Advanced Steganography Method for Digital Images Using Eight-sided Side Match

Salah Mahdi Saleh
Department of computer science, College of science for women, Babylon University
E-mail: mrsalahdotcom@yahoo.com

Abstract
In this paper, we present an advanced steganography method by using the side information of the eight neighboring pixels. It aims to provide a secure steganography system and to minimize the distortion of the stego-image. The method exploits the correlation between neighboring pixels to estimate the degree of smoothness or contrast of pixels to determine the amount of payload in each embedding position. If the pixel is located in edge area, then it tolerates larger changes than those in smooth areas. The eight-sided side match method is used in our scheme. It attempts to share all the surrounded pixels in determining the capacity of each embedding position. The experimental results show that our method provides a high quality image without making noticeable distortion. Another important thing in our advanced method is that the embedded data can be extracted from the stego-image without referencing the original image. This means that the stego-image will depend on itself in extracting embedded information without using any external factor or key.

Keywords: Steganography; Side information; Neighboring; Eight-sided side match

طريقة إخفاء مطورة للصور الرقمية باستعمال التطابق الجانبي الثماني

صلاح مهدي صالح
قسم علوم الحاسبات، كلية العلوم للبنات، جامعة بابل

الخلاصة
في هذا البحث، قدمنا طريقة إخفاء مطورة باستعمال المعلومات الثانوية لعناصر الصورة المتجاورة الثمانية. إنها تهدف لتوفير نظام إخفاء آمن ولتقليل تشوه صورة الإخفاء. تستغل الطريقة العلاقة الموجودة بين عناصر الصورة المتجاورة لتقدير درجة تجانس أو تباين العناصر لتحديد كمية التحمل في كل موقع إخفاء. فإذا حُدِّدَ موقع عنصر الصورة في منطقة حافة فانه يتحمل تغييرات أكبر من تلك العناصر الواقعة في منطقة متجانسة. استُعملت طريقة التطابق الجانبي الثماني في بحثنا. تحاول طريقتنا إشراك جميع العناصر المحيطة بموقع الإخفاء لتحديد سعة كل موقع إخفاء. أظهرت النتائج التجريبية إن طريقتنا توفر صورة ذات نوعية عالية بدون إحداث تشوه لافت للنظر. من أكثر الأشياء المهمة الأخرى في طريقتنا المطورة إن استرجاع البيانات المخفية من صورة الإخفاء يتم دون الإشارة إلى الصورة الأصلية. يعني هذا بأن صورة الإخفاء ستعتمد على نفسها في استرجاع المعلومات المخفية بدون استعمال أي عامل أو مفتاح خارجي.

1. Introduction

In the last years, Internet becomes a popular communication channel. Exchange of information using Internet channel, over far distance, is now an everyday activity. However, that information still have to face some problems [1], such as data security, copyright control, and …, etc. Thus, we need secure secret communication methods for transmitting massage over the Internet. Two techniques are used to transmit secrets using unprotected communication media [2][3][4]: encryption and steganography are the preferred techniques for protection the transmitted data. Encryption conceals the message by scrambling the data being communicated, while steganography hides the message in innocent digital file [3]. The purpose of steganography is to conceal the fact that some communication is taking place [5][2]. With any type of hidden communication, the security of the message often lies in the secrecy of its existence and/or the secrecy of how to decode it. The steganography hides the secret information behind a cover so that it draws no special attention [6]. The cover represents any digital file like image, text, video, sound and …, etc. If we used the digital image, the cover-image after embedding is called stego-image.

The common well-known and widely used steganographic method today is the least significant bits (LSBs) substitution [7][8][2]. Many public steganographical softwares, such as S-Tools, EZstego and Steganos apply this technique [9]. On significant advantage of this method is that it is simple to understand and implement [8]. This technique replaces the fixed-length LSBs of pixels with the embedding data. However, not all pixels in cover image can tolerate equal amount of changes occur in smooth areas can be easily noticed by human eyes. Adaptive methods for steganography are introduced in which the amount of embedding data in a pixel is variable [10][11]. These adaptive methods provide more imperceptible results than those employed by simple LSBs substitution schemes. In our proposed method, we will hide an embedded data by using variable-length LSBs of pixels, and determined depending on the correlation between eight neighboring pixels. Therefore, this method does not replace the bits of embedded data directly, but changes the pixel value into another similar value according to the result of correlation and still nearest to its neighbors especially in smooth areas. The range of changeable pixel value in smooth areas is small and in edge areas is large, so that the stego image still maintains good perceptual quality. This steganographic method provides an acceptable embedding capacity with little perceptual distortion.

