I.J. Image, Graphics and Signal Processing, 2014, 3, 12-20
Published Online February 2014 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2014.03.02

An Image Steganography Algorithm with Five Pixel Pair Differencing and Gray Code Conversion

Gulve Avinash K
Associate Professor, MCA Government College of Engineering, Aurangabad, Maharashtra, India. 431 005
akgulve@yahoo.com

Joshi Madhuri S
Professor, Computer Technology Jawaharlal Nehru Engineering College, Aurangabad, Maharashtra, India. 431 005
madhuris.joshi@gmail.com

Abstract — Steganography involves hiding information in another media. PVD based steganography techniques uses the difference between the pixel values of a pair directly to hide the information. The proposed steganography system modifies the difference value before being used for hiding the information. This makes extraction of hidden data harder in case the steganography system fails. The algorithm divides the cover image in the block of 2 x 3 pixels and calculates average (N) of the bits that can be hidden in five pairs of that block. Thus if the difference value allows M-bits to be hidden in the pair, then only N-bits are hidden in that pair when M >N otherwise M (if M≤N) bits are hidden in that pair. Second level of security is added by converting the secret information into gray code before embedding it in the cover image. The algorithm provides good hiding capacity and improved quality of stego image with two levels of security for the secret information.

Index Terms — Data hiding, Steganography, Average, Gray code, Pixel value difference (PVD), PSNR.

I. INTRODUCTION

With the development of computer and Internet technologies, and its use in different areas of life and work, it has become easier to share information cost effectively. People can share text, videos, audios, images without difficulty. It takes a few seconds to transfer the information electronically from one place to another. Therefore it is becoming more important to adopt security measures so that data is protected from being stolen. In the past methods like cryptography, steganography, coding etc have been used for secured exchange of data. However, during recent years, steganography has attracted more attention with adoption of various algorithms. Steganography involves hiding information so that it appears that no information is hidden. The valuable data is hidden into multimedia documents so that it becomes un-noticed. It is also relatively easy to place information in multimedia documents. In these documents, lots of redundant areas are available where valuable information could be placed in an imperceptive way. It is expected that the steganography systems should maximize the embedding capacity while maintaining the security of the data.

The PVD approach hides the secret information using the difference between pixel values of two consecutive pixels. The difference value is used to locate suitable range from the range table. The width of the range is used to estimate the amount of secret data that can be hidden in the pair. The two pixels in the pair are then adjusted so that the new difference value stands for the secret data. The security level of PVD based steganography can be further improved by encrypting the secret message before hiding it in the cover image. In such system, it is necessary to decrypt the extracted data for getting the original message. But it involves the overhead of encryption and decryption. Some steganographic methods are discussed in section II, which first encrypt the secret data and then hide this encrypted data in the cover image. Instead of encrypting the secret data, the same level of security can be achieved by converting the form of the secret data.

The proposed method uses the concept of code conversion to improve the performance of the steganographic system. The secret data is converted into binary form before hiding it in the cover image. The Gray code, also known as reflected binary code, is a binary numeral system where two successive values differ in only one bit. It is possible to convert secret data in binary form into its gray code equivalent form before hiding.

The proposed method distorts the original secret data by converting it into gray code form and then hides the distorted data in the cover image to produce the stego image. The distorted data is extracted from the stego image and it is then converted to its binary form to get the original data. The proposed method provides two levels of security for the secret message: (1) the original difference between pixel values of two consecutive pixels is modified and then it is used to hide the secret information. (2) The secret message is hidden in the form of gray code. Instead of converting the secret message into gray code form at a time, it is converted in parts. The combination of steganography and code conversion provides two tier security for the secret data. Another level of security is provided by modifying the difference value of the pixels in the pair. Thus even if the steganography system fails, it will be hard for the intruder to extract the original message.

The remainder of the paper is organized as follows. In section II, a review of the necessary background required to effectively implement our algorithm is presented. The proposed algorithm is described in Section III. After that, results of the proposed algorithm are discussed in section IV, and conclusion is provided in the last section.

II. BACKGROUND

Most of the steganography techniques use images as cover media because transmission of images through email or sharing of images through web applications is becoming very general. Bender *et al.* [1] has discussed some data hiding techniques using least significant bits. Least significant bit (LSB) insertion is a common and simple approach to embed information in a cover object. For images as covering media, the LSB of a pixel is replaced with a message's bit. To the human eye, the resulting stego image looks identical to the cover image [2]. The LSB-based methods directly embed the secret data into the image without taking into consideration the difference in hiding capacity between edges and smooth areas. A steganography system proposed by Sutaone *et al.* [3] is designed for encoding and decoding a secret file embedded into an image file using random LSB insertion method. The secret data is spread out among the cover image in a seemingly random manner. The key is used to generate pseudorandom numbers, which identifies where, and in what order, the hidden message is laid out.

