# ZIG-ZAG PVD – A Nontraditional Approach

M.Padmaa
Associate Professor/ECE
Saranathan college of engineering
Trichy 620012

Dr.Y.Venkataramani
Director(Academics)
Saranathan College of engineering
Trichy 620012

## ABSTRACT
Communication has to be secure in order to be kept private. The latest developments in the field of digital communication have made secret communication possible. Image hiding is a method in which a secret image is hidden in a cover image thereby forming a hybrid or stego image. In this paper, data hiding is performed by taking difference value of three and two neighbouring pixels by adapting Zig-Zag traversing scheme (ZZTS). This method enhances security and the quality of image in spite of high capacity of concealed information. Error correction mechanism using hamming code is applied to ensure reliable secret communication. The effectiveness of the proposed stego system has been estimated by computing Mean Square Error(MSE),Peak Signal to Noise Ratio (PSNR), Mean Structural SIMilarity index(MSSIM) and Bits per colour Pixel. This paper also illustrates how security has been enhanced using this algorithm

## Categories and Subject Descriptors
Information hiding
Security and Protection

## General Terms
Data Security

## Keywords
LSB steganography, Information hiding, Pixel value Differencing (PVD), Zig-Zag traversing scheme (ZZTS)

## 1. INTRODUCTION
Steganography is the science of invisible communication. Information is transmitted by hiding it in innocuous cover objects to maintain security and confidentiality. In image steganography the cover object is the image and information is embedded in to images which may be color, grayscale or binary. A stego image is obtained from the cover image by accommodating the secret message into a digital image using some embedding algorithm that slightly modifies the cover image.

Digital Image Steganographic techniques have grown enormously [1-3] in order to enhance the security in a communication channel. The stego-image is later transmitted via a public channel. The public channel can have many trespassers who will want to disrupt the data flow from the sender to the receiver or might want to extract the data transmitted without the knowledge of the communicating parties. Out of the numerous steganographic methods proposed, Least Significant Bits (LSB) substitution is the most popular and simple method that utilizes the least bits of a pixel in the cover image for embedding.

A comparative analysis of various digital steganographic techniques are discussed in [2] which are capable of producing a secret-embedded image that is indistinguishable from the original image to the human eye.

A genetic algorithm based optimal LSB substitution is available to get better stego-image quality than the simple LSB method [18]. In addition, Chang et al. proposed [6] a fast and efficient optimal LSB method based on the dynamic programming strategy that improves the computation time of Wang *et al*.'s scheme [18]. A novel simple LSB technique based on optimal pixel adjustment is presented in[4] and Lin also presented a simple LSB scheme based on the modulus function for improving the stego-image quality [14]. Wang has proposed two new schemes based on the modulo operator [17]. An effective steganographic scheme has to be implemented that thwarts the attacker from extracting the secret information during transmission and reception [20].

The LSB-based methods mentioned above, directly embed the secret data into the spatial domain in an unreasonable way without taking into consideration the difference in hiding capacity between edge and smooth areas. In general, the alteration tolerance of an edge area is higher than that of a smooth area. That is to say, an edge area can conceal more secret data than a smooth area. With this concept in mind, Wu and Tsai presented steganographic scheme that offers high imperceptibility to the stego-image by selecting two consecutive pixels as the object of embedding. The payload of Wu and Tsai's scheme is determined by the difference value between the pixels [22, 23]. Various authors have discussed different techniques in steganography ([5], [20] and [22]).

Recently, various kinds of steganalysis detectors have been under steady development. For example, the well-known RS steganalytic algorithm [7] by Fridrich *et al*. is able to detect the existence of LSB steganography based on the capacity of the hidden message. Especially the proposed algorithm could detect the existence of the LSB scheme with high precision if the hidden capacity is more than 0.005 bits per pixel. In these schemes, majority of the time, authors have adopted Raster scan [1-18, 20-25] for data embedding and extracting processes it traverse the image pixels from left to right and top to bottom as shown in Fig 1.

