Colour Image Steganography Based on Pixel Value Differencing in Spatial Domain

J. K. Mandal and Debashis Das

Department of Computer Science and Engineering,
University of Kalyani, Kalyani,
West Bengal, India
E-mail: jkm.cse@gmail.com , debashisitnsec@gmail.com

ABSTRACT

In a color image every pixel value composed of red, green and blue component and each of which ranges from 0 to 255 in case of 8-bit representation. In this paper, we have used pixel value differencing (PVD) method for secret data embedding in each of the component of a pixel in a color image. But when we use pixel-value differencing (PVD) method as image steganographic scheme, the pixel values in the stego-image may exceed the range 0~255. We have eliminated this overflow problem of each component pixel. Furthermore for providing more security, we have used different number of bits in different pixel components. It would be very difficult to trace how many bits are embedded in a pixel of the stego image. From the experiments it is seen that the results obtained in proposed method provides better visual quality of stego-image compared to the PVD method.

Keywords

Steganography, Pixel-value differencing, Pixel component, Stego-image

1. INTRODUCTION

Security measures have become very necessary issue in the age of digital transmission of information via Internet. Two schemes are used to protect secret messages from being captured during transmission. One is encryption where the secret information is encoded in another form by using a secret key before sending, which can only be decoded with secret keys. The most popular encryption techniques   are DES, RSA etc. Other way is steganography which is a technique of hiding secret information into a cover media or carrier. If the cover media is a digital image, it is called cover image and the cover image with hidden data is called stego-image. Steganographic technique can be used in military, commercial, anti-criminal and so on. There are various steganographic techniques available where a digital image is used as a carrier. The most common and simplest method is least-significant-bit (LSB) substitution, where the LSB position of each pixel of the cover image is replaced by one bit of secret data. Wang et al.[7] proposed a method to embed data by using genetic algorithm to improve the quality of the stego-image. However, genetic algorithm takes more computational time. Chang et al. Proposed[9] an efficient dynamic programming strategy to reduce the computational time. Chan and Cheng [11] proposed to embed data by simple LSB substitution with an optimal pixel adjustment process. Wu and Tsai[1] proposed a new scheme to hide more data with outstanding quality of stego-image pixel-

value-differencing (PVD) method. Thereafter, based on PVD method various approaches have been proposed [3,4,5,8,10]. In this paper, a steganographic approach on color images, using PVD has been proposed. The colour pixel-components may exceed the range 0~255 in the stego image when applying PVD method. We also have removed this drawback of PVD method here. In the proposed method a digital colour image has been used as a cover image. It will provide more security in data hiding and also better stego image quality than Wu-Tsai's PVD method.

2. REVIEW OF PVD METHOD

In PVD method[2], gray scale image is used as a cover image with a long bit-stream as the secret data. At first the cover image is partitioned into non-overlapping blocks of two consecutive pixels, p_i and p_{i+1}. From each block the difference value d_i is calculated by subtracting p_i from p_{i+1}. The set of all difference values may range from -255 to 255. Therefore, $|d_i|$ ranges from 0 to 255. The blocks with small difference value locates in smooth area where block with large difference values are the sharp edged area. According to the properties of human vision, eyes can tolerate more changes in sharp-edge area than smooth area. So, more data can be embedded into edge area than smooth areas. Therefore, in PVD method a range table has been designed with n contiguous ranges R_k (where k=1,2,…,n) where the range is 0 to 255. The lower and the upper bound are denoted as l_k and u_k respectively, then $R_k \in [l_k, u_k]$. The width of R_k is calculated as $w_k = u_k - l_k + 1$. w_k decides how many bits can be hidden in a pixel block. For security purpose R_k is kept as a variable, as a result, original range table is required to extract the embedded data. The embedding algorithm is given as algorithm 1

Algorithm 1:

1. Calculate the difference value d_i of two consecutive pixels p_i and p_{i+1} for each block in the cover image. This is given by $d_i = |p_{i+1} - p_i|$.
2. Compute the optimal range where the difference lies in the range table by using d_i. This is calculated as $R_i = \min(u_k - d_i)$, where $u_k \geq d_i$ for all $1 \leq k \leq n$
3. Compute the number of bits 't' to be hidden in a pixel block can be defined as $t = \lfloor \log_2 w_i \rfloor$. where w_i is the width of the range in which the pixel difference d_i is belonging
4. Read t bits from binary secret data and convert it into its corresponding decimal value b. For instance if t=010, then b=2
5. Calculate the new difference value d_i' which is given by $d_i' = l_i + b$
6. Modify the values of p_i and p_{i+1} by the following formula:
   $(p_i', p_{i+1}') = (p_i + \lceil m/2 \rceil, p_{i+1} - \lfloor m/2 \rfloor)$, if $p_i \geq p_{i+1}$ and $d_i' > d_i$
   $\quad\quad\quad (p_i - \lfloor m/2 \rfloor, p_{i+1} + \lceil m/2 \rceil)$, if $p_i < p_{i+1}$ and $d_i' > d_i$
   $\quad\quad\quad (p_i - \lceil m/2 \rceil, p_{i+1} + \lfloor m/2 \rfloor)$, if $p_i \geq p_{i+1}$ and $d_i' \leq d_i$
   $\quad\quad\quad (p_i + \lceil m/2 \rceil, p_{i+1} - \lfloor m/2 \rfloor)$, if $p_i < p_{i+1}$ and $d_i' \leq d_i$

where $m = |d_i' - d_i|$. Now we obtain the pixel pair (p_i', p_{i+1}') after embedding the secret data into pixel pair (p_i, p_{i+1}). Repeat step 1-6 until all secret data are embedded into the cover image. Hence we get the stego-image.

When extracting the hidden information from the stego-image, original range table is required. At first partition the stego-image into pixel blocks, containing two consecutive non-overlapping pixels each. Calculate the difference value for each block as $d_i' = |p_i' - p_{i+1}'|$. Then find the optimum range R_i of d_i'. Then b' is obtained by subtracting l_i from d_i'. Convert b' into its corresponding binary of 't' bits, where $t = \lfloor \log_2 w_i \rfloor$. These t bits are the hidden secret data obtained from the pixel block (p_i', p_{i+1}').

3. PROPOSED METHOD

Every pixel in a colour image composed of three colours (channels) i.e. Red, Green and Blue. So, every pixel contains 24 bits (for 8-bit representation) where 8 bits for red component, 8 bits for green and 8 bits for blue component in a pixel, shown in Fig.1. In the proposed technique, all the three components have been used for data embedding. First, we have separated each colour component from a pixel and get three separate M*N matrix, one for each component colour, where the original image size is M*N. Now, apply pixel value differencing method for data hiding in each matrix separately, but in a sequencing manner. First embed bits in 1st pixel block (two consecutive non-overlapping pixels) of the red component matrix, then in 1st block of green component matrix and lastly in blue component matrix, then again 2nd block of red matrix and so on. In this way secret data is embedded into the total image. Further we have embedded different number of bits for different component pixel blocks for increasing security as well as improving the visual quality of the stego-image, shown in Fig.2.
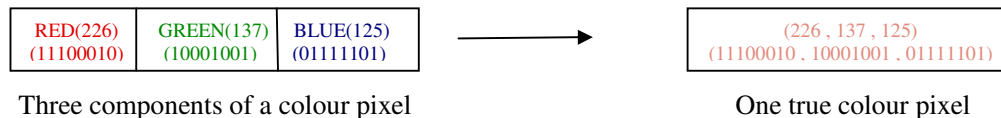
RED(226)	GREEN(137)	BLUE(125)
(11100010)	(10001001)	(01111101)

(226 , 137 , 125)
(11100010 , 10001001 , 01111101)

Three components of a colour pixel   One true colour pixel

Fig.1 *Components of a colour pixel*

0~7	8~15	16~31	32~63	64~127	128~255

Difference Range Table

G(max 3bits)