There are two types of LSB insertion techniques- fixed size and variable size. The fixed size method embeds the fixed number of message bits in each pixel of the cover image. Based on number of LSB's used for hiding the information, LSB based image steganography methods are classified as 1 bit stego, 2 bit stego, 3 bit stego and 4 bit stego [4]. For the variable-size embedding method, the number of LSB's in each pixel used for hiding message depends on the contrast and luminance characteristics. Lee *et al.* [5] has proposed a steganography model based on variable length LSB insertion to enhance the embedding capacity. In each pixel of a gray-scale image, at least four bits of the message can be embedded. Maximum embedding capacity of each pixel is determined based on contrast and luminance characteristics of that pixel and then the minimum-error replacement method is adopted to find a grey scale as close to the original one as possible. Finally, the improved gray scale compensation, which takes advantage of the peculiarities of human visual system, is used to eliminate the false contouring effect. The author has proposed two methods, pixel wise and bit wise, to deal with the security issue [5].

In pixel value difference method, the difference of pixel values between two consecutive pixels is calculated. This difference value is used for hiding the secret message. The difference value is compared with the bits of secret message to be hidden. If they are unequal then the two consecutive pixels are directly adjusted so that their difference value can stand for the secret data. However, considerable stego-image distortion can happen when the PVD method adjusts the two consecutive pixels to hide the secret data in the difference value. Two benchmarks are adopted by steganography 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 pixel-value differencing (PVD) method proposed by Wu and Tsai successfully provides both high embedding capacity and outstanding imperceptibility for the stego-image [6].

Zaker *et al.* [7] has proposed modification in the original PVD method so as to make it more resistive to histogram attacks. Among the two rules suggested by the author, first rule causes the absolute difference between two pixels of a block to be less than or equal to its initial value. And the other range overlapping rule is used to allow some new difference values to be shifted to the left neighbour range which increases the embedding capacity. In Wu and Tsai's method [6], edged and smooth areas are identified using a pixel-value differencing (PVD) method. The difference between pixel values for a pixel pair in edged area is large whereas difference between pixel values in smooth area is small. This difference is used for estimating the amount of message bits that are to be hidden in that pair. The capacity of hidden data in edged areas is higher than that of smooth areas. However, to increase the hiding capacity of the cover image, Wu *et al.* [8] has proposed a approach that uses fixed-size least-significant-bits (LSB) method to hide data in smooth area whereas PVD method is used to hide data in the edged areas.

In the PVD method, two vertical/ horizontal and consecutive pixels represent a vertical/ horizontal edge, but the edge can have different directions. This motivates to improve the PVD method by considering three directions [9]. In the steganography method suggested by Chang *et al.* [9][10], data can be hidden in vertical and diagonal edges along with the horizontal edges. The whole image is divided into non-overlapping blocks of 2 × 2 pixels. One of the pixels is used to form 3 pixel pairs with remaining 3 pixels in the block. Each 2 x 2 block includes four pixels $P_{(x,y)}$, $P_{(x+1,y)}$, $P_{(x,y+1)}$ and $P_{(x+1,y+1)}$ where x and y are the pixel location in the image. Assuming pixel $P_{(x,y)}$ to be the starting point, the three pixel pairs can be formed by grouping $P_{(x,y)}$, with the right, the lower, and the lower right neighbouring pixels. Those three pairs are named by P_0, P_1 and P_2 where $P_0 = (P_{(x,y)},P_{(x+1,y)})$, $P_1= (P_{(x,y)}, P_{(x,y+1)})$ and $P_2 = (P_{(x,y)}, P_{(x+1,y+1)})$, respectively. For each pair difference value d_i is calculated. The block with a small value of d_i indicates the smooth area, whereas a block with a large value of d_i is considered as sharply edged block. According to the properties of human vision, eyes can tolerate more changes in the sharply edged area than in the smooth area so that more data can be embedded into the edged areas than into the smooth areas. While using this method to embed the secret data, pixel values of both the pixels in each pair gets modified and for each pair, a new difference value d'_i for i = 0, 1, 2 is calculated [9]. The new pixel values in each pair are different from their original values. That is, three different values are obtained for the starting pixel $P_{(x,y)}$. However, in every block, only one value for starting pixel $P_{(x,y)}$ can exist. Therefore, one of the P_i is selected as the reference pair to offset the other pixel values. That is, two pixel values of one pair are used to adjust the values of pixels in other two pairs and construct a new 2x2 block. The embedded secret data is unaffected because new difference values for three pixel pairs are unaltered [9][10].

Asmari *et al.* [11] proposed a steganography method based on pixel value differencing and LSB substitution. The cover image is divided into sub-blocks of 4x4 pixels each. The data is hidden in two consecutive pixels vertically depending on the pixel value difference. The embedding process begins with hiding 3 LSB in each pixel at corner. Thus 12 bits are directly hidden in the 4 pixels at corner with LSB substitution method. Then remaining 12 pixels form the semi hexagonal shape. The embedding of data is applied on two consecutive pixels vertically. The embedding process determines the range for each pair. If the range is higher, then PVD method is used for hiding the data, otherwise 3 bits are directly hidden in each pixel of the pair. This method offers higher capacity for data hiding and produces stego images of good quality [11].