The concept of side match is proposed by Kim [12]. It used to enhance the performance of vector quantization (VQ) coding scheme to get high compression ratio. He utilizes the information of two neighboring blocks (upper and left blocks) to predict the state codebook of an input vector. Wei et al. proposed the three-sided and four-sided side match method for VQ encoding [13]. The other two neighboring blocks (bottom and right block) are contributed in prediction. The prediction is improved. Marvel and Retter [14] used the edge information as side information that extracted from the received stego-image to better correct bit errors. This side information is used for denoising to estimate the value of noise in the vicinity of an edge. Chang and Tseng were used the side information of neighboring pixels for each input pixel to help the capacity estimation [1]. A novel steganographic method to hide data in spatial domain of image imperceptible. The number of bits to be embedded for each pixel is variable and determined by the correlation between neighboring pixels. It determined whether the input pixel is located in edge area or not. If pixel is located in edge area, then it may embed more data than those in smooth area. The two-sided side match method is employed. In addition, the three-sided and four-sided side match are used to perform more precise estimation. In this paper, we enlarge the number of neighbors, so all the eight neighbors will help in estimation the capacity of each input pixel and make the process of embedding more and more precise and the stego-image less distortion. The rest of this paper is organized as follows. A newly scheme is described in Section 2. The implementation results are presented in Section 3. The conclusions are shown in Section 4.

2. The Proposed Method

The correlation between neighboring pixels determines whether the input pixel is located in edge area or not. Knowing the location of each input pixel contributes in determining how many bits can embed in that

pixel (capacity estimation). If the input pixel is located in edge area, then it may embed more data than those in smooth area. In this paper, we developed the equations that used in [1] to fit eight-sided side match. The secret data are embedded in the cover image in a raster scan order.

2.1. Eight-sided Side Match Steganography

Our method uses the side information of all eight neighboring pixels (upper, left, right, bottom, bottom left, bottom right, upper left, upper right) instead of only two, three, or four of them to make a more precise estimation. Let P_U, P_L, P_R, P_B, P_{BL}, P_{BR}, P_{UL} and P_{UR} be the eight neighboring pixels of a given pixel P_X with gray level g_x and let g_u, g_l, g_r, g_b, g_{bl}, g_{br}, g_{ul} and g_{ur} be the gray levels of its eight neighbors, respectively. Then the difference value *d* between a given pixel and its neighbors is computed as:

$$d = \frac{\left(g_u + g_l + g_r + g_b + g_{bl} + g_{br} + g_{ul} + g_{ur}\right)}{8} - g_x. \qquad (1)$$

From the above equation we can known the location of input pixel. A small difference value indicates that the pixel in a smooth area, whereas a large difference value indicates that pixel in an edge area. The pixels in edge area tolerate larger changes than those in smooth area. The number of bits *n*, which can be embedded in this pixel, is calculated by

$$n = \log_2 |d| \qquad \text{if } |d| > 1. \qquad (2)$$

A sub stream with n bits in the embedded data is converted to integer value *b*. Then a new difference d' is computed as

$$d' = \begin{cases} 2^n + b, & \text{if } d > 1, \\ -(2^n + b), & \text{if } d < 1. \end{cases} \qquad (3)$$

Finally, the new value of the input pixel P_X is defined to be

$$g'_x = \frac{\left(g_u + g_l + g_r + g_b + g_{bl} + g_{br} + g_{ul} + g_{ur}\right)}{8} - d'. \qquad (4)$$