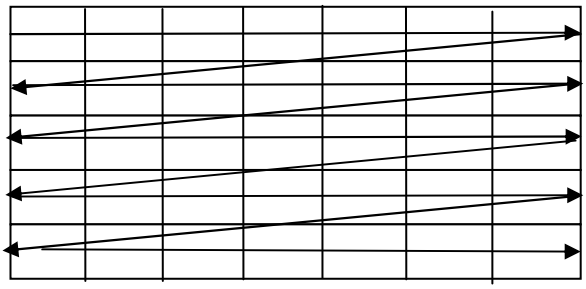


Figure 1. Raster scanning in images

However, the stego-image creation uses simple raster scan for embedding and extraction so there is a possibility of vulnerability of secret data threat. but it is obvious that, if random scan is employed instead of raster scan in secret data embedding, the effectiveness can be improved significantly.

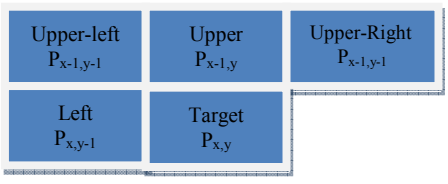
Therefore, it has been obvious that an information hiding scheme must consists of the following. Initially, the stego image should be acceptable so that the human eye cannot identify the embedded data from the stego image. Next, the scheme should offer high payload so that more secret information could be embedded with high imperceptibility. In addition, the complexity against an adversary will be increased to many folds by adapting Zig-Zag traversing scheme (ZZTS) based embedding instead of raster scan for embedding. Last but not the least, in random scan applied to any stego system the key length plays a major role. But in the proposed method no key is shared between the parties. So the legitimate user could more correctly extract the embedded data from the stego-image without keys.

This proposed methodology enhances the Chang et al. technique [5] by increasing the embedding capacity and improves the stego image quality uses Thein *et al*. algorithm [14] by adapting Zig-Zag traversing scheme (ZZTS). The observations made with various images using the proposed scheme validate the same. To further extend the overall capacity of the proposed algorithm, a procedure to determine how many bits to be inserted into a target pixel by using largest difference value between the immediate three pixels is adopted in color image. Additionally modular operation and error correction scheme are also implemented to heighten the image quality and for reliable secret data transfer. This method is found to be have more efficient compared to the existing methods and quality of the image is also not degraded. The paper is arranged into the following sections: Section 2: steganographic algorithm for color image. Section 3: experimental results. Section 4: conclusion.

## 2. Steganographic algorithm for color image

### 2.1 The proposed methodology:

The proposed system uses three pixels adjacent to a target pixel in the embedding process as shown in Fig 2 and 3.




$P_{x,y}$ :pixel value at point(x,y: coordinates)

Figure 2. Schematic of the pixel arrangement in color image RED plane alone.

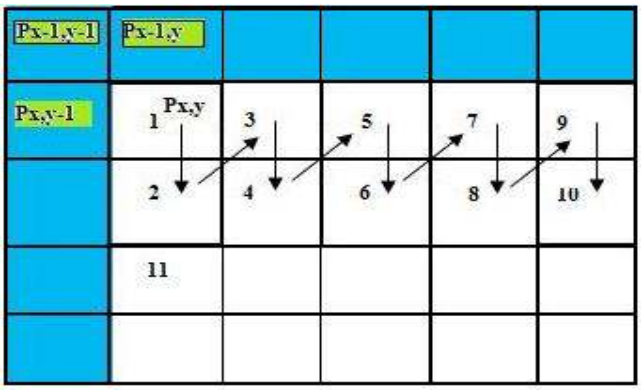



Figure 3. the proposed Zig-Zag traversing scheme for embedding

The Insertion of secret Information

The present method refers to the three neighboring pixels that have already finished insertion process to embed the secret message into the target pixel (refer to Fig. 1 and Fig. 2). in the cover image: Given target pixel $P_{x,\,y\,,1}$, with color pixel value in plane 1 with gray value $g_{x,y}$, let $g_1$, $g_2$, $g_3$ be the gray values of its upper $P_{x-1,\,y\,,1}$, left $P_{x,\,y-1\,,1}$ and upper-left $P_{x-1,\,y-1\,,1}$ pixel values in red plane respectively

### 2.2.1 The embedding procedure:

**Input:** Cover Image (C), Secret data(M).

**Output:** Stego output (S).

Step 1: Read the color cover Image(C) and secret data to be embedded (M).

