I.J. Image, Graphics and Signal Processing, **2013, 6, 25-32**
Published Online May 2013 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijigsp.2013.06.04

A New Steganography Technique Using Snake Scan Ordering Strategy

Rajeev Kumar, Khushil K. Saini, and Satish Chand
Division of Computer Engineering, Netaji Subhas Institute of Technology, Delhi, India-110078
Rajeev.garg.it@gmail.com, khushil@rediffmail.com, schand86@hotmail.com

Abstract — In this paper, we propose a new steganography technique using the snake scan ordering strategy. The proposed method hides the secret data that is an image in another image, known as the cover image. In this method, the pixel values of the secret image are organized in snake scan order, which are preprocessed to reduce their size. The resultant data is embedded into the Least Significant Bits (LSBs) of the pixels of the cover image. To minimize the error/distortion, the pixel values of the stegoimage are adjusted using Optimal Pixel Adjustment Process (OPAP). The performance of the proposed method is compared with that of the simple LSB substitution method, Chang et al. method, Thein & Lin method, and Chen method in terms of Peak Signal to Noise Ratio (PSNR). Our proposed method has higher PSNR in almost all cases.

Index Terms — Steganography, cover image, stego image, snake scan ordering strategy, PSNR

I. Introduction

Due to advancements in computing and communication technologies, a communication can be established between two or more persons being at different places through the computer networks or using Internet in case they are at distance, which requires large resources in terms of money and time to move physically to the destination. Some messages, which are supposed to be very important and their revelation may cause unrepairable damage, cannot be transmitted in their original form. These types of messages are known as confidential message and they must be communicated in a very secure way. One of the possible ways is to encrypt the message before sending it. The transmission of secret message may also be used authenticating the media itself. One of the important media to transmit a secret message is image data. The secret to be transmitted is called the secret data and the image data is called the cover image. If the existence of the confidential data in the cover image is detected by a masquerader, it can be misused for different purposes. Such encryption process is said to be vulnerable. The solution to this type of problem can be steganography. The steganography is the art of concealing secret data into a carrier for conveying the secret message confidentially [1], [2]. The basic model of steganography uses the cover media in which the secret data is embedded, the secret data that is to be hidden, and the algorithm with secret key through which the secret data is embedded into the cover media. The outcome of this process is stegomedia (the media that has the secret message) that is sent to the receiver. The word 'steganography' is the combination of two Greek words - *stegano* and *graphy*, which mean covered and writing, respectively, in English. It has been used practically since ancient time. The steganography has many applications in today's life such as digital watermarking for authentication purpose, or to maintain confidentiality as well as integrity of the valuable data in order to avoid access by an unauthorized user or masquerader.

In image steganography, the cover media is the cover image and the stegomedia is the stegoimage. The images in a computer system are stored as an array of pixels, which can be gray or colored images. The gray scale images, also called intensity images, have only on channel and each pixel is generally represented in terms of 8 bits. In colored images, the array is a combination of three colors: red (R), green (G) and blue (B). These are also called channels. The numeric value (which is in the range of 0-255 for digital images represented by using 8 bits per pixel) of each pixel is called the intensity of that pixel. For representing a color component by 8 bits requires 24 bits per pixel. The pixels in a digital image are widely used to hide the secret data on the internet. The images can be manipulated in either spatial domain or frequency domain and accordingly there are two main classes into which the steganography techniques may be divided.

- Transform based techniques
- Spatial domain based techniques

The transform based techniques [3,4] have been inspired by the functioning of the Human Visual System (HVS). The human eyes are more sensitive to random noise in smooth area than in the busy area. Thus, more data can be hidden in busy regions than the smoother regions because the degradation of the image quality is more noticeable in smoother regions. In a transform based technique, the cover image is transformed into frequency domain by applying a unitary transform to the original image. The unitary transform breaks the original image into different frequencies and its inverse transform (matrix) is simply transpose of the original transformation (matrix). The unitary transforms include

as Discrete Fourier Transform (DFT), Discrete Cosine Transform (DCT) or Discrete Wavelet Transform (DWT). The secret data is then embedded into the high frequency coefficients [5]. These types of methods are more secure and robust; they are however more complex and slower than the spatial domain based methods as we need to apply transform and inverse transform operation. The spatial domain based techniques [6,7] hide the secret message directly into the pixel intensity of the cover image. In one of the important type of techniques, some prespecified bits of the pixels are modified based on the secret message, which comes under the least significant bit (LSB) substitution.