The following figure illustrates the sampling arrangement for the eight-sided side match. This figure is an example of 8×8 image, the shaded pixels remains unchanged while the blank pixels are used for embedding the secret data. From figure 1, we can see that the positions of embedding in the last column and last row haven't all the eight neighboring pixels and each embedding position in it has only five neighboring pixels and lost three. Although these positions are little, we can use it to increase the capacity. Therefore, for this situation, we suppose that there is an additional column and additional row with values of the previous column and row, respectively. In order to guarantee that the new value of pixel P_X haven't fall off the boundary of the range [0,255], the fall off boundary checking process is applied to data embedding and extraction. If the new value of pixel P_X falls off the boundary, it's not used for embedding. Similarly, if the checking process in data extraction found that the pixel probably falls off the boundary, then the pixel is

skipped without extraction. The process of extraction the embedded data is easy. As the process of embedding, the embedded data is extracted in the raster scan order except for the first column and first row. Given an input pixel P_X^* with gray level g_X^*. Let g_u, g_l, g_r, g_b, g_{bl}, g_{br}, g_{ul} and g_{ur} be the gray levels of its eight neighbors, respectively. Then, the difference value d^* is computed as:

$$d^* = \frac{\left(g_u + g_l + g_r + g_b + g_{bl} + g_{br} + g_{ul} + g_{ur}\right)}{8} - g_x^*. \qquad (5)$$

The number of bits *n*, which can be embed in this pixel, is calculated by:

$$n = \log_2 \left| d^* \right| \qquad \text{if } |d| > 1. \qquad (6)$$

Finally, the embedded value *b* is extracted out using the following equation:

$$b = \begin{cases} d^* - 2^n, & \text{if } d^* > 1, \\ -d^* - 2^n, & \text{if } d^* < 1. \end{cases} \qquad (7)$$

The value *b* is then converted to its corresponding binary string with *n* bits length. The key factor in our algorithm is the difference value *d*. Therefore, the process of extraction the embedded data is no need of referencing the original image. In the eight-sided side match, all eight surrounding pixels cannot be used for embedding data. Thus the embedded capacity will be less than that of two, three and four-sided side match. However, fewer changes and precise estimation make a more imperceptible and high quality steganography.

From the above paragraph, we see that the algorithm depends on the value of difference *d*. Therefore, our attention concentrates on this value to modify our algorithm in order to enhance the results as we see later in the experiments. Thus, in the following, we make the overall of the nine pixels (eight neighboring pixels as well as the given pixel P_X) contributing in determining the value of *d*. We mean by that we compute the difference value depending on the principle of window. The difference value *d* is computed using the following formula:

$$d = \frac{\left(g_u + g_l + g_r + g_b + g_{bl} + g_{br} + g_{ul} + g_{ur} + g_x\right)}{8} - g_x. \qquad (8)$$

and the new value of the input pixel P_X is computed as:

$$g'_x = \frac{\left(g_u + g_l + g_r + g_b + g_{bl} + g_{br} + g_{ul} + g_{ur} + g_x\right)}{8} - d'. \qquad (9)$$

In the extraction process, the equation 5 is modified as follow:

$$d^* = \frac{\left(g_u + g_l + g_r + g_b + g_{bl} + g_{br} + g_{ul} + g_{ur} + g_x\right)}{8} - g_x^*. \qquad (10)$$

The other equations 2, 3, 6 and 7 still without any modification.

3.The Experiments and Results.

To stand on the performance of the advanced method, four standard 512×512 gray level images (8 bit/pixel) showing in appendix A were used as the test images in our experiments illustrate in below. Also, we are used the peak signal to noise ratio (PSNR) as a measure of the steganography quality. For the secret message, we used a random bit stream as the embedded data in the experiments.

3.1. Embedding in Gray-level Images Without Embedding in Last Row and Last Column.

In this experiment, we will embed the secret message in gray level images using Equation 1 to compute the value of *d*. The position of embedding in the last column and the last row will not be used to embed the secrete data. The appendix B illustrates the cover-image and stego-image using eight-sided side match. We can see from it that the resulted distortion from embedding the secret data is imperceptible to human visual system. It means that distortions will be difficult to noticeable because most of the highest changes happen in edge areas of images and they are less conspicuous to human eyes. Also, we found from that appendix most of the distortions appeared in the edges of images and that make it unnoticeable. The embedded capacity and the PSNR values are given in Table 1. It also contains a comparison between the PSNR values of our proposed method and that used in [1].