Step 2: Separate Color image into RGB individual plane.

Step 3: Run the Hamming code module on secret data (M), convert them to binary format (b).

Step 4: Call the Zig-Zag traversing Path for embedding the data, repeat 4.1 to 4.5 till the last secret data obtained from Step 3. is embedded.

Step 4.1: Select the maximum and the minimum gray values(g) among the three pixel values that have already finished the embedding process. Calculate the difference value d between the maximum pixel value and the minimum pixel value using the following among the upper pixel (g1), left pixel (g2) and the upper left pixel (g3) in a given target pixel $g_{x,y}$ by

$$d = [\max(g1, g2, g3) - \min(g1, g2, g3)] \quad (1)$$

Using equation (1), we get an idea as to whether the target pixel is included in an edge area or in smooth area. This is how the number of bit n, inserted into the target pixel is determined by value d.

Step 4.2: Calculation of n: the number of the insertion bits in a target pixel $P_{x,y}$ is calculated, using the following formula:

$$n = \begin{cases} \lfloor \log_2 d \rfloor & if \quad d > 3 \quad d = odd \\ \lfloor \log_{2d} \rfloor - 1 & if \quad d > 3 \quad d = even \\ 1 & if \quad d < 3 \end{cases} \quad (2)$$

If the value of d is less or equal to 3, n in $P_{x,y}$ is determined to be 1, otherwise value of n is taken from the result calculated using equation (2). We appropriately adjust n to enhance both the capacity and the imperceptibility within the cover image.

Step 4.3: Calculate a temporary value $t_{x,y}$ using:

$$\mathbf{t_{x,y} = b - g_{x,y} \bmod 2^n} \quad (3)$$

Where, b is the decimal representation of secret messages for the n bits.

Step 4.4: To make the quality of the image higher, select the nearest value to the target pixel's value of the cover image by optimal pixel adjustment process[4].

$$t1 = \begin{cases} t & if \quad -\lfloor (2^n - 1)/2 \rfloor \le t \le \lfloor (2^n - 1)/2 \rfloor \\ t + 2^n & if \quad -(2^n + 1) \le t < -\lfloor (2^n - 1)/2 \rfloor \\ t - 2^n & if \quad (2^n - 1)/2 \le t < 2^n \end{cases} \quad (4)$$

Step 4.5: Finally, we can get the new pixel value g *

$$g^* = g + t1; \quad (5)$$

Step 5: Combine RGB plane to form color stego image (S)

### 2.2.1 The Extraction procedure:

In the extraction process, given the stego-image S, the embedded messages can be readily extracted without referring to the original cover-image. Using the same sequence as in the embedding process, the set of pixels storing the secret message bits are selected from the stego-image by adapting Zig-Zag traversing scheme.

**Input:** Stego Image(S)

**Output:** Secret data (M).

Step 1: Read the stego image(S)

Step 2: Call the Zig-Zag Traversing procedure for extraction of the data, repeat 3 to 5 till the last secret data is extracted.

Step 3: Calculate the difference value d between the upper pixel (g1) left pixel (g2) and the upper left pixel (g3)

$$d = [\max(g1, g2, g3) - \min(g1, g2, g3)] \quad (6)$$

Step 4: Calculate n that is the number of the insertion bits in a target pixel P from d

$$n = \begin{cases} \lfloor \log_2 d \rfloor & if \quad d > 3 and \quad d = odd; \\ \lfloor \log_2 d \rfloor - 1 & if \quad d > 3 and \ d = even; \\ 1 & if \quad d < 3 \end{cases} \quad (7)$$

Step 5: Finally, Calculate the value of b by

$$b = \bmod(g^*, 2). \quad (8)$$

The decimal value b is represented into a binary number of n bits.

Step 6: Call the hamming code retrieval module to extract the data without errors

Step 7: Convert them to suitable format and save secret data (M).

### 2.2.3 Hamming code generation:

Let N be the number of characters and art1 be the corresponding 8 bit representation of the character.

For example if there are 3 characters art1 will be a 3*8 matrix.