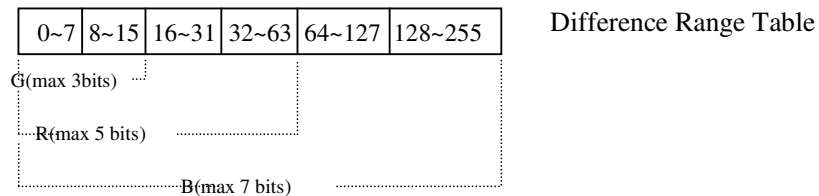R(max 5 bits)

B(max 7 bits)

Fig.2 *Number of bits can be hidden in various components*

In addition, it is seen that pixel values in the stego image may exceed the range 0~255 on applying PVD which is not desirable as it may lead to improper visualisation of the stego image. In this section we introduce a technique to overcome this problem. In the proposed method we have used the original PVD method to embed secret data. If any pixel value exceeds the range (0 to 255), then check the bit-stream 't' to be hidden. If MSB(most significant bit) of the selected bit stream 't' is 1 then we embed one less number of bits, where MSB position is discarded from t; otherwise the bit number of hidden data depends on w_i . For instance, if pixel value exceeds the range and selected bit-stream t=101, then set t=01 and embed it. If it is seen that the pixel value again exceeding range, then embed the value at one pixel, rather than both pixels(of the pixel block), which will not exceed the range after embedding; where the other pixel is kept unchanged. It will keep the pixel values within the range because both pixels of a block cannot exceed at the same time as per the PVD method by Wu and Tsai. Keep the information within each block, whether one less bit is embedded or not, as overhead. The embedding algorithm is presented in section 3.1. Fig.3 shows the block diagram of the embedding algorithm.

3.1. Embedding Algorithm

Step 1: Separate the colour image into three component colour matrix and apply the following steps on each of them sequentially i.e. apply following steps on 1st pixel block of RED

matrix, then 1st block of GREEN matrix and finally on BLUE matrix, then started from 2nd block of each matrix and so on.

Step 2: Calculate the difference value di for each block of two consecutive non-overlapping pixels p_i and p_{i+1} , is given by $d_i=|p_i-p_{i+1}|$.

Step 3: Find optimal range R_i for the d_i such that $R_i =min(u_i-d_i)$, where $u_i \geq d_i$ .Then $R_i \in [l_i , u_i]$ is the optimum range where the difference lies.

Step 4: Compute the amount of secret data bits t from the width w_i of the optimum range, can be defined as $t= \lfloor log2 \ w_i \rfloor$.

Step 5: For RED matrix,   if $t \leq 5$ then execute the next steps; otherwise no embedding in the block
For GREEN matrix, if $t \leq 3$ then execute the next steps; otherwise no embedding
For Blue matrix execute the following steps.

Step 6: Read t bits and convert it into a decimal value b. Then calculate the new difference value by the formula $d_i'=l_i+b$.

Step 7: Now, calculate the pixel values after embedding t bits (p_i' , p_{i+1}') by original PVD method.

Step 8: Check the embedded pixel values whether it exceed the gray-level range or not . If it exceeds then check the embedded bit-stream t . Otherwise go to step 9.

Step 8.1: If the left most position of the bit-stream is 1 then select 't' by discarding one bit from its   left most position. Convert 't' bits into its corresponding decimal value b and find new difference value as $d_i'=l_i+b$. Otherwise do not discard any bit from 't' and calculate d_i' .

Step 8.2: Calculate new pixel values (p_i' , p_{i+1}') using original PVD method and check again if it is in the gray range. If it is in the range then go to step 9.Otherwise do the following :

$(p_i' , p_{i+1}')=(p_i-m , p_{i+1})$,   if $p_{i+1} \geq p_i$ and p_{i+1} crossing the upper range(i.e 255) ;
$(p_i' , p_{i+1}')=(p_i , p_{i+1}-m)$,   if $p_{i+1}<p_i$ and p_i crossing the upper range(i.e 255) ;
$(p_i' , p_{i+1}')=(p_i , p_{i+1}+m)$,   if $p_{i+1} \geq p_i$ and p_i crossing the lower range(i.e 0);
$(p_i' , p_{i+1}')=(p_i+m , p_{i+1})$,   if $p_{i+1}<p_i$ and p_{i+1} crossing the lower range(i.e 0) .

where $m=|d_i'-d_i|$.