In least significant bit substitution method, the secret data bits are embedded into the LSBs of the pixels of cover image. The least significant bit substitution method is the most obvious, but it is the most commonly known approach for hiding the secret information in a cover image. This method is quite simple, but increasing the size of the secret data distorts the stegoimage. In this method, every pixel value is modified in equal amount without analyzing the observation. In order to develop a good steganography method, it must have the following characteristics: good imperceptibility, sufficient data hiding capacity, and robustness [8,9]. Some researchers have discussed the suitability of the cover images and detection of secret message. Fridrich et al. [10] report that the JPEG images are a very poor choice for cover images because the quantization in JPEG introduces some changes that may serve as a "watermark". In [11], Fridrich discusses the Raw Quick Pairs (RQP) method for detection of LSB embedding in 24-bit color images. However, it is not applicable for gray scale images. Pfitzmann and Westfeld [12] discuss a method that is based on statistical analysis of Pairs of Values (PoVs), which are exchanged during message embedding. This method works very well for the known locations of the message. For randomly scattered messages, it can detect if the message length is comparable with the number of pixels in the image. Zhang et al, [13] discuss a method using the transition coefficients between difference image histograms of an image and its variant obtained by setting all bits in the LSB plane to zero. The paper [14] discusses a robust steganalytic method for detecting LSB embedding in digital images, which is based on a finite state machine model. The states are multisets of sample pairs, which are known as trace multisets. The statistical parameters of sample pairs are very much susceptible to LSB embedding even for the short messages.

In our proposed work, we use the simple LSB substitution method to hide the secret data into the cover image after exploiting redundancy in the neighboring pixels of the secret image. The proposed method performs better than that of the simple LSB substitution method [15], Chang et al. method [16], Chen method [17], and Thein & Lin method [18] in terms of peak signal to noise ratio. The rest of this paper is organized as follows. Section 2 discusses the related work. Section 3 reviews the Optimal Pixel Adjustment Process (OPAP) in LSB. The proposed method is discussed in section 4. The experimental results are given in section 5, and finally section 6 concludes the paper.

II. RELATED WORK

Wang et al. [19] have discussed a method based on exhaustive LSB substitution. This method improves the simple LSB method in terms of the image quality. It overcomes the limitation of the exhaustive LSB method by using genetic algorithm to hide the secret image into the cover image. However, it takes huge computation time and provides *approximate* optimal solution. Chang et al. [16] have discussed a method, which uses dynamic programming to get the optimal solution. This method not only provides the optimal solution, but also takes less computation time. Chang and Cheng discuss an important method known as Optimal Pixel Adjustment Process (OPAP) [15]. This method reduces the error (distortion) occurred in the pixels of the stegoimage due to the LSB substitution method. The pixel values are adjusted to reduce the distortion/error, which significantly improves the quality of the stegoimage.

Thein & Lin [18] discuss a simple method, called digit by digit data hiding method, based on the modulus function to hide the secret data into the cover image. In this method, the secret data is partitioned into non-overlapping segments and a segment is embedded directly into a pixel. To minimize the distortion, the pixel value is adjusted after embedding the segment of the secret data. This method outperforms the methods discussed in [5,19] in terms of the stegoimage quality. It is a simple method to execute and takes less time. Chen [17] discusses a method that improves the Thein & Lin [18] method in terms of the image quality. This method makes use of the repetition of the data so that the number of modified pixels is reduced without using any extra bit for representing the repetition. Thus, the quality of the stegoimage is improved significantly. In our proposed method, we try to reduce the size of the secret data and then make use of optimal pixel adjustment technique [15] to embed the secret data and reduce the distortion.