1. Define generator matrix

$$ge = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 1 & 1 \end{bmatrix} \quad (9)$$

2. Generate parity check matrix 'h' using gen2par() function available in matlab.

3. Multiply generator and take modulo 2 for the product that is the code to be transmitted. Let it be R.

### 2.2.4 Hamming code Extraction:

1. Obtain c= h*R'
2. Take modulo2 for c
3. Compare the column of c with column of h and if it is equal error is present then complement the corresponding bit of received codeword R else no error is present.

## 3. Result and Discussion

For implementing the above discussed process, four standard color cover images shown in Figure 4a, 4b, 4c and 4d of size 256 x 256 pixels namely Lena, Baboon, Airplane and Brihadeeswarar Temple have been selected. The effectiveness of the stego process proposed has been studied through the following three different metrics for all the four digital images in RGB planes. The results obtained have been tabulated separately for the three different planes and also the average of all the planes has been calculated along with the number of bits embedded per pixel.

Figure 4 a. Lena 4.b. Baboon

Figure 4.c. Airplane 4.d. Brihadeeswarar Temple

Peak Signal to Noise Ratio (PSNR)

The PSNR is calculated using the equation,

$$PSNR = 10 \log_{10}\left(\frac{I_{\max}^2}{MSE}\right) dB \quad (9)$$

where $I_{max}$ is the intensity value of each pixel which is equal to 255 for 8 bit gray scale images. Higher the value of PSNR better the image quality

Mean Square Error (MSE)

The MSE is calculated using the equation,

$$MSE = \frac{1}{MN}\sum_{i=1}^{M}\sum_{j=1}^{N}\left(X_{i,j} - Y_{i,j}\right)^2 \quad (10)$$

where *M* and *N* denote the total number of pixels in the horizontal and the vertical dimensions of the image *Xi, j* represents the pixels in the original image and *Yi, j*, represents the pixels of the stego-image.

Mean Structural SIMilarity Index (MSSIM) (Zhou Wang et al 2004)[26]

We use a mean SSIM (MSSIM) index to evaluate the overall image quality using the equation,

$$\text{MSSIM(X,Y)} = \frac{1}{M}\sum_{j=1}^{M} SSIM\left(X_j, Y_j\right)$$

$$\text{SSIM(X,Y)} = \frac{\left(2\mu_x\mu_y + C_1\right)\left(2\sigma_{xy} + C_2\right)}{\left(\mu_x^2 + \mu_y^2 + C_1\right)\left(\sigma_x^2 + \sigma_y^2 + C_2\right)} \quad (11)$$

where $C_1 = (K_1 L)^2$ $L=255$

$K_1 = .01$

$C_2 = (K_2 L)^2$ $L=255$

$K_2 = .03$

$$\mu_x = \frac{1}{N}\sum_{i=1}^{N} x_i$$

where $\mu_x$ is the estimate of the mean intensity of the cover image for N= 255, which is representing the total number of pixels, $\sigma_x$ is the standard deviation (the square root of variance) as an estimate of the signal contrast.

$$\sigma_x = \left(\frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \mu_i\right)^2\right)^{\frac{1}{2}}$$

Geometrically, the correlation coefficient $\sigma_{xy}$ is computed using $\sigma_x$ and $\mu_x$ is given by

$$\sigma_{xy} = \frac{1}{N-1}\sum_{i=1}^{N}\left(x_i - \mu_x\right)\left(y_i - \mu_y\right)$$

The value of MSSIM is in the interval [1, 0]. The value 1 means that the two images are exactly the same and 0 means they are totally unrelated.

All the estimating parameters of the four stego covers have been performed using Intel Core2 Duo CPU processor @ 1.60 GHz, 1GB RAM using indigenous MATLAB 7.1 coding.

The number of pixels embedded in the proposed ZZTS based PVD method, differs from cover to cover as in other methods. This difference is attributed to the contrast of the images used. If there is a cluster of pixels, with similar pixel values would have less information embedded in them when compared to a similar cluster with different values. This is the reason why the images Lena and Airplane would have very less no. of bits per pixel when compared to Baboon.

The Table 1 confirms these points above. Mean Squared Error (MSE) increases with increase in the no. of bits embedded in the image. PSNR is inversely proportional to MSE.

Zig zag method betters the conventional pixel value differencing method in terms of no. of bits embedded per pixel. This is accomplished because of the traversing scheme that is used in the method.