Phad et al. [12] has proposed a high security model combining cryptographic and steganography systems. The security provided by cryptography and steganography is combined to give two tier security to secret data. The secret information is encrypted by using advance encrypted standard (AES) encryption algorithm. The encrypted message is then embedded into cover image by using PVD steganography and K-bit LSB substitution method.

Khalil Challita et al. [13] has proposed a steganography method in which both the sender and the recipient agree on a cover image to send a secret message. Instead of modifying the cover image, the method determines the bits of the secret message that match the ones in the cover image and stores the locations of such bits in a vector. The vector is then encrypted by using a classical cryptography tool and sent to the recipient. So even if the vector is intercepted, it will be difficult for the steganalyst to extract the secret data since he will have no idea about the cover image. The other method proposed by the author hides a secret message in more than one cover object. The secret message is hidden in k > 0 cover images. The greater the value of k, the harder it will be for a steganalyst to extract the secret message. A secret key algorithm is used for hiding the secret message in k cover images which is shared by both, the sender and recipient. So the steganalyst must have the information about the number of cover images (k) used for hiding the secret data, all the stego-objects created after hiding secret data, and the algorithm (i.e. secret key) used to hide the secret data in the cover-images to extract the secret data.

Dipti Kapoor Sarmah et al. [14] has suggested a steganography method that combines cryptography method with steganography. The secret message is first distorted and this distorted message is hidden in the cover image. For retrieval of the hidden message from the stego image, first the distorted message is extracted and then reverse of the distortion process is used to get the actual message.

AES algorithm using 128 bit key is used to encrypt the secret message. The encryption process generates the cipher texts in hexadecimal form. The alphabets and digits are separated from the cipher text. A secret key is generated to keep track of the original position of the alphabet and the digits. Another key is generated by separating first seven alphabets and adding the remaining alphabets at the end of the separated digits. The seven alphabets are scrambled using 64 bit key. Instead of hiding the complete encrypted text into an image, a part of the encrypted message is hidden in the DCT coefficients of the image. The 56 larger positive coefficients from the low-mid frequency range are selected. The selected coefficients are ordered by magnitude and then modified by the corresponding bit in the message stream. Unhidden part of the encrypted message is converted into two secret keys.

III. THE PROPOSED ALGORITHM

A. Conversion from Binary to Gray and Gray to Binary-

The term “Gray code” is used to refer to any single-distance code, that is, one in which adjacent code words differ by 1 in one digit position only.

The binary word can be converted to gray code using equation (1)

$$G_i = B_{i+1} \oplus B_i,\ i = n-1, \ldots\ldots, 0,\ \text{where } B_n \text{ is taken as } 0 \qquad (1)$$

And the gray code can be converted to binary code using equation (2)

$$B_i = B_{i+1} \oplus G_i,\ i = n-1, \ldots\ldots, 0,\ \text{where } B_n \text{ is taken as } 0 \qquad (2)$$

Extra level of security can be added in steganography systems by encrypting the secret message. But overhead of encryption and decryption is involved in such systems. Instead, the proposed system converts the binary equivalent of the secret message into gray code form and hides the gray code form of secret message in the cover image. Thus the proposed system provides the same level of security without introducing the overhead of encryption and decryption.

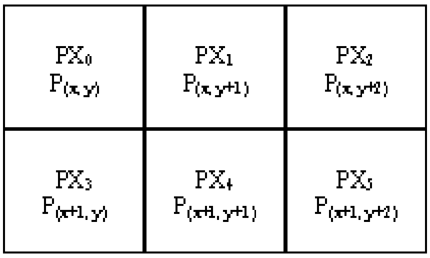
B. The Embedding and extraction algorithm-

The algorithm extends the approach suggested by Chang [9][10] to improve the hiding capacity and security of the steganography method. In the PVD approach, the data is embedded in the pixel pair using the difference between the pixel values in that pair. Amount of data hidden in the pixel pair is directly proportionate to the difference in the pixel values. Due to the small difference between the pixel values in the pair in smooth area, it is possible to hide small amount of data whereas large amount of data can be hidden in the edged area. Since the difference value is directly used to hide the data, it is easy to retrieve the embedded data in case the steganography system fails. To enhance the security of the hidden data, the proposed algorithm modifies the difference between the pixel values in the pixel pair and this modified difference value is used to hide the message. This imposes extra layer of security making harder extraction of original secret data from stego image using the difference values directly.

The cover image is a grey scale image, which is divided into non-overlapping 2 x 3 blocks of pixels. Five pairs of pixels are formed and these pairs are used to embed the secret data. The arrangement of pixels into non-overlapping blocks of 2 x 3 pixels is shown in figure 1.