Step 9: Now, the pixel block (p_i , p_{i+1}) is replaced by (p_i' , p_{i+1}').

Step 10: To keep the information whether 't' bits or 't-1' bits has been embedded, do the following for each modified block :

Step 10.1:

If no bit has been discarded then do the following:

LSB of P'$_i$ LSB of P'$_{i+1}$ then do

i)   0   0 P'_{i+1} +1

ii)   0   1

a) P'_{i+1} <255 and $P'_I \geq 0$ P'_{i+1} +1

b) P'_i >0 and P'_{i+1} =255 P'_i-2 and P'_{i+1} -1

c) P'_i=0 , P'_{i+1}=255 P'_{i+1}

iii)   1   0 P'_i -1

iv)   1   1 P'_i -1

Step 10.2:

One bit has been discarded

LSB of P'_i LSB of P'_{i+1} then do

i)   0   0 P'_i +1

ii)   0   1 P'_i +1

iii)   1   0

a) P'_{i+1} >0 and $P'_i \leq 255$ P'_{i+1} -1

b) P'_i <255 and P'_{i+1} =0 P'_i+2 and P'_{i+1}+1

iv)   1   1 P'_{i+1} -1

Step 11: Now we get the stego blocks and hence the stego image.

3.2. Extraction Algorithm

Figure 3 shows the block diagram of the extraction algorithm. The steps used for extracting the hidden data are as follows :

Step 1: Divide the stego image into three component matrix RED, GREEN and BLUE and execute the   following steps for each pixel block, consist of two consecutive non-overlapping pixels, of RED, GREEN, BLUE respectively i.e. extract bits from one stego pixel at a time.

Step 2: Check the LSB positions of Pi (first pixel of the block) for each block and do the following :

LSB of P_i Convert it as

0   P_i +1

1   P_i -1

Step 3: Now calculate the difference value di of two consecutive pixels of each block by using the formula $d_i = |p_i - p_{i+1}|$.

Step 4: Find the appropriate range R_i for the difference d_i.

Step 5: Calculate the number of bits hidden into each block by the formula $t = \lfloor \log 2\ w_i \rfloor$, wi is the width of the optimal range R_i .

Step 6: If operating on RED block and 't' $\leq$ 3 then execute following steps; otherwise no extraction
        If operating on GREEN block and 't' $\leq$ 5 then execute following steps; otherwise no extraction
        If operating on BLUE block and 't' $\leq$ 7 then execute the following steps

Step 7: Extract 't' bits, by the extracting method of original PVD.