From the results it has been observed that Babboon has the highest embedding capacity of 646713 bits with utmost 10 bpp. Where it supersede the results of the methods proposed in [5, 14, 24]. Lowest among four observed in Lena 345297 bits with 5.31 bpp.

The corresponding stego output for full embedding capacity has been shown in Fig 5.

Fig 5. Stego output for full embedding capacity

## 4. SECURITY ANALYSIS:

Since hamming code is used it transforms a character of 8bits into 12 bit code word so number of bits increases by a factor of 1.5. So number of combinations may be 2^12

Then number of bits embedded in a pixel depends on a factor n=log2 (d);

d may vary between 0 and 255.

n may vary from 1 to 7So number of bits embedded in a pixel will vary from 1 to 7. n can be anything between 1 to 7. So, there are 7 possibilities

Hence complexity increases by a factor 2^12*7. In addition, if the secret information is encrypted before embedding then the complexity level to extract the secret information will be high.

## 5. CONCLUSION

The proposed methodology is found to be superior to normal LSB substitution and other sequential methods in terms of embedding capacity and imperceptibility. This methodology was also tested for various cover images by vigorously passing through various performance criteria. The greatest asset of pixel value differencing is its discern ability. One disadvantage of pixel value differencing is its predictability of the order of image. The Zig Zag method exploits the ability of the PVD and betters PVD in terms of predictability. Since the scanning process is random, it will not be possible for the would be attackers to even determine the method of scanning, so that their attack, if any, fails.

## 6. Acknowledgement

The first author wishes to thank Prof. R.Amirtharajan Assistant Professor / ECE School of Electrical & Electronics Engineering, for his valuable guidance, D.ADHARSH and V.VIGNESH former ECE Students / SEEE/ SASTRA University for their technical support.

Table 1: The estimating parameters for Zig-Zag PVD Scheme

| Cover image | RGB Plane | Total number of bits embedded | Bits per Pixel (bpp) | MSE | PSNR (dB) | MSSIM |
|---|---|---|---|---|---|---|
| Lena | R | 115460 | 1.776 | 1.7737 | 45.6419 | 0.9938 |
| | G | 117321 | 1.804 | 1.9951 | 45.1312 | 0.9939 |
| | B | 112516 | 1.730 | 1.6365 | 45.9917 | 0.9937 |
| Lena | RGB avg | 345297 | 5.31 | 1.8017 | 45.5883 | 0.9938 |
| Baboon | R | 213359 | 3.281 | 8.8952 | 38.6393 | 0.9844 |
| | G | 213897 | 3.290 | 8.8410 | 38.6658 | 0.9852 |
| | B | 216408 | 3.330 | 9.1579 | 38.5129 | 0.9851 |
| Baboon | RGB avg | 646713 | 9.901 | 8.9647 | 38.6060 | 0.9849 |
| Airplane | R | 125771 | 1.934 | 3.1901 | 43.0928 | 0.9927 |
| | G | 125837 | 1.935 | 3.5753 | 42.6712 | 0.9928 |
| | B | 122490 | 1.884 | 2.8424 | 43.5940 | 0.9919 |
| Airplane | RGB avg | 374098 | 5.753 | 3.2026 | 43.1193 | 0.9925 |
| Temple | R | 142245 | 2.187 | 3.9739 | 42.1385 | 0.9927 |
| | G | 139880 | 2.151 | 3.7759 | 42.3606 | 0.9926 |
| | B | 138715 | 2.133 | 3.5625 | 42.6133 | 0.9922 |
| Temple | RGB avg | 420840 | 6.471 | 3.7708 | 42.3708 | 0.9925 |

## 7. REFERENCES

[1]. Abbas Cheddad, Joan Condell, Kevin Curran, Paul Mc Kevitt, Digital image steganography: Survey and analysis of current methods Signal Processing 90 (2010) 727–752

[2]. R.Amirtharajan.R.Akila. P.Deepikachowdavarapu Article: A Comparative Analysis of Image Steganography. *International Journal of Computer Applications* 2(3):41–47, May 2010

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

[4]. C.K. Chan, L.M. Chen, Hiding data in images by simple LSB substitution, Pattern Recognition 37 (3) (2004) 469–474.