As shown in figure 1, each 2 x 3 block includes six pixels $P_{(x,y)}$, $P_{(x,y+1)}$, $P_{(x,y+2)}$, $P_{(x+1,y)}$, $P_{(x+1,y+1)}$ and $P_{(x+1,y+2)}$ where x and y are the pixel locations in the image. Let $P_{(x,y+1)}$ be the starting point, then five pixel pairs can be formed as

P0 = ($P_{(x,y)}$,$P_{(x,y+1)}$),
P1 = ($P_{(x,y+2)}$,$P_{(x,y+1)}$),
P2 = ($P_{(x+1,y)}$,$P_{(x,y+1)}$),
P3 = ($P_{(x+1,y+1)}$,$P_{(x,y+1)}$)
P4 = ($P_{(x+1,y+2)}$,$P_{(x,y+1}$).



PX_0 $P_{(x,y)}$	PX_1 $P_{(x,y+1)}$	PX_2 $P_{(x,y+2)}$
PX_3 $P_{(x+1,y)}$	PX_4 $P_{(x+1,y+1)}$	PX_5 $P_{(x+1,y+2)}$

Figure 1. Pixel Block

The difference value d_i is calculated for each pixel pair P_i. This difference value is used to identify the range $R_{k,i}$ from the range table R. The range table is designed with ranges [0-7], [8-15], [16-31], [32-63], [64-127], [128-255]. The width $W_{k,i}$ of the range $R_{k,i}$ is used to determine the number of bits t_i (t_i = | $\log_2 W_{k,i}$ |) that can be hidden in each pair. This t_i is then used to calculate the average value (*N*) of number of bits possible to be hidden in each block. The average value *N* is used to calculate the revised difference $d1_i$ as $d1_i$ = remainder ($d_i/2^N$) so that $d1_i <= 2^N$ where d_i is the original difference. The offset difference OD_i is calculated as $|d_i|$- $|d1_i|$ for each pair in the block. The revised difference $d1_i$ is then used to determine the number of bits t_i for each pair in the block. Thus if the original difference value d_i allows *M* (if $M > N$) bits to be hidden in the pair, then only *N* bits are hidden in that pair otherwise *M* (if $M \leq N$) bits are hidden in that pair. Thus $t_i = M$ if $M \leq N$ otherwise $t_i = N$ if $M > N$.

The ti bits of secret data for each pixel pair are then converted into gray code form. After embedding ti bits (in gray code form) of the message in the pixel pair, new difference d' is calculated as ODi+li+b where li represents lower boundary of the range Ri in the range table R and b represents the decimal equivalent of ti message bits hidden in that pair.

Embedding ti bits in the pair modifies the values of both the pixels in the pair. The new pixel values in each pair are different from their original values. That is, five different values are obtained for the common pixel. However, in every block, only one value for common pixel can exist. Therefore, one of the Pi is selected as the reference pair to determine the other pixel values. The difference m between di and d'i is calculated. Small value of |m| indicates that the new difference value d'i is close to original difference value di. Thus new pixel values of the pair with minimum |m| are close to the original pixel values. So the pair with minimum |m| is selected as reference pair. Two pixel values of the reference pair are used to adjust the values of pixels in other pairs and construct a new 2 x 3 block. The embedded secret data in newly constructed block is unaffected because difference values for three pixel pairs are unaltered.

During the extraction process, average value (N) is calculated using the same process adopted during embedding of the secret message. The average value N is used to calculate the revised difference d1'i as d1'i = remainder (di/2N). Suitable range Rk,i is identified using this revised difference. The secret message is extracted in the decimal form by subtracting | d1'i | from lk. The secret message is then converted into a binary stream with ti (ti = | log2Wk,i |) bits. Since this binary stream is in gray code form, it is converted to binary code to get the original secret message.

The details of data embedding steps are described as follows.

1. Read the cover image pixel values in 2-dimensional decimal array.

2. Partition the array into non-overlapping blocks of 2 x 3 pixels
3. Calculate the difference values d_i for the five pixel pairs in each block

$$
\begin{aligned}
d_0 &= P_{(x,y)} - P_{(x,y+1)}\\
d_1 &= P_{(x,y+2)} - P_{(x,y+1)}\\
d_2 &= P_{(x+1,y)} - P_{(x,y+1)}\\
d_3 &= P_{(x+1,y+1)} - P_{(x,y+1)}\\
d_4 &= P_{(x+1,y+2)} - P_{(x,y+1)}
\end{aligned}
$$