Step 8: Check the LSB of P_i . If it is 1, replace the MSB position of extracted 't' bits with '1'. Otherwise do nothing.
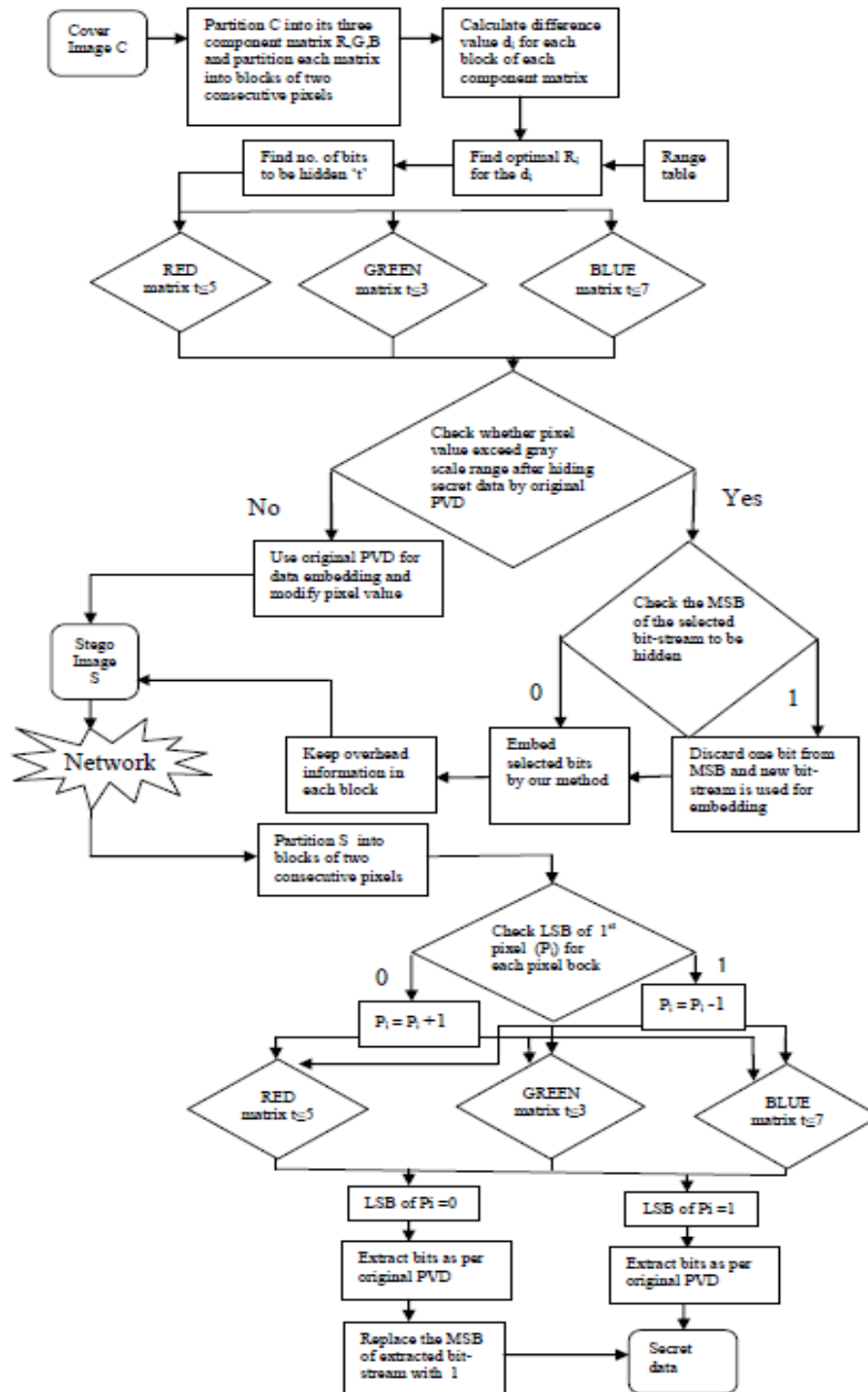
Fig. 3 Block diagram of embedding and extraction algorithm

4. EXPERIMENTAL RESULTS

The proposed scheme hides secret data into a colour image using pixel value differencing method but various bits in different channels or colour components. Also the problem of overshooting 0~255 range for each component pixels, occurs in PVD, has been removed here. This scheme results better visual quality of the stego image. To prove the proposal practically we have used C programming language to implement our scheme and Wu-Tsai's PVD scheme. We have also shown the management of pixel value range for a gray scale cover image, where the capacity remains same with original PVD method. The range table width used here are w_i ={ 8, 8, 16, 32, 64, 128 }.We have used cover images of size 512*512 and hide a large bit stream as the secret information. We have used the Peak-Signal-to-Noise ratio (PSNR) to evaluate the quality of stego-image. The experimental results are shown in Fig. 4 for colour image and Fig. 5 showing the results of gray images. A table of results for colour images is given in Table 1 depicting the capacity of hidden message and the PSNR values. The comparison, of message payload and PSNR value, between proposed method (in both gray image and colour image) and original PVD method is shown in Table 2. The PSNR value and payload capacity of each stego-image is given as average value by executing 100 rounds using standard digital images where the hidden information are different.

