = count + 1.

End.

Get the next neighbor pixel $P_{r'c'}$ for extracting based on previous $P_{r'c'}$ and repeat.

End loop.

Binmsg is converted back to Original message.

Return Original Message.

End.

D. Proposed Pixel Selection Algorithm

The heart of the PMM is pixel selection algorithm. So, the reason to propose pixel selection algorithm is to develop powerful embedding method. Like pixel selection algorithm of PMM, proposed algorithm depends on the pixel intensity value to find the seed pixel. Each four bit sequence of secret message is hidden in each neighbour pixel. Unlike pixel selection algorithm of PMM, it limits seed pixel intensity value ranging from 61 to 180 and then selects seed pixel position randomly. Therefore, it makes the number of seed pixels increases. As consequence, the embedding capacity is improved with minimum distortion of Stego image quality. The proposed pixel selection algorithm for embedding is described below:

Input: C, previous pixel position (x, y), pixel intensity value (v)

no: of char = number of characters contain (msg)

no: of seed pixels = ⌈no: of char/4⌉

n = 0

do

{

select random seed pixel intensity and position (not at boundary).

If (intensity > 60 && intensity < 181)

{

if (seed pixel position is not already contained in selected seed pixels).

save seed pixel position and its neighbor pixels position.

n = n + 1.

}

} while (n < no: of seed pixels)

Return previous pixel position (x, y) and its 8 neighbor pixels.

End

IV. Proposed System Architecture

The proposed system is organized with two portions. The first portion is general structure of data hiding approach which is depicted in Fig. 3. At first, the secret message is embedded in a cover image through an embedding algorithm (proposed algorithm or PMM). The output stego image is sent over a transmission channel to the receiver where it is performed by the extraction algorithm.

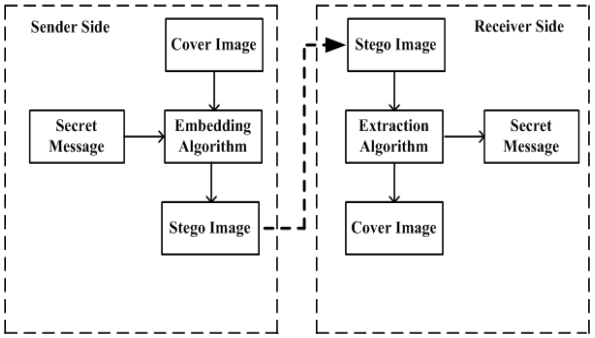

Fig. 3 General block diagram of data hiding approach

The second portion focuses on performance comparison as illustrated in Fig. 4. The interest of performance comparison falls on the output stego images of embedding processes. Firstly, the user loads the secret message. Secondly, the user needs to choose proposed embedding method and PMM alternatively for embedding process in order to yield stego images.

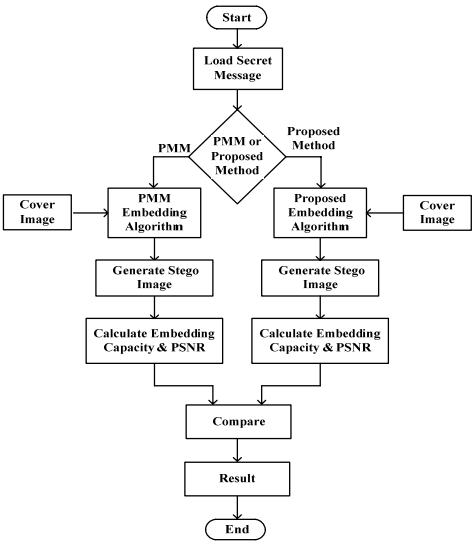

Fig. 4 Flow chart for performance comparison process

Finally, the embedding capacity and PSNR values of stego images which are generated from proposed embedding method and PMM are calculated and then the results are compared.

V. Implementation Results

This system is implemented as a series of interfaces by using C# programming language. Some of them are illustrated in this section.

To perform the embedding process using PMM, the secret message and selected cover image are inserted as inputs. Then, the message is hidden into the cover image as show in Fig. 5.

Fig. 5 Hiding message using PMM

After hiding the message into cover image, the stego image is generated. Then, the PSNR value and embedding capacity of PMM are calculated and displayed as shown in Fig. 6.

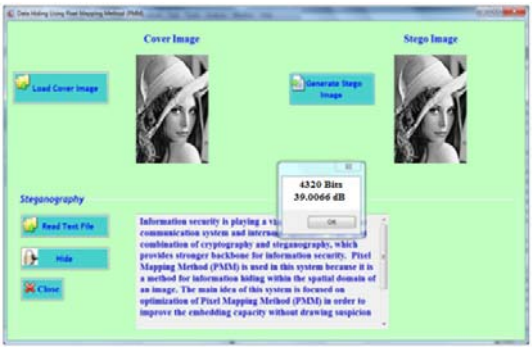

Fig. 6 Generating stego image and results using PMM

In the receiver side, the extraction process of PMM is illustrated in Fig. 7.

Fig. 7 Extracting message from stego image using PMM

Like the embedding and extraction processes of PMM, the stego image of proposed embedding method is also generated and the results are displayed as shown in Fig. 8, Fig. 9 and Fig. 10.