4. Use | d_i | where i = 0,1,2,3,4 to locate suitable range R_k in the designed range table. Use this range to calculate number of bits t_i that can be hidden in each pair P_i. Then calculate the average bits using the equation (3)

$$\text{avg} = (\text{floor}\left(\sum_{i=1}^{5} t_i\right))/5 \qquad (3)$$

5. Calculate the revised difference $|d1_i|$ where i = 0,1,2,3,4 as $d1_i$ = remainder($d_i/2^{avg}$) so that $d1_i <= 2^{avg}$.
6. Calculate the difference OD_i as $OD_i= |d_i |– |d1_i |$for each pixel pair.
7. Use | $d1_i$ | where i = 0,1,2,3,4 to locate suitable range R_k in the designed range table.
8. Compute the number of bits t_i that can be embedded in each pair using the corresponding range given by R_k. The value t_i can be estimated from the width w_k of R_k, which is given by t_i = | $\log_2 w_k$ | where width $w_k = u_k-l_k+1$ and u_k and l_k are upper and lower boundaries of the range R_k
9. Read t_i bits from the binary secret data. Convert t_i bits to its gray code equivalent and transform this bit sequence into a decimal value b.
10. Calculate the new difference value d'_i given by (4)

$$
\begin{aligned}
&d'i = ODi + lk + bi, \text{ if } di \geq 0\\
&d'i = - (ODi + lk + bi), \text{ if } di < 0
\end{aligned}
\qquad (4)
$$

11. Modify the values of pixels in pixel pair P_i by using the equation (5).

$$(P'_n, P'_{n+1}) = (P_n - \lceil m/2 \rceil, P_{n+1} + \lfloor m/2 \rfloor) \qquad (5)$$

where Pn and Pn+1 represents two pixels in the pair Pi and m is the difference between di and d'i.

12. Select the pair with minimum | m | as the optimal reference pair and use this pair to adjust the pixel values of the other four pairs. Thus new values are assigned to remaining four pixels in the block.
13. Check the new pixel values for fall-off boundaries i.e. check whether all the pixel values are within the range 0 to 255. If not, modify the pixel values preserving the difference between the pixel values of each pair in the block.
    a. Find out smallest of all the pixel values. If smallest is less than 0 then add **|smallest|** in all the pixel values in that block.
    b. Find out largest of all the pixel values. If largest is greater than 255, subtract **largest-255** from all the pixel values in that block.
    c. If fall-off boundary problems still exist, the cover image is not suitable for hiding secret information.
14. Now, reconstruct the block from all pixel pairs with modified pixel values.
15. Repeat steps 2 through 14 till the message gets embedded in the cover image.

Following algorithm describes how to retrieve the embedded secret data from the stego-image. The extraction process is blind. It does not require original cover image for extracting hidden information from the stego image.

1. Read the cover image pixel values in 2-dimensional decimal array.
2. Partition the array into non-overlapping blocks of 2 x 3 pixels. Keep the partition order same as that of data embedding.
3. Calculate in difference values separately for each block in the stego-image given by.

$$
\begin{aligned}
d_0 &= P_{(x,y)} - P_{(x,y+1)}\\
d_1 &= P_{(x,y+2)} - P_{(x,y+1)}\\
d_2 &= P_{(x+1,y)} - P_{(x,y+1)}\\
d_3 &= P_{(x+1,y+1)} - P_{(x,y+1)}\\
d_4 &= P_{(x+1,y+2)} - P_{(x,y+1)}
\end{aligned}
$$

4. Use | d_i | where i = 0,1,2,3,4 to locate suitable range R_k in the designed range table. Use this range to calculate number of bits, t_i that is hidden in each pair P_i. Then calculate the average bits using the equation (6).

$$\text{avg} = (\text{floor}\left(\sum_{i=1}^{5} t_i\right))/5 \qquad (6)$$

5. Calculate the revised difference | $d1'_i$ | where i = 0,1,2,3,4 as $d1'_i$ = remainder($d_i/ 2^{avg}$).
6. Use | $d1'_i$| where i = 0,1,2,3,4 to locate suitable R_k in the designed range table.
7. After R_k is located, l_k is subtracted from | $d1'_i$ | and b'_i is obtained in decimal form. A binary sequence is generated from b'_i with t_i bits where t_i = | $\log_2 w_k$ |. This binary sequence is in gray code form. Convert the gray code sequence into its binary code equivalent sequence to get the original data.
8. Repeat steps 2 through 7 till embedded message is extracted.

IV. RESULTS

The purpose of the algorithm is to increase the security of embedded message with acceptable visual quality of stego-images. The algorithm calculates average of the number of bits that can be hidden in each pair in the block and limit the numbers of bits to be hidden in each pixel pair, to the calculated average value. This reduces the hiding capacity but increases the quality of the stego image. Thus sacrifying a small amount of hiding capacity, the secret data can be securely hidden.

The text files of various sizes are used for experimentation. The text file is used as secret message, which is to be hidden in the cover image. The experimentation is carried out with various cover images in TIFF format.

In table 1, the proposed method is compared with PVD [6] and TPVD method [9][10]. It is observed that the proposed method provides better hiding capacity with improved PSNR.

Figure 2 shows the cover image and the corresponding stego images obtained using the proposed method. The cover and the stego images are indistinguishable.

