An Efficient Hybrid Steganography Method Based on Edge Adaptive and Tree Based Parity Check

Hayat Al-Dmour, Noman Ali, and Ahmed Al-Ani

Faculty of Engineering and Information Technology, University of Technology, Sydney, Ultimo NSW 2007 Australia
{HayatShahir.T.Al-Dmour,Noman.Ali}@student.uts.edu.au,
Ahmed.Al-Ani@uts.edu.au

Abstract. A major requirement for any steganography method is to minimize the changes that are introduced to the cover image by the data embedding process. Since the Human Visual System (HVS) is less sensitive to changes in sharp regions compared to smooth regions, edge adaptive has been proposed to discover edge regions and enhance the quality of the stego image as well as improve the embedding capacity. However, edge adaptive does not apply any coding scheme, and hence it embedding efficiency may not be optimal. In this paper, we propose a method that enhances edge adaptive by incorporating the Tree-Based Parity Check (TBPC) algorithm, which is a well-established coding-based steganography method. This combination enables not only the identification of potential pixels for embedding, but it also enhances the embedding efficiency through an efficient coding mechanism. More specifically, the method identifies the embedding locations according to the difference value between every two adjacent pixels, that form a block, in the cover image, and the number of embedding bits in each block is determined based on the difference between its two pixels. The incorporation of TBPC minimizes the modifications of the cover image, as it changes no more than two bits out of seven pixel bits when embedding four secret bits. Experimental results show that the proposed scheme can achieve both large embedding payload and high embedding efficiency.

Keywords: steganography, edge adaptive, human visual system (HVS), Tree-Based Parity Check (TBPC)

1 Introduction

Internet is playing an essential role in data transmission and sharing. The protection of confidential information when transmitting sensitive information over the Internet by government organizations, industry and individuals is necessary. Accordingly, intensive research has been conducted on information security [1].

X. He et al. (Eds.): MMM 2015, Part I, LNCS 8935, pp. 1–12, 2015.

Cryptography is used to add some kind of secrecy to communication channels [2]. It encrypts information into a non-readable form using substitution or permutation operations so that if obstructed, the transmitted information cannot be understood [3,4]. However, this approach attracts the attention of unauthorized intruders [5]. An alternative solution to this problem is steganography. Steganography is a method of concealing the existence of confidential data under cover media in such a way that no one has knowledge about the existence of the secret data except the authorized receiver [2,4,5,6,7]. Thus, steganography is concerned with hiding the existence of a data while cryptography hides the meaning [3,4]. Steganography algorithms aim to enhance imperceptibility, security and capacity [8].

In recent years, a large number of steganography techniques have been published. Some steganography schemes hide the secret data in the spatial domain of an image. Other steganography methods use transform domain such as Discrete Cosine Transform (DCT) and Discrete Transform Wavelet (DWT) [6]. In order to achieve a highly secured system of data hiding, a number of researches attempted to encrypt the data before embedding it in a cover media [3,4]. Most of the existing steganography algorithms are lossless because in some applications such as those related to health and military loss of confidential data is not acceptable.

This paper introduces a reversible steganography method which combines edge adaptive and Tree Based Parity Check (TBPC) to embed the secret data inside a cover image. To prevent detection, the proposed method identifies the pixel locations of sharp regions for embedding to produce minimum distortion during the embedding process. This approach takes advantage of human eyes characteristic, which are less sensitive to large changes in edge regions and more sensitive to small changes in the smooth regions. Thus, edge regions provide a good carrier for hiding data.

The reminder of this paper is divided as follows. In section 2, we briefly review some well-known Steganography methods. Section 3 describes the data embedding and extraction processes of the proposed hybrid system. Experimental results are presented in section 4. Finally, a conclusion is given in section 5.

2 Related Work

The Least Significant Bit (LSB) replacement is a well-known steganographic method. Using any digital carrier, LSB replaces the n−LSBs of each pixel by n−bits from the secret message [2,6,7,9]. However, due to its simplicity, some steganalysis methods are not only able to discover the presence of the embedded message, but can also estimate its length [10,11].