5. ANALYSES AND DISCUSSION

From the experimental results, using colour image as cover media, we can see that PSNR values are increasing for every stego image compared to original PVD method. In a colour image, the contribution of the component colours (red, green or blue) of a pixel is different. Green component contributes 59% where red component contributes 30% and blue component provide 11% contribution to make a colour pixel. So, we have embedded more bits to blue channels and lower bits to the green and red components. This results less distortion of the pixels in stego image. Also it provides more security because different number of bits has been hidden in different channels of a pixel, so is hard to trace how many bits are hidden within a pixel. On the other hand, according to the proposed method, if pixel component value exceeds 0~255 ranges on embedding, one MSB bit is discarded from the selected bit-stream to be hidden. So, decimal value of the reduced bits will be half or less than half. As a result the distortion of the pixel value in the stego-image will be less. Also It should be noted that whether the pixel exceeds range or not, we have to keep the information in every component pixel block by adding or subtracting 0, 1, or 2. But this has no remarkable effect for reducing PSNR value for colour image.

In case of gray scale cover image, capacity remains same for the proposed method as in original PVD method. PSNR values have been changing between -0.62 to +0.32 dB. For gray image pixels, keeping of overhead information affects the PSNR to reduce in some cases.
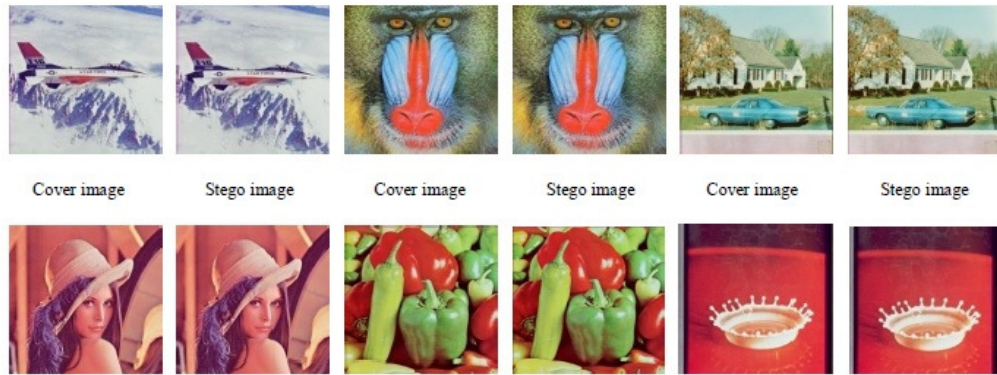
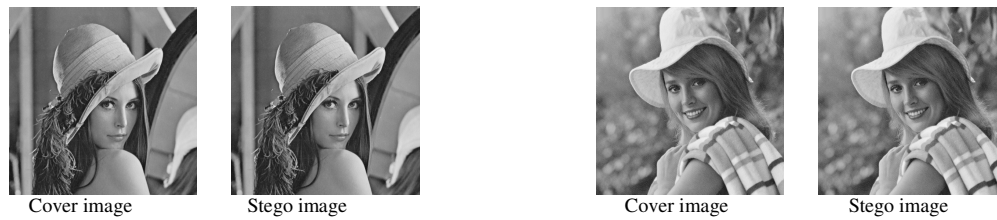Fig.4. Cover images and their corresponding stego images

Fig.5. Cover images and corresponding Stego images for gray scale image

Table 1: **Experimental results of the proposed method**

Cover image (512× 512)	Proposed method	
	Capacity	PSNR
Lena	145787	42.26
Baboon	144916	38.44
Jet	145648	42.60
Pepper	145995	42.28
Girl	144285	42.80
House	145374	41.41
Splash	146732	42.86
Sailboat	143278	40.66

Capacity in Bytes and PSNR in dB and Cover images of size 512*512

Table 2: **Comparison of results of the proposed and Wu and Tsai's method**

Cover image (512× 512)	Wu and Tsai's method		Proposed method			
			Gray Image		Colour Image	
	Payload	PSNR	Payload	PSNR	Payload	PSNR
Lena	1.56	41.70	1.56	40.61	1.48	42.26
Baboon	1.75	36.86	1.75	36.67	1.47	38.44
Jet	1.55	41.31	1.55	40.94	1.48	42.60
Pepper	1.55	40.55	1.55	40.61	1.48	42.28

Payload in bpB and PSNR in dB and Cover images of size 512*512 .