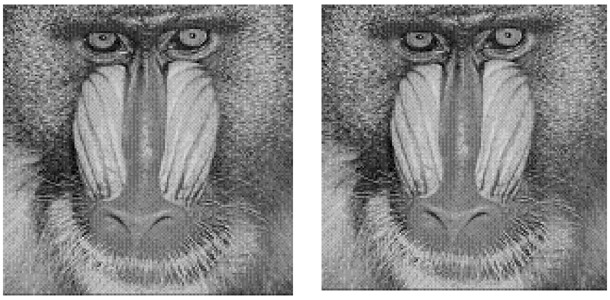
Figure 3 shows the histogram of the cover and stego image obtained using the proposed method and using full capacity of data hiding. It can be observed that the shape of the histogram is preserved after embedding the secret data.

Figure 4 shows the histogram difference. From the figure, it is obvious that, bins close to zero, are more in number and the bins, which are away from zero, are less in number. This confirms the quality of stego-image.

TABLE 1. COMPARISON OF HIDING CAPACITY IN BYTES

Cover image	PVD Method		TPVD Method		Proposed Method	
	Capa city	PSNR	Capa city	PSNR	Capa city	PSNR
Lena	50960	41.79	75836	38.89	81305	42.86
Baboon	56291	37.90	82407	33.93	81766	41.99
Peppers	50685	41.73	75579	38.50	81326	42.80

a) Cover Image-lenna.tiff b) Stego Image- lenna.tiff



c) Cover Image- baboon.tiff d) Stego Image- baboon.tiff

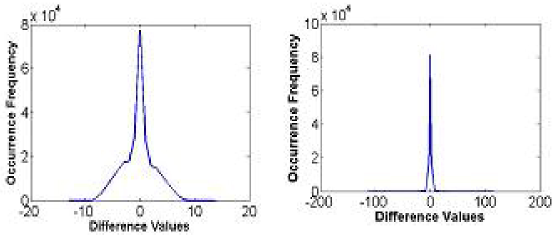
Figure 2. Cover and stego images using the proposed method

Histogram of cover image is represented as $[h_0,h_1,\ldots,h_{255}]$ whereas histogram of stego-image is represented as $[h'_0,h'_1,\ldots,h'_{255}]$.

The change in histogram [15] can be measured by (7)

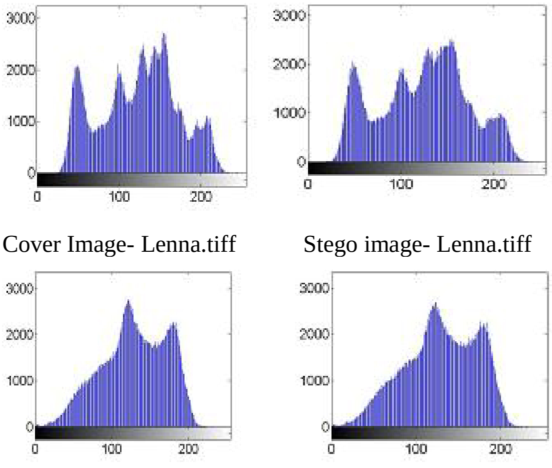
$$D_h = \sum_{m=1}^{255} |h'_m - h_m| \qquad (7)$$

Figure 5 is drawn for lena.tiff which compares the value of Dh of the 3 bit LSB replacement method and the proposed method. It can be observed that the change in histogram difference is very small for the proposed method even if almost full capacity of cover image is utilized for embedding secret data.



Lenna.Tiff Baboon.Tiff

Figure 4. Histogram difference



Cover Image- Lenna.tiff Stego image- Lenna.tiff

Cover Image- Baboon.tiff Stego image- Baboon.tiff

Figure 3. Histograms of Cover and stego images using proposed method

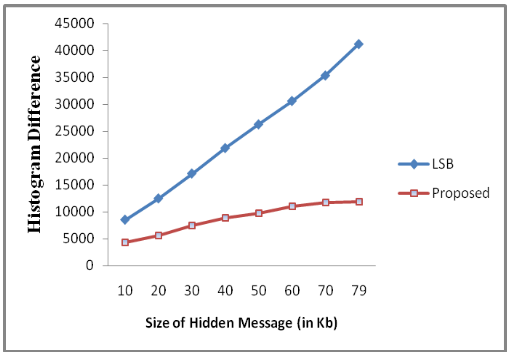


Figure 5. Histograms comparision of 3 Bit LSB substitution and proposed method

The output images are tested under the RS steganalysis. It is observed from figure 6 that the difference between RB_{MB} and RB_{-MB}, SB_{MB} and SB_{-MB} is very small. The rule $RB_{MB} \cong RB_{-MB}$ and $SB_{MB} \cong SB_{-MB}$ is satisfied for the output images [16]. So the proposed method is secure against RS attack.