Fig. 8 Hiding message using Proposed Embedding Method

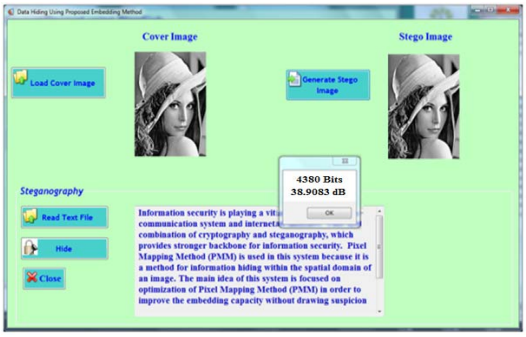

Fig. 9 Generating stego image and results using proposed embedding method

Fig. 10 Extracting message from stego image using proposed embedding method

VI. Performance Consideration

In this section, the experimental results are analyzed by two portions: embedding capacity consideration and the calculation of PSNR values for stego image.

A. Consideration on Embedding Capacity

The embedding capacity consideration depends on three parts: (1) number of seed pixels, (2) number of hidden bits in each pixel and (3) eight neighbors surrounding the seed pixels. The embedding capacity can be calculated as follows:

$$\text{Embedding capacity (bits)} = s \times t \times n$$

Where,

s = number of seed pixels,

t = number of hidden bits in each pixel and

n = eight neighbours surrounding the seed pixels

The experimental results are done in various sizes of different images. Depending on these experimental results, Table 1 represents the comparative study of embedding capacity for proposed embedding method and PMM. Therefore, it can be seen that the embedding capacity is improved in proposed embedding method.

TABLE I

Comparative Study of Embedding Capacity

Image	Image Size	Embedding Capacity (bits)		
		PMM	Proposed method	Differences (%)
Lena	128 × 128	4320	4380	1.3889
	256 × 256	19553	19614	0.3119
	512 × 512	90210	90268	0.0643
Pepper	128 × 128	5248	5307	1.1242
	256 × 256	22913	22978	0.2837
	512 × 512	92653	92715	0.0669

B. 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. It is assumed 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 is calculated as equation (1).

$$\text{MSE} = \frac{1}{[\text{N} \times \text{N}]2} \sum_{i=1}^{N} \sum_{j=1}^{N} [C(i,j) - S(i,j)]\,2 \qquad (1)$$

The PSNR is computed by using equation (2).

$$\text{PSNR} = 10 \log_{10} 255^2 / \text{MSE db} \qquad (2)$$

The comparative study of PSNR values between proposed embedding method and PMM is shown in Table 2. Like comparative study of embedding capacity, the experimental results are performed on various sizes of different images in this study.

TABLE II

Comparative Study of Psnr Values

Image	Image Size	PSNR (dB)		
		PMM	Proposed method	Differences (%)
Lena	128 × 128	39.0066	38.9083	0.252
	256 × 256	38.4749	38.3774	0.2534
	512 × 512	37.1579	37.0589	0.2664
Pepper	128 × 128	37.3678	37.2693	0.2636
	256 × 256	38.6321	38.5343	0.2532
	512 × 512	39.3494	39.2506	0.2511

According to the comparison results, it is found that the PSNR values of the proposed embedding method are slightly decreased when comparing that of PMM. Therefore, the proposed embedding method has higher embedding capacity and lower PSNR values.

VII. Conclusions

The performance comparison of proposed embedding method and PMM is presented in this paper. The analytical results show that the proposed embedding method not only provides larger embedding capacity but also produces stego image quality with minimum distortion. However, it processes only gray scale image. This proposed method can be applied in security awareness applications. As further extensions, it can be enhanced by integrating with other embedding methods to achieve more efficient method. In addition, it can be modified and tested to process on colour images also.

Acknowledgment

I would like to give my special thanks to my supervisor, Dr. Su Wai Phyo, Associate Professor of our university, MTU, (Mandalay Technological University) for her invaluable recommendations, detailed guidance and patient supervision throughout the preparation of this research. I would like to express my deepest thanks to all my teachers and colleagues who lend a hand directly or indirectly during the arduous process of completing this work successfully. I also show admiration to my parents for their mental supporting.

References

[1] Chetna Mehto, Rachana Kamble and Dr. Bhupesh Gour, "Investigation of digital image steganography: a survey," *Int. J. Computer Technology and Applications*, vol. 5, pp. 1711–1717, Sept- Oct. 2014.

[2] K. Sullivan, Z. Bi, U. Madhow, S. Chandrasekaran and B. S. Manjunath, "Steganalysis of quantization index modulation data hiding," in *Proc. IEEE*, 2004, p. 1165 - 1168.

[3] V. Kumar and S. K. Muttoo, "A graph theoretic approach to sustainable steganography," *MIS Review: An Int. Journal*, vol. 17, pp. 19–37, 2010.

[4] Souvik Bhattacharyya, Aparajita Khan, Aunkita Nandi, Aveek Dasmalakar, Somdip Roy and Gautam Sanyal, "Pixel mapping method (PMM) based bit plane complexity segmentation (BPCS) steganography," in *Proc. IEEE*, 2011, p. 36 - 41.

[5] Prince Kumar Panjabi and Parvinder Singh, "An enhanced data hiding approach using pixel mapping method with optimal substitution approach," *International Journal of Computer Applications*, vol. 74, pp. 36–43, July. 2013.

[6] Souvik Bhattacharyya and Gautam Sanyal, "A data hiding model with high security features combining finite state machines and PMM method," *International Journal,* vol. 4, pp. 324–331, 2010.