It is seen that the proposed method has large PSNR value for Lena image and Baboon image than in [1]. The two-sided side match steganography has larger embedded capacity, whereas eight-sided side match steganography has less distortion. This came from contributing all the eight neighboring pixels in determining the capacity of the input pixel, and makes the embedded process more precise.

The results of this experiment also show that the image Baboon has much embedded capacity than the other images in same experiment. The reason is that the baboon image is more complicated and has more edges than other images. It conforms to our issues that are mentioned before, which that the pixels in edge areas may tolerate larger changes than smooth areas without making perceptible distortion. However, we can see from Figure 4 that Baboon image stills have a good quality after embedding large amount of data.

3.2. Embedding in Gray-level Images Using Sample Arrangement as in Figure 1.

In this experiment, we will embed the secret message in gray level image using the sample arrangement as in Figure 1. This figure is an example of 8×8 image with embedding secret data in the last column and the last row. We illustrate the cover-image and stego-image for this experiment in appendix C. This experiment has a large capacity than previous because it embeds data in the last column and the last row. The qualities of stego-images still good and their distortions are imperceptible to human visual system because contributing all the eight neighbors in determining the value of payload in each embedding position and that make them results precise. The embedded capacity and the PSNR values are given in Table 2. It also contains a comparison between the PSNR values of our proposed method and that used in [1].

It is seen that the proposed method has large PSNR value for Lena image and Baboon image than the results in [1]. This came from contributing all the eight neighboring pixels in determining the capacity of the

input pixel, and makes the embedded process more precise. The capacity of embedding in this experiment has been increased than previous experiment as a result to using the last column and last row.

3.3. Using Equation 8 to Compute the Value of Difference.

In this experiment, we will make all the surrounded pixels of the input pixel as well as the input pixel itself contributing in determining the value of difference *d* depending on the principle of window (meaning by that we will compute the mean). The value of *d* can be obtained by using Equation 8. The same four standard gray images are used to stand on the result of this experiment. The values of PSNR are given in Table 3. It also contain a comparison between the PSNR values of our proposed method and that used in [1].

It is seen that the two, three and four-sided side match steganography have large capacity than proposed algorithm, whereas our method has less distortions. Another thing can be seen in this experiment that is when we compare the values of PSNR of our proposed algorithm that appear in Table 3 with those in Tables 1 and 2, we will find that it perform best. This is due to contribute all the eight neighboring pixels as well as the input pixel itself in determining the payload in each embedding position and make the results more precise. We can illustrate the cover-image and stego-image for this experiment in appendix D.

We must indicate some problem facing us in this experiment that is some data of the restored message will not be fully as the origin. This came from contributing the input pixel in determining the payload and its value will be changed after insertion a message bits in it. In the extraction stage, that pixel also contributing in determining the quantity of embedded data but its value now differ as compared before embedding and will effect on the extraction process. Our aim from this experiment is that to prove if we use more neighboring pixels in determining the payload we will get best results.

4.Conclusions.

In our advanced steganography algorithm, there is no need of referencing the original image when extracting the embedded data from a stego-image. The stego image depends on itself to extract the secret message by finding the value of difference using the pixels of stego image only. The pixels in edge areas embed more data than those in non-edge areas because the value of difference is large. The explanation for this large value of difference in edge area is that in edge area there is a large change in image brightness (gray levels) over a short spatial distance. We see from the results of that the Baboon image has much larger embedding capacity than other images because it complicated and has more edges. The method has best performance when we are computed the value of PSNR using all the eight neighboring pixels as well as the input pixel (principle of window) in experiment 3.3. It is clear that our method performs better than conventional LSBs substitution method in both visual effect and security. Our experimental results have shown that the advanced algorithm provides better way for embedding large amount of data into cover images without making noticeable distortion.

References.