III. REVIEW OF OPTIMAL PIXEL ADJUSTMENT PROCESS (OPAP) [15]

The Optimal Pixel Adjustment Process (OPAP) method reduces the distortion caused by the simple LSB substitution method. The basic concept of the OPAP is based on the technique discussed in [20]. In this paper, the worst mean square error (WMSE) has been obtained as ½ of that of the simple LSB substitution method. This method is briefly described below. Suppose a pixel has its value as **P**, the value of the rightmost **r** LSBs is $\mathbf{P_r}$. Let **P'** be the pixel value after embedding **r** message bits using the LSB replacement method, **d** be the decimal value of these message bits and **P''** be the pixel

value after adjustment. The pixel values are adjusted in OPAP as follows:

If $P^r - d > 2^{r-1}$ *and* $P' + 2^r \leq 255$

$P'' = P' + 2^r$

Else if $P^r - d < -2^{r-1}$ *and* $P' - 2^r \geq 0$

$P'' = P' - 2^r$

Else

$P'' = P'$

The values of the right most **r** LSBs in **P''** remain the same as in **P'.** This means that the LSBs can be directly extracted from the pixels to recover the secret data. In the next section, we discuss our proposed method.

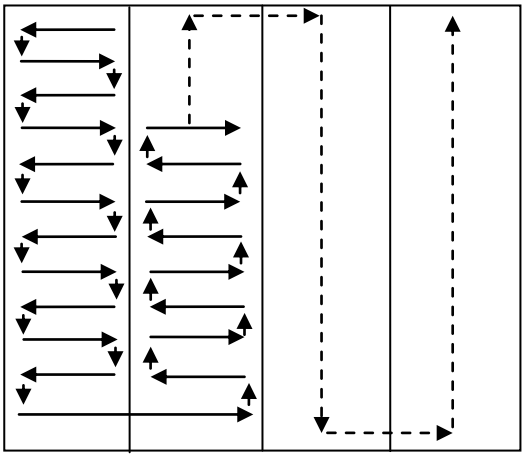


Fig.1 Snake scan ordering strategy

IV. PROPOSED METHOD

Our proposed method consists of two processes: encoding process and decoding process. The encoding process is further divided into two phases: preprocessing phase and embedding phase.

A. Encoding process

The encoding process consists of two phases. In the first phase, the secret data is arranged in snake scan ordering from the secret image and preprocessed to reduce its size, while in the second phase the resultant secret data is embedded in the cover image and the pixel adjustment is performed for obtaining the nearest stegopixel.

(i) Preprocessing phase

1. Load secret image and partition it into blocks of 16*h, where h is the height of the secret image.
2. Extract the pixel values from the partitioned secret image in snake scan order in an array, say A, as shown in Fig. 1.
3. Maintain the elements of the array A in clusters in such a way that the number of elements in a cluster can be at most 255 and the difference value of its minimum and maximum values is not more than 63.
4. The size and minimum value of each cluster are stored in another array, say B.
5. Find the difference of each element of the cluster with its minimum value. Carry out this process for all clusters.
6. Number of clusters is represented in 16 binary bits in an array, say C.
7. Each element of B is represented in 8 binary bits and stored in an array, say D.
8. Each value obtained from step (5) is represented in 6 binary bits and maintained in an array, say E. Arrays C, D, and E are concatenated as C followed by D and D followed by E; name the concatenated array as F.

(ii) Embedding phase

1. Load the cover image and represent the pixel values in binary form and set a counter $k_1 = 0$.
2. Increment the value of k_1 by 1 and embed the elements of the array F (obtained in preprocessing phase) into k_1 LSBs of the pixels of the cover image.
3. Repeat step (2) till all the elements of the array E are not embedded.
4. The pixel values of the resultant image are adjusted using OPAP [15] that is discussed in section 3.