6. CONCLUSIONS

In this paper, we have discussed a steganographic method for data hiding by using pixel value differencing in colour images which also guarantees that no pixel value will exceed the range 0 to 255 in stego-image. We have shown the pixel range overflow management also for gray scale images.   We have used original PVD method where pixel value does not cross the range, elsewhere proposed method has been used for embedding data. Proposed scheme on colour images gives more security than the original PVD used in grey images and also provides better visual quality of stego-image. Furthermore, proposed method extracts the hidden secret message efficiently without using the original cover image.

ACKNOWLEDGEMENTS

REFERENCES

[1]   J.K. Mandal, Debashis Das "Steganography Using Adaptive Pixel Value Differencing(APVD) for Gray Images through Exclusion of Underflow/Overflow ", Computer Science & Information Series, ISBN : 978-1-921987-03-8, pp. 93-102, 2012.

[2]   H.C. Wu, N.I. Wu, C.S. Tsai and M.S. Hwang, "Image steganographic scheme based on pixel value differencing and LSB replacement method", IEEE Proceedings on Vision, Image and Signal processing, Vol. 152, No. 5,pp. 611-615, 2005.

[3]   Schneier B.:'Applied cryptography'(John Wiley & Sons, New York, 1996, 2nd Edn.)

[4]   W. bender, D. gruhl, N. Morimoto, A.Lu, "Techniques for data hiding", IBM Systems Journal Vol. 35(3-4),pp. 313-336, 1996.

[5]   F.A.P peticolas, R.J Anderson and M.G. Kuhn, "Information Hiding – a Survey" proceedings of the IEEE,VOL. 87,PP. 1062-1078, 1999.

[6]   Y.K. Lee, L.H. Chen, "High capacity image steganographic model", IEEE Proceedings on Vision, Image and Signal processing, Vol. 147, No.3,pp. 288-294, 2000.

[7]   R.Z. Wang, C.F. Lin, J.C. Lin, "Image hiding by optimal LSB substitution and genetic algorithm", Pattern Recognition Vol. 34, pp. 671-683, 2001.

[8]   Tseng, Y.C. , Chen Y.Y. Pan H.K.:'A secure data hiding scheme for binary images', IEEE Trans. Commun. , 2002, 50,pp. 1227-1231

[9]   C.C. Chang, J.Y. Hsiao, C.S. Chan, "Finding optimal least-significant -bit substitution in image hiding by dynamic programming strategy", Pattern Recognition Vol. 36, Issue 7,pp. 1583-1595, 2003.

[10] D.C. Wu, and W.H. Tsai, "A Steganographic method for images by pixel-value differencing", Pattern Recognition Letters, Vol. 24, pp. 1613-1626, 2003.

[11] C.K. Chan, L.M. Cheng," Hiding data in images by simple LSB substitution" ,pattern recognition vol. 37, Issue 3,pp. 469-474, 2004.

[12] C.M. Wang, Nan-I Wu, C.S. Tsai, M.S. Hwang, "A high quality Steganographic method with pixel value differencing and modulus function", The journal of Systems and software, 2007.

[13] Amanpreet Kaur, Renu Dhir and Geeta Sikka, "A New Image Steganography Based on First Component Alteration Technique", IJCSIS, Vol. 3, No. 6, 2009.

Authors

Jyotsna Kumar Mandal, M.Tech(Computer Science, University of Calcutta), Ph.D.(Engg. , Jadavpur University) in the field of Data Compression and Error Correction Techniques, Professor in Computer Science and Engineering, University of Kalyani, India. Life member of Computer Society of India since 1992 and life member of Cryptology Research Society of India. Dean Faculty of Engineering, Technology and Management, working in the field of Network Security, Steganography, Remote Sensing & GIS Application, Image Processing. 25 years of teaching and research experiences. Eight scholars awarded Ph.D., one submitted and eight are pursuing. Total number of publications is more than two hundred and thirty in addition of publication of five books from LAP, Germany .

Debashis Das pursuing his M. Tech. in Computer Science and Engineering from University of Kalyani, under the Department of Computer Science and Engineering. Received his B. Tech degree in Information Technology in 2009. He has two public ations in the in ternational conference proceedings.