[1] C.-C. Chang and H.-W. Tseng, A Steganographic Method for Digital Images Using Side Match, Pattern Recognition Letters, Elsevier, Vol. 25, pp. 1431-1437, (2004).

[2] M.A.B. Younis and A. Jantan, A New Steganography Approach for Image Encryption Exchange by Using Least Significant Bit Insertion, International Journal of Computer Science and Network Security, Vol. 8, No. 6, (2008).

[3] A. Al-Jaber and I. Aloqily, High Quality Steganography Model with Attacks Detection, Pakistan Journal of Information and Technology, Vol. 2, No. 2, pp. 116-127, (2003).

[4] H. Motameni, M. Norouzi, M. Jahander and A. Hatami, Labeling Method in Steganography, Proceedings of World Academy of Science, Engineering and Technology, Vol. 24, (2007).

[5] S. Hetzl and P. Mutzel, A Graph-Theoretic Approach to Steganography. In J. Dittmann, S. Katzenbeisser, and A. Uhl, Editors, Communications and Multimedia Security, 9th IFIP TC-6 TC-11 International Conference, CMS 2005, Vol. 3677 of Lecture Notes in Computer Science, pp. 119-128, Salzburg, Austria, (2005).

[6] P. Moulin and Y. Wang, New Results on Steganographic Capacity, In Proceedings of CISS Conference, Princeton, NJ, USA, (2004).

[7] M. Juneja, P.S. Sandhu and E. Walia, Application of LSB Based Steganographic Technique for 8-bit Color Images. Proceedings of World Academy of Science, Engineering and Technology, Vol. 38, pp. 427-429, (2009).

[8] M. Amiri and M.R. Resketi, An Edge Method in Steganography, Proceedings of World Academy of Science, Engineering and Technology, Vol. 37, pp. 1058-1063, (2009).

[9] X. Luo, F. Liu and P. Lu, A LSB Steganography Approach Against Pixels Sample Pairs Steganalysis, International Journal of Innovative Computing, Information and Control, Vol. 3, No. 3, pp. 575-588, (2007).

[10] W.-N. Lie and L.-C. Chang, Data Hiding in Images with Adaptive Number of Least Significant Bits Based on the Human Visual System. Proceedings of IEEE International Conference on Images Processing, Vol. 1, pp. 286-290, (1999).

[11] D.-C. Wu and W.-H. Tsai, A Steganography Method for Images by Pixel-value Differencing, Pattern Recognition Letters, Vol. 24, pp. 1613-1626, (2003).

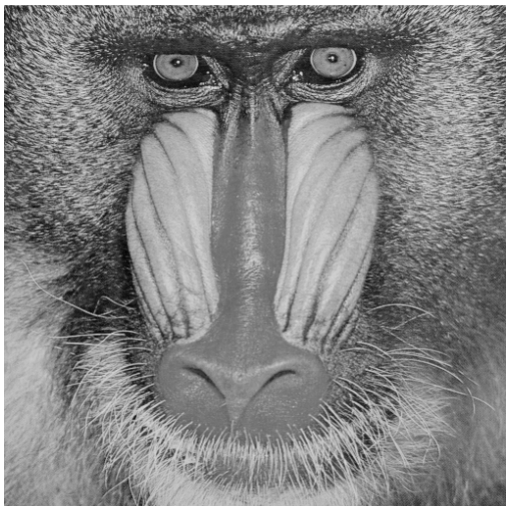
[12] T. Kim, Side Match and Overlap Match Vector Quantizers for Images. IEEE Trans. Images Processing, Vol. 1, pp. 170-185, (1992).

[13] H.-C. Wei, P.–C. Tsai and J.–S. Wang, Three-sided Side Match Finite-state Vector Quantization, IEEE Trans. Circ. Syst. Video Technol. Vol. 10, No. 1, pp. 51-58, (2000).

[14] L.M. Marvel and C.T. Retter, The Use of Side Information in Image Steganography, International Symposium on Information Theory and Its Applications, Honolulu, Hawaii, USA, (2000).