B. Decoding process

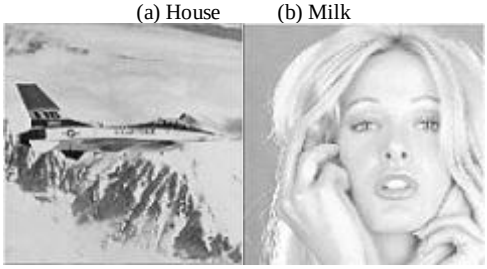
1. Set a counter $k_2 = 0$.
2. Increment k_2 by 1. Extract k_2 LSBs of all the pixels of the stegoimage.
3. The decimal value of the first 16 extracted bits gives the number of clusters.
4. Repeat step (2) till the number of bits extracted are not equal to *number of clusters*16 +16*.
5. The decimal value of the bits after 16 bits in groups of 8 bits gives the minimum value of the corresponding cluster. The number of groups will be equal to the number of clusters.
6. The decimal value of the next bits (after extraction of the minimum values) in groups of 8 bits represents the cluster size. Thus, such groups will be equal to the number of clusters.
7. If all the k_2 LSBs have been extracted from the pixels of the stegoimage, then repeat step (2); otherwise first extract k_2 LSBs of the remaining pixels and repeat step (2) until we have another N bits, where N is the sum of all the elements of the cluster size obtained in step (6).

8. The difference values are obtained by getting the decimal value of each 6 binary bits obtained from step (7).
9. Add the difference values obtained in step (8) of each element of the clusters to their minimum values obtained in step (5). Perform this process for each cluster. The resultant values thus obtained give the pixel intensity of the secret image.
10. Arrangement of these pixel values in the reverse order of Fig. 1 gives the secret image.

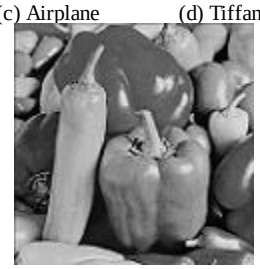
IV. EXPERIMENTAL RESULTS

In this section, we discuss the experimental results of our proposed method. The secret images we have considered in our experiments are five images: House, Milk, Airplane, Tiffany, and Pepper, each of size 256×512 pixels as shown in Figs. 2(a) – (e). The cover images taken in our experiments are two images: Lena and Baboon as shown in Figs. 3(a) & (b), respectively. We have chosen these images as these are the very commonly used images in literature by the researchers. Each cover image is of size 512×512 pixels. The images considered as the secret images are also very commonly used image in image processing community.

(a) House (b) Milk



(c) Airplane (d) Tiffany



(d) Pepper

Fig. 2 Secret images each of size 256×512 pixels.

The stegoimages after hiding the secret images into each of the cover images are shown in Figs. 4(a) - 8(b). We have used the Peak Signal to Noise Ratio (PSNR) as the quality metric to evaluate the quality of the stegoimages. The mathematical formula to calculate the PSNR is given as follows:

$$PSNR = 10*\log_{10}[(255\times255)/MSE]$$

where, the Mean Square Error (MSE) is defined as the square of the difference between the corresponding pixel values of the original image and the stegoimage after dividing it by the size of the image. The mathematical formula for computing the mean square error between C and S images of size W×H is given by

$$MSE = \frac{1}{W*H}\sum_{x=1}^{W}\sum_{y=1}^{H}[C(x,y)-S(x,y)]^2$$

C(x, y) and S(x, y) are the two images of size W×H each. W is the width and H is the height of each image. In this case, C may be considered as the original image or cover image and S as the stegoimage.

The higher PSNR value signifies lower error in the cover image, which means that the quality of the stegoimage is better. We have compared our results with some of the important methods that include simple LSB substitution method [15], Chang et al. method [16], Thein & Lin method [18], and Chen method [17].

Tables I and II show the PSNR values of simple LSB substitution method [15], Chang et al. method [16], Thein & Lin method [18], Chen method [17], and our proposed method. In Table I, the PSNR values after hiding the secret images of Fig. 2 in the cover image Lena and in Table II the PSNR values after hiding the secret images (of Fig. 2) into the cover image Baboon are given. The corresponding stegoimages are shown in Figs. 4(a)-8(b). Our proposed method utilizes the characteristics of the images, i.e. the closeness among the pixel values. This characteristics helps reducing the size of the secret data which significantly improves the quality of the stegoimage. We have extracted the pixel values according to the snake scan ordering strategy as shown in Fig 1. The pixels which are close in their locations most probably have their pixel values close to each other or may have same values. Thus, the snake scan ordering strategy helps get the pixel values, which are close in their values in contiguous locations. Hence, the size of the secret data is reduced significantly. It is evident from both the tables that our method performs better than the simple LSB substitution [15], Chang et al. [16], Thein & Lin [18] and Chen [17] methods. Our method however will not work for the secret images having random noise.