[5]. Chang, C.C., Tseng, H.W., 2004. A steganographic method for digital images using side match. Pattern Recognition Letter 25 (September), 1431–1437.

[6]. Chang, C.C., Hsiao, J.Y., Chan, C.S., 2003. Finding optimal least-significant-bit substitution in image hiding by dynamic programming strategy. Pattern Recognition 36 (July), 1583–1595.

[7]. Fridrich, J., Goljan, M., Du, R., 2001. Reliable detection of LSB steganography in color and grayscale images. In: Proceedings of ACM Workshop on Multimedia and Security, pp. 27–30.

[8]. S. Katzenbeisser, F.A.P. Petitcolas, Information Hiding Techniques for Steganography and Digital Watermarking, Artech House, Norwood, MA, 2000.

[9]. Ker, A.D., 2004. Quantitative evaluation of pairs and RS steganalysis. In: Proceedings of the SPIE Security, Steganography, and Watermarking of Multimedia Contents, vol. 5306, pp. 83–97.

[10]. Liao, Z., Huang, Y., Li, C., 2007. Research on data hiding capacity. International Journal of Network Security 5 (September), 140–144.

[11]. Lin, C.C., Tsai, W.H., 2004. Secret image sharing with steganography and authentication. The Journal of Systems and Software 73 (November), 405–414.

[12]. Lou, D.C., Liu, J.L., 2002. Steganographic method for secure communications. Computers and Security 21 (October), 449–460.

[13]. F.A.P. Petitcolas, R.J. Anderson, M.G. Kuhn, Information hiding—a survey, Proc. IEEE 87 (7) (1999) 1062–1078.

[14]. C.C. Thien, J.C. Lin, A simple and high-hiding capacity method for hiding digit-by-digit data in images based on modulus function, Pattern Recognition 36 (11) (2003) 2875–2881.

[15]. C.M. Wang, N.I. Wu, C.S. Tsai, M.S. Hwang, A high quality steganography method with pixel-value differencing and modulus function, J. Syst. Software 81 (1) (2008) 150–158.

[16]. R.Z. Wang, C.F. Lin, J.C. Lin, Image hiding by optimal LSB substitution and genetic algorithm, Pattern Recognition 34 (3) (2000) 671–683.

[17]. Wang, S.J., 2005. Steganography of capacity required using modulo operator for embedding secret image. Applied Mathematics and Computation 164 (May 2005), 99–116.

[18]. Wang, R.Z., Lin, C.F., Lin, J.C., 2001. Image hiding by optimal LSB substitution and genetic algorithm. Pattern Recognition 34 (March), 671–683.

[19]. Westfeld Space filling curves in steganalysis in E.J Delp III & P.W. Wong(Eds), Security, steganography and watermarking of multimedia contents VII SPIE 5681, (2005) 28-37

[20]. Wen-Jan Chen, Chin-Chen Chang, T. Hoang Ngan Le, High Payload steganography mechanism using hybrid edge detector. Expert Systems with Applications 37 (2010) 3292–3301

[21]. Wu, N.I., Hwang, M.S., 2007. Data hiding: current status and key issues. International Journal of Network Security 4 (January), 1–9.

[22]. Wu, D.C., Tsai, W.H., 2003. A steganographic method for images by pixel-value differencing. Pattern Recognition Letters 24 (June), 1613–1626.

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

[24]. Young-Ran Park, Hyun-Ho Kang, Sang-Uk Shin, and Ki-Ryong Kwon, An Image Steganography Using Pixel Characteristics Y. Hao et al. (Eds.): CIS 2005, Part II, Springer-Verlag Berlin Heidelberg LNAI 3802, (2005) 581–588.

[25]. Yuan-Hui Yu , Chin-Chen Chang, Iuon-Chang Lin, A new steganographic method for color and grayscale image hiding Computer Vision and Image Understanding 107 (2007) 183–194

[26]. Zhou Wang, Alan Conrad Bovik, Hamid Rahim Sheikh, Eero P. Simoncelli, Image Quality Assessment: From Error Visibility to Structural Similarity, IEEE Transactions on Image Processing, 13(4) (2004) 600-612.