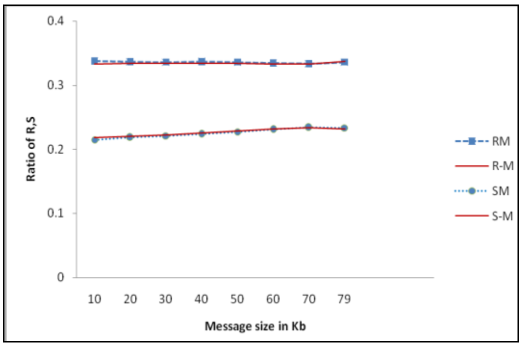


Figure 6. RS Diagram

The PSNR and MSE are utilized to evaluate the quality of the stego image. The simplest and most widely used full-reference quality metric is the mean squared error (MSE), computed by averaging the squared intensity differences of distorted (stego) and reference (cover) image pixels, along with the related quantity of peak signal-to-noise ratio (PSNR) [17]. The higher the PSNR, the better is the quality of the reconstructed image.

Given a grayscale image as the cover image to hide secret data in, it is hard for any human being to perceive any difference between the cover image and the stego-image if the PSNR value of the stego-image goes beyond 36 dB [18]. PSNR is given by the equation (8)

$$\mathrm{PSNR} = 10 \log_{10}\left(\frac{\mathrm{R}^2}{\mathrm{MSE}}\right) \tag{8}$$

where R is 255 for grey scale images.
Mean square error (MSE) is given by (9)

$$MSE = \frac{\sum_{M,N} [I_1(m,n) - I_2(m,n)]^2}{M*N} \tag{9}$$

where I_1 and I_2 represents cover image and stego image respectively.

The hiding capacity (H.C.) is calculated for each image as a percentage of the cover image size. The algorithm gives hiding capacity about 30% of the cover image size. The quality of the stego image is analyzed using PSNR, MSE and universal quality index [19].

Table 2 shows the PSNR values, MSE and universal quality index for different images obtained using proposed method. The PSNR values are above the threshold of 36 dB even after utilising more than 95 % of the hiding capacity. Also universal quality index (Q) values are close to 1, which proves that the stego images are visually indistinguishable from original cover images. The results show that the algorithm provides increased data hiding capacity and maintains the imperceptible quality of the stego image simultaneously.

TABLE 2. RESULTS OF THE PROPOSED METHOD

Cover Image	Capacity (Kb)	Size of message file (Kb)	PSNR	MSE	Q
Resolution of cover image – 256 x 256					
Baboon	19.93	19	42.14	3.970	0.974
Lena	19.98	19	42.33	3.799	0.932
Resolution of cover image – 512 x 512					
Elaine	79.69	76	42.42	3.723	0.923
Baboon	80.19	76	41.91	4.185	0.975
Lena	79.71	76	42.65	3.524	0.851
Tank	79.68	76	42.75	3.450	0.931
Peppers	79.73	76	42.65	3.530	0.873
Barbara	80.09	76	42.29	3.837	0.895
Boat	79.78	76	42.31	3.817	0.918
Resolution of cover image – 1024 x 1024					
Wall	319.69	304	42.24	3.876	0.952
Grass	319.68	304	42.10	4.005	0.987

V. CONCLUSION

In steganography, image quality and security are two important factors. Quality implies that stego image should not be visually distinguished from the original cover image, while security implies that the message should be undetectable and no one other than the eligible person should be able to extract the secret message. There is always a trade-off between higher data hiding capacity and degree of perceptibility. In this paper, an algorithm is proposed to embed secret data into still images using five pixel pair differencing method. The proposed method not only hides much larger information more securely, but also maintains a good visual quality of stego-image.

Instead of converting the binary equivalent of the whole secret message into gray code at a time, the part of the binary equivalent of the secret message that is hidden in each pair is converted in the grey code form. So even if the steganography system fails and theembedded data is extracted, it will be hard to convert it from gray code form to binary code equivalent.

Also the algorithm revises the original difference between two pixels in the pair and this revised difference is used for hiding the data in that pair. This will make estimation of exact number of bits hidden in the pair difficult. Image steganography techniques hiding textual data requires 100% accuracy for successful retrieval of hidden data from stego image. If the steganography method fails, correct estimation of number of bits hidden in the pair will be a challenge.

The algorithm produces better PSNR values (minimum MSE). This demonstrates that better quality stego images are produced even after utilizing 95% of the data hiding capacity. The experimentation is carried out with the range widths of 8,8,16,32,64 and 128 which partitions the total range [0-255] into a range table with ranges [0-7], [8-15], [16-31], [32-63], [64-127], [128-255]. The PSNR can be further improved by using the range widths 4,4,8,16,32,64 and 128 with ranges [0-3], [4-7], [8-15], [16-31], [32-63], [64-127], [128-255]. Although use of this range table improves PSNR, it reduces the hiding capacity.