While human eyes perception is sensitive to slight modifications in smooth areas, it cannot recognize more substantial modifications in edge areas. Several Pixel Value Differencing (PVD) methods, such as [1,12,13,5] have been proposed to improve the embedding rate without introducing obvious visual artefacts. PVD is an edge adaptive technique, in which the number of hidden bits is decided by the difference values between non-overlapping blocks of two consecutive

pixels. More bits of secret message can be embedded in blocks with high difference values, compared to those of small difference values. Accordingly, PVD can provide a larger embedding capacity with more than 2 bits per block.

In [9], Luo et. al. introduced edge adaptive image steganography based on LSB matching revisited to enhance the security compared with the original LSB method. EA-LSBMR divides the cover image into non-overlapping block of equal size ($bz \times bz$), and each block is rotated by random degree to discover the edge pixels in more than one direction. The difference value between two adjacent pixels is computed. If the difference is greater than the threshold value then one bit of the secret data is hidden in each pixel using LSBMR.

Crandall[14] suggested the idea of hiding data based on matrix coding to improve the embedding efficiency. It hides and retrieves the message by utilizing the parity check matrix of a linear code. Fridrich et. al. [15] introduced a steganography method based on linear codes with small dimension, which can achieve high embedding efficiency for only large embedding rates.

Li et. al. [16] proposed a data hiding method called Tree-Based Parity Check (TBPC) to improve the embedding efficiency by reducing the difference between the cover and the stego images. In order to minimize the modifications in the cover pixels, TBPC represents the LSB of the cover pixels using a complete N−ary tree. The method in[16] can be formulated as another specific matrix embedding, which was improved by Hou et. al. [17], where they introduced a majority-vote parity check (MPC) instead of the original matrix embedding. In [18] Liu et. al. introduced an adaptive steganography algorithm based on block complexity and matrix embedding. The embedding strategy sets are defined for seven kinds of image blocks with different complexity. The corresponding embedding strategies are determined by resolving the embedding risk minimization problem. The adaption guarantees that the message bits are mainly embedded into the regions with higher complexity values.
The next section presents our proposed method which preserves higher visual quality of the stego images. It based on an edge adaptive to detect the sharp regions and TBPC to hide the secret data into the cover image. The number of embedded bits at each block is varied based on the difference value between each two adjacent pixels of the block. It introduces minimum possible distortion during the embedding process to prevent discovering the secret data.

3 The Proposed Method

Embedding capacity is one of the major requirements of any steganography methods. However, it is important for steganography methods not to leave any noticeable changes to the human eyes after hiding the secret data. We present a hybrid image steganography method that combines edge adaptive and TBPC. The proposed method utilizes the high contrast regions of an image as embedding locations. It is well known that human eyes cannot discover modifications in the edge areas as they can do in smooth areas. Therefore, the number of hidden bits is based on the difference value between the two pixels of each block. The incorporation of

TBPC leads to a better embedding capacity. Thus, the proposed method combines the strengthes of edge adaptive and TBPC.

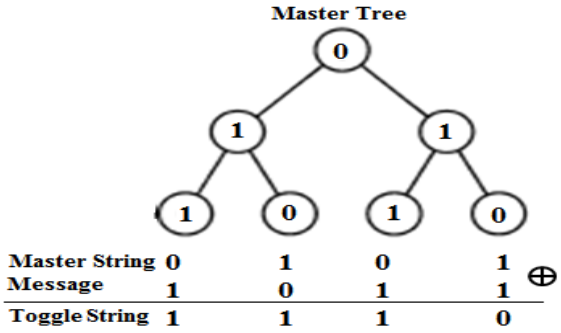
3.1 The Embedding Algorithm

Details of the data embedding process are described below.
Algorithm 1: The Embedding Procedure.
Inputs: Cover image (C) of size $W \times H$, secret message (M).
Output: Stego image (S) of size $W \times H$.