(a) Lena cover image (b) Baboon cover image

Fig. 3 Cover images of size 512×512 pixels: (a) Lena, (b) Baboon

Table I: PSNR values of simple LSB substitution [15], Chang et al. [16], Thein & Lin [18], Chen [17] and proposed methods after hiding the secret images: House, Milk, Airplane, Tiffany, and Pepper in the cover image: Lena.

Secret image 256×512	Simple LSB substitution Method [15]	Chang et al.'s Method [16]	Thein & Lin's Method [18]	Chen Method [17]	Proposed method
House	32.69	33.10	34.78	36.37	36.64
Milk	32.26	32.53	34.86	35.76	36.61
Airplane	31.95	32.89	34.76	35.73	35.05
Tiffany	31.32	32.90	34.80	35.49	35.64
Pepper	32.44	32.57	34.79	35.46	35.41

Table II: PSNR values of simple LSB substitution [15], Chang et al. [16], Thein & Lin[18], Chen [17] and proposed methods after hiding secret images: House, Milk, Airplane, Tiffany, and Pepper in the cover image: Baboon

Secret image 256×512	Simple LSB substitution Method[15]	Chang et al.'s Method [16]	Thein & Lin's Method [18]	Chen Method [17]	Proposed method
House	32.70	33.14	34.79	36.37	36.67
Milk	32.26	32.56	34.80	35.80	36.65
Airplane	32.04	32.93	34.82	35.72	35.09
Tiffany	31.40	32.93	34.82	35.52	35.66
Pepper	32.46	32.58	34.80	35.44	35.43

(a) Stegoimage Lena (b) Stegoimage Baboon

Fig. 4 Stegoimages after hiding secret image of House (Fig. 2(a))

(a) Stegoimage Lena (b) Stegoimage Baboon

Fig. 5 Stegoimages after hiding secret image of Milk (Fig. 2(b))

(a) Stegoimage Lena (b) Stegoimage Baboon

Fig. 6 Stegoimages after hiding secret image of Airplane (Fig. 2(c))

(a) Stegoimage Lena (b) Stegoimage Baboon

Fig. 7 Stegoimages after hiding secret image of Tiffany (Fig. 2(d))

(a) Stego image Lena (b) Stego image Baboon

Fig. 8 Stegoimages after hiding secret image of Pepper (Fig. 2(e))

V. Conclusions

In this paper, we have discussed a new image hiding technique using the snake scan ordering strategy. This method can hide a secret image into a cover image. The pixel values of the secret image are extracted in the snake scan order. After extraction, the clusters are constructed that help reducing the size of the secret data. We have used the LSB substitution with OPAP to hide the resultant secret data into the pixels of the cover image. Our method maintains good quality stegoimage. The limitation of our method is that it will not be applicable if the secret image contains random noise.

References

[1] J. Fridrich, Steganography in Digital Media: Principles, Algorithms, and Applications. Cambridge, U.K.: Cambridge Univ. Press, 2009.

[2] N. Provos and P. Honeyman, "Hide and seek: An introduction to steganography," *IEEE Security Privacy*, vol. 3, no. 3, pp. 32–44, May/Jun. 2003.

[3] A. Westfeld, "F5 A steganographic algorithm: High capacity despite better steganalysis," Proceedings of 4th International Information Hiding Workshop, Springer-Verlag, Vol. 2137, pp. 289-302, 2001.

[4] M. Swanson, M. Kobayashi, and A. Tewfik, "Multimedia data embedding and watermarking technologies," Proceedings of the IEEE, Vol. 86, No. 6, pp. 1064-1087, 1998.