Appendix A

(a)



(b)

(c)

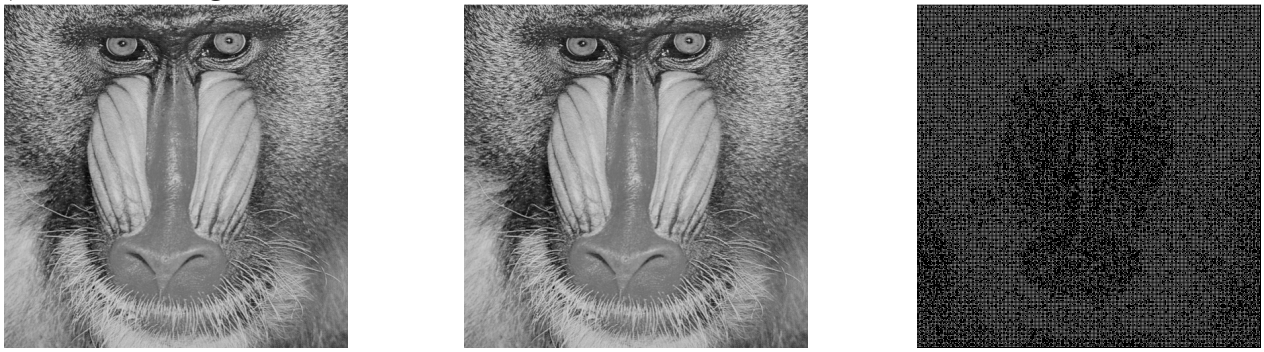
(d)

Figure 2: Original test images: (a) Lena, (b) Baboon, (c) Peppers and (d) Zelda.

Appendix B

(a) (b) (c)

Figure 3: The results of eight-sided side match with Lena image (a) original-image, (b) stego-image and (c) difference image.



(a) (b) (c)

Figure 4: The results of eight-sided side match with Baboon image (a) original-image, (b) stego-image and (c) difference image.

(a) (b) (c)

Figure 5: The results of eight-sided side match with Peppers image (a) original-image, (b) stego-image and (c) difference image.

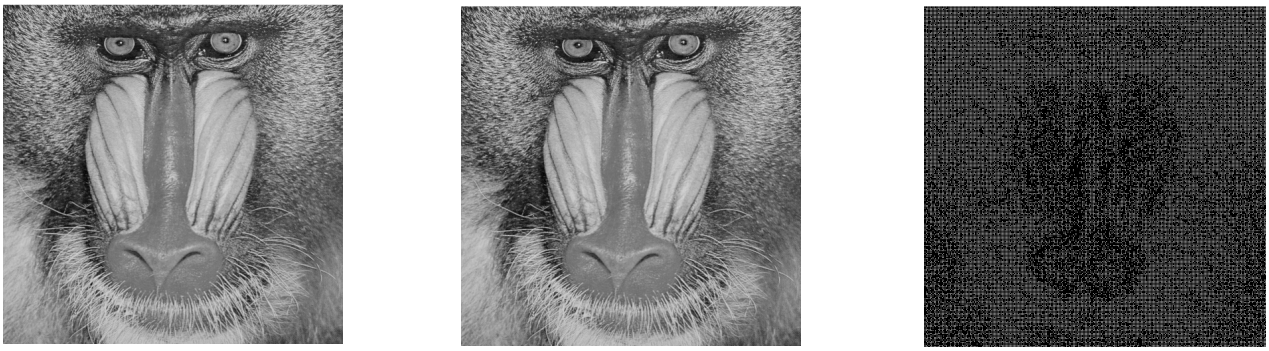
(a) (b) (c)

Figure 6: The results of eight-sided side match with Zelda image (a) original-image, (b) stego-image and (c) difference image.