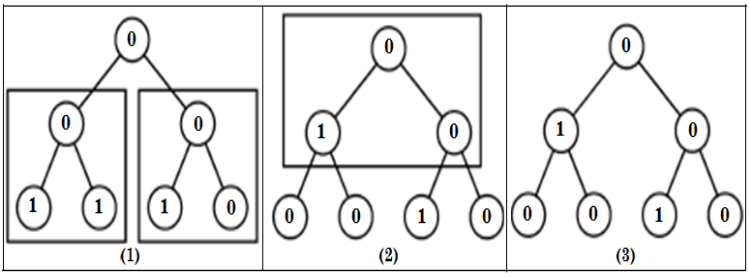
Step 1. Divide the cover image into (1×2) non-overlapping blocks of two adjacent pixels (p_i and p_{i+1}).
Step 2. Compute the absolute difference value between the two adjacent pixels $d_i = |p_i - p_{i+1}|$.
Step 3. Arrange the blocks into six groups as shown in the Table 1. The blocks are sorted in descending order according to the difference value between consecutive pixel pairs, as a measure for region selection. This process can help in minimizing distortion when embedding the data.
Step 4. Construct the TBPC 2−ary tree. n−LSBs from each pixel are used in embedding to enhance the embedding rate.
Step 4.1. A $2-ary$ complete tree called the "master tree" is constructed to represent the n−LSBs from each pixel. Then the nodes in the master tree are filled up with n−LSB level by level, from the root to the leaf nodes.
Step 4.2. Calculate the "master string" by performing a bitwise Exclusive-Or from the root to the leaf nodes in the master tree, as shown in Figure 1(a).
Step 4.3. Perform a bitwise Exclusive-Or between the master string and the message bits to obtain the "toggle string", as shown in Figure 1(a).
Step 4.4. Create a new complete 2−ary tree, called the "toggle tree" in a bottom−up order. In Figure 1(b), the leave nodes are filled up with the toggle string and the rest of the nodes are assigned a value of "0".
Step 4.5. To reduce the number of modifications, the "1s" in the toggle tree should be minimized. Since "1" represents the number of modifications required on the master tree to embed the secret message. Level by level, from the leaf nodes to the root, each parent with its child nodes are flipped if its both children have a value of "1".
Step 4.6. As shown in the Figure 1(c), the stego tree is constructed by performing a bitwise Exclusive-Or between the master tree and the toggle tree.
Step 5. Check the new difference of the pixel pair after the embedding to ensure that the new difference is in the same range of the old difference. If it is not, then it can be corrected by adding or subtracting 2^{n+1}.

Table 1. Range Table

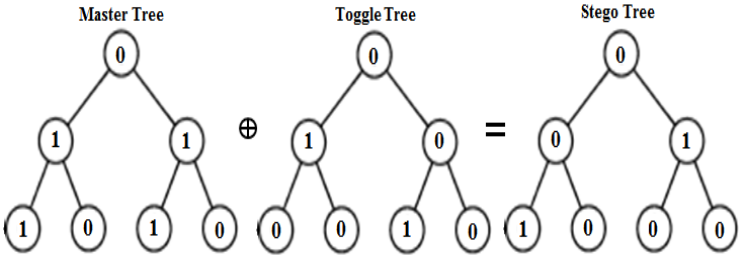
Group	G1	G2	G3	G4	G5
Range	$[0-7]$	$[8-15]$	$[16-31]$	$[32-63]$	$[64-255]$
Number of bits to embed (per pixels)	2	2	3	4	4



(a)



(b)



(c)

Fig. 1. (a)Master String and Toggle String of a 2-ary Master Tree with four leaves, (b.1) Toggle String, (b.2) and (b.3)Construction of a toggle tree and (c) Stego Tree

3.2 The Extraction Algorithm

Details of the data extraction algorithm are as follows:
Algorithm 2: The Extraction Procedure.
Inputs: Stego image (S) of size $W \times H$.
Output: Secret message (M).

Step 1. Divide the stego image into (1×2) non-overlapping blocks of two adjacent pixels (p_i and p_{i+1}).
Step 2. Compute the absolute difference value between the two adjacent pixels $d_i = |p_i - p_{i+1}|$.
Step 3. Arrange the blocks into six groups as shown in