Unlike the PVD[6] and TPVD [9][10] method, where the hiding capacity and PSNR values are dependent on the cover image, the algorithm provides favorable consistency in terms of uniform hiding capacity (about 79 Kb) and PSNR (about 42) for the image of size 512 x 512. Also as compared to PVD and TPVD method, the algorithm provides two extra levels of security for the secret information hidden in the cover image.

The secret data hidden in the stego image can be extracted correctly without the participation of original cover images.

REFERENCES

[1] W Bender, D Gruhl, N Morimoto, A Lu. Techniques for data hiding. IBM Systems Journal, 1996, 35 (3–4), 313–336.

[2] Bhattacharyya, Das, Bandyopadhyay, and Kim. Text Steganography: A Novel Approach. International Journal of Advanced Science and Technology, 2009, 3, 79-86.

[3] Sutaone, Khandare. Image based steganography using LSB insertion technique. IET international conference on Wireless, Mobile and Multimedia Networks. 2008, 146-151.

[4] Tiwari, Shandilya. Evaluation of Various LSB based Methods of Image Steganography on GIF File Format. International Journal of Computer Applications, 2010, 6(2), 1-4.

[5] Lee, Chen. High Capacity Image Steganographic Model. IEE proceeings, Vision, Image Signal Processing, 2000, 147(3), 288-294.

[6] D C Wu, Wen-Hsiang Tsai. A steganographic method for images by pixel-value differencing. Pattern Recognition Letters, 2003, 24, 1613–1626.

[7] Nazanin Zaker, Ali Hamzeh, Seraj Dean Katebi, Shadrokh Samavi. Improving Security of Pixel Value Differencing Steganographic Method. IEEE international conference on New Technologies, Mobilty and Security (NTMS), Cairo, December 20-23, 2009, 1-4.

[8] H C Wu, N I Wu, Tsai, Hwang. Image steganographic scheme based on pixel-value differencing and LSB replacement methods. IEE Proceedings on Vision, Image and Signal Processing, 2005, 152(5), 611-615.

[9] Ko Chin Chang, Ping S Huang, T M Tu, Chien Ping Chang. Image Steganographic Scheme Using Tri-way Pixel-Value Differencing and Adaptive Rules. IEEE international conference on Intelligent Information Hiding and Multimedia Signal Processing, Kaohsiung, Taiwan, December 26-28, 2007, 449-452.

[10] Ko Chin Chang, Chien Ping Chang, Huang, Tu. A Novel Image Steganographic Method Using Tri-way Pixel-Value Differencing. Journal of Multimedia, 2008, 3(2), 37-44.

[11] Asmari, Ghamdi. High Capacity Data Hiding Using Semi-Hexagonal Pixels Value Difference. International Conference on High Performance Computing, Networking and Communication Systems (HPCNCS-09), Orlando, Florida, USA, 2009, 14-17.

[12] V S Phad, R S Bhosale, A R Panhalkar. A Novel Security Scheme for Secret Data using Cryptography and Steganography. International Journal of Computer Network and Information Security, 2012, 2, 36-42.

[13] Khalil Challita, Hikmat Farhat. Combining Steganography and Cryptography: New Directions. International Journal on New Computer Architectures and Their Applications (IJNCAA), 2011, 1, 199-208.

[14] Dipti Kapoor Sarmah, Neha Bajpai. Proposed System for Data Hiding Using Cryptography and Steganography. International Journal of Computer Applications, 2010, 8(9), 7-10.

[15] Xin peng Zhang, Shuozhong Wang. Efficient data hiding with histogram-preserving property. Telecommunication Systems, 2012, 49(2), 179-185.

[16] J Fridrich, M Goljan, R Du. Detecting LSB steganography in color, and gray-scale images. IEEE Multimedia Magazine, 2001, 8(4), 22–28.

[17] Wang, Bovik, Sheikh, Simoncelli. Image quality assessment: From error visibility to structural similarity. IEEE Transactios on Image Processing, 2004, 13(4), 600-612.

[18] Wu, Hwang. Data Hiding: Current Status and Key Issues. International Journal of Network Security, 2007, 4(1), 1-9.

[19] Z Wang, A C Bovik. Universal Image Quality Index. IEEE Signal Processing letters, 2002, 9(3), 81-84.

Gulve Avinash K., male, Associate Professor. His areas of interest are steganography and image processing.

Joshi Madhuri S., female, has completed her BE from College of Engineering, Pune (1985), M.Tech. (CS) (1993) from IIT, Madras and Ph.D. from SRT University, Maharashtra, India. She has published 24 research papers in various international journals, international and national conferences. Her areas of interest are data mining and pattern recognition.

How to cite this paper: Gulve Avinash K, Joshi Madhuri S,"An Image Steganography Algorithm with Five Pixel Pair Differencing and Gray Code Conversion", IJIGSP, vol.6, no.3, pp.12-20, 2014.DOI: 10.5815/ijigsp.2014.03.02