Appendix C

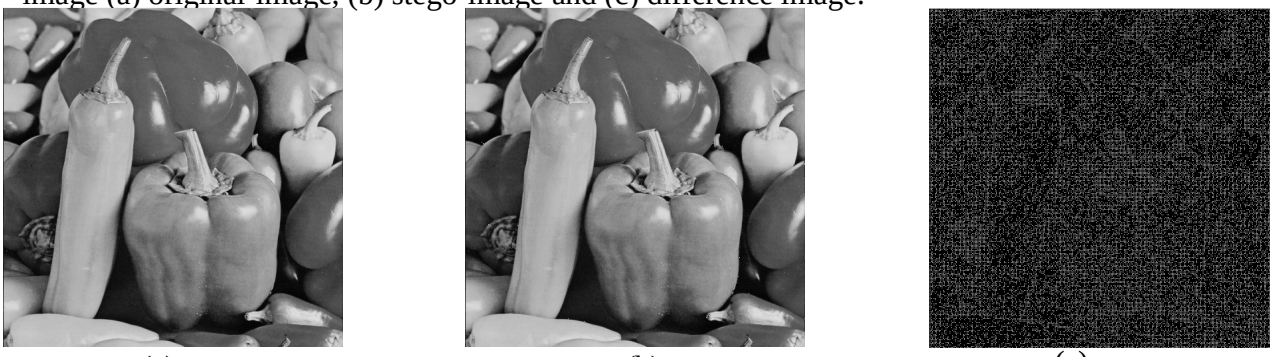
(a) (b) (c)

Figure 7: The results of eight-sided side match using sample arrangement as in Figure 1 with Lena image (a) original-image, (b) stego-image and (c) difference image.



(a) (b) (c)

Figure 8: The results of eight-sided side match using sample arrangement as in Figure 1 with Baboon image (a) original-image, (b) stego-image and (c) difference image.



(a) (b) (c)

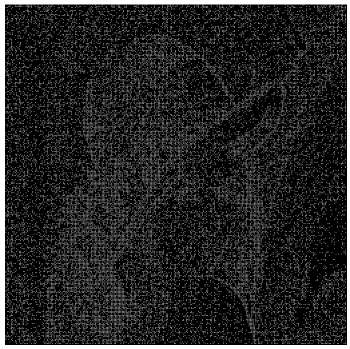
Figure 9: The results of eight-sided side match using sample arrangement as in Figure 1 with Peppers image (a) original-image, (b) stego-image and (c) difference image.

(a) (b) (c)

Figure 10: The results of eight-sided side match using sample arrangement as in Figure 1 with Zelda image (a) original-image, (b) stego-image and (c) difference image.

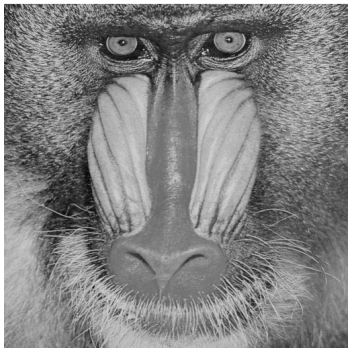
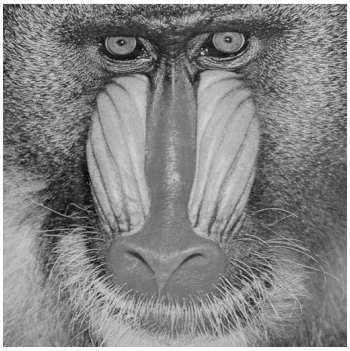
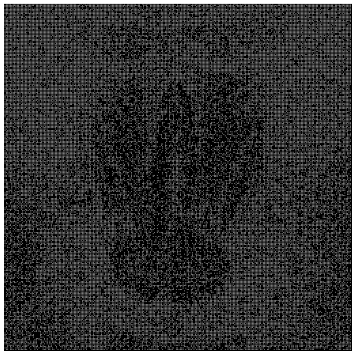
Appendix D