[5] C. C. Chang, T. S. Chen, and L. Z. Chung, "A steganographic method based upon JPEG and quantization table modification", Information Sciences, vol. 4, pp. 123-138, 2002

[6] I. Cox, J. Kilian, T. Leighton, and T. Shamoon, "Secure spread spectrum watermarking for multimedia," IEEE Transactions on Image Processing, Vol. 6, No. 12, pp. 1673-1687, 1997.

[7] N. Provos, "Defending against statistical steganalysis," Proceedings of 10th Usenix Security Symposium, pp. 323-335, 2001.

[8] T. Filler, J. Judas, and J. Fridrich, "Minimizing embedding impact in steganography using trellis-coded quantization," in *Proc. SPIE, Media Forensics and Security*, 2010, vol. 7541, DOI: 10.1117/12.838002.

[9] S. Lyu and H. Farid, "Steganalysis using higher-order image statistics," *IEEE Trans. Inf. Forensics Security*, vol. 1, no. 1, pp. 111–119, Mar. 2006.

[10]J. Fridrich, M. Goljan, and R. Du, "Steganalysis based on JPEG compatibility," Proceedings of Special Digital Watermarking Data Hiding, pp. 275-280, 2001.

[11]J. Fridrich, R. Du and L. Meng, "Steganalysis of LSB encoding in color images," Proceedings of IEEE International Conference on Multimedia, Vol. 3, pp. 1279-1282, 2000.

[12]A. Westfeld and A. Pfitzman, "Attacks on steganographic systems," Proceedings of 3rd International Information Hiding Workshop, Springer-Verlag, pp. 61-76, 1999.

[13]Zhang T. and Ping X., "A new approach to reliable detection of LSB steganography in natural images," Elsevier Journal of Signal Processing, Vol. 83, pp. 2085–2093, 2003.

[14]Dumitrescu S., Wu X. and Wang X., "Detection of LSB steganography via sample pair analysis," IEEE Transactions on Signal Processing, Vol. 51, No. 7, pp. 1995-2007, 2003.

[15]C. K. Chan and L. M. Cheng, "Hiding data in images by simple LSB substitution," Pattern Recognition, vol. 37, no. 3, pp. 469–474, 2004.

[16]C. C. Chang, J. Y. Hsiao, C. S. Chen, "Finding optimal Least-Significant-Bit substitution in image hiding by dynamic programming strategy," Pattern Recognition, vol. 36, pp. 1583–1595, 2003.

[17]S. K. Chen, "A module-based LSB substitution method with lossless secret data compression", *Computer Standards & Interfaces*, Vol. 33, pp. 367–371, 2011.

[18]C. C. Thien and J. C. Lin, "A simple and high-hiding capacity method for hiding digit-bydigit data in images based on modulus function," Pattern Recognition, vol. 36, pp. 2875–2881, 2003.

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

[20]Chi-Kwong Chan, L.M. Cheng, Improved hiding data in images by optimal moderately signi7cant-bit replacement, IEEE Electron. Lett. 37 (16) (2001) 1017–1018.

Rajeev Kumar received his B.Tech. in Information Technology from Utter Pradesh Technical University, Lucknow, India and M.Tech. in Information Systems from Netaji Subhas Institute of Technology, Delhi University, India. He is pursuing his doctoral degree is at the Computer Engineering Department, Netaji Subhas Institute of Technology, Delhi University, Delhi, India. His research areas lie in the area of digital watermarking and image processing.

Khushil K. Saini received his B.Tech. in Instrumentation Engineering from Kurukshetra University, Kurukshetra and M.Tech. in Computer Technology from Indian Institute of Technology, Delhi, India. He is pursuing his doctoral degree is at the Computer Engineering Department, Netaji Subhas Institute of Technology, Delhi University, Delhi, India. His research areas lie in the area of digital watermarking, image processing, and compiler design.

Satish Chand received the Master of Science in Mathematics from Indian Institute of Technology, Kanpur, India, Master of Technology in Computer Science from Indian Institute of Technology, Kharagpur, India and PhD in Computer Science from Jawaharlal Nehru University, Delhi, India. Currently he is working as Professor in Computer Engineering Department, Netaji Subhas Institute of Technology, Delhi, India. His research interests include image processing, digital watermarking, video processing, and wireless sensor networks.