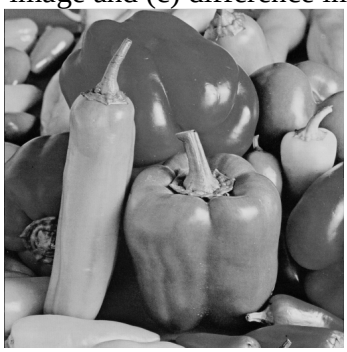
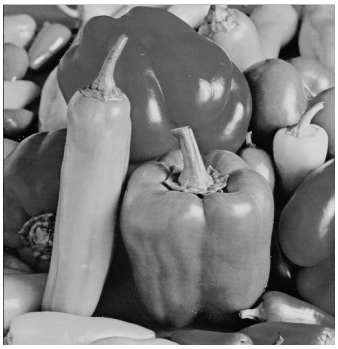
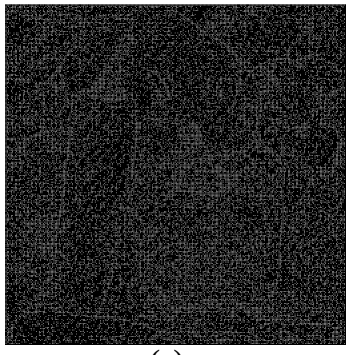
(a) (b) (c)

Figure 11: The results of embedding using equation 8 with Lena image (a) original-image, (b) stego-image and (c) difference image.

(a) (b) (c)

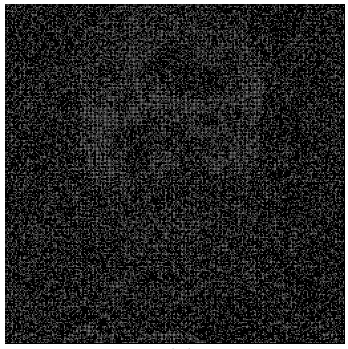
Figure 12: The results of embedding using equation 8 Baboon image (a) original-image, (b) stego-image and (c) difference image.

(a) (b) (c)

Figure 13: The results of embedding using equation 8 with Peppers image (a) original-image, (b) stego-image and (c) difference image.





(a) (b) (c)

Figure 14: The results of embedding using equation 8 with Zelda image (a) original-image, (b) stego-image and (c) difference image.

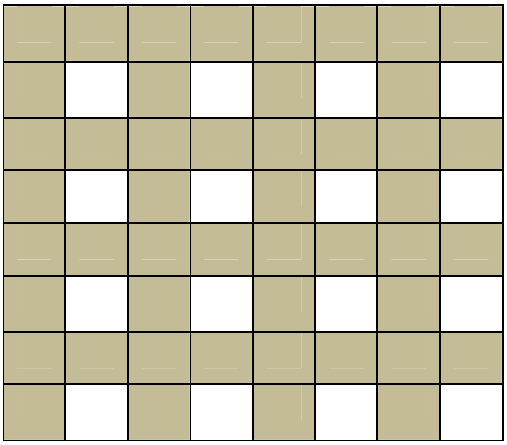


Figure1: The sampling arrangement for eight-sided side match with using last column and last row.

Table 1: Experimental results for our proposed algorithm without embedding in the last column and the last row.

Image name	Proposed algorithm		Algorithm in Chang and Tseng (2004)					
	Eight-sided		Four-sided		Three-sided		Two-sided	
	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)
Lena	48.225	59564	48.18	164538	45.03	267242	41.22	389004
Baboon	39.141	148243	38.56	298413	34.93	483758	33.53	660725
Peppers	41.858	74570						
Zelda	51.956	53236						

Table 2: Experimental results for our proposed algorithm using the sampling arrangement in Figure 1.

Image name	Proposed algorithm		Algorithm in Chang and Tseng (2004)					
	Eight-sided		Four-sided		Three-sided		Two-sided	
	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)
Lena	48.556	60403	48.18	164538	45.03	267242	41.22	389004
Baboon	38.644	149657	38.56	298413	34.93	483758	33.53	660725
Peppers	40.126	75881						
Zelda	51.128	54049						

Table 3: Experimental results for our proposed algorithm using the sampling arrangement in Figure 1 with computing the mean.

Image name	Proposed algorithm		Algorithm in Chang and Tseng (2004)					
	Eight-sided		Four-sided		Three-sided		Two-sided	
	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)	PSNR (dB)	Capacity (bits)
Lena	50.244	52636	48.18	164538	45.03	267242	41.22	389004
Baboon	39.829	138649	38.56	298413	34.93	483758	33.53	660725
Peppers	43.085	66499						
Zelda	52.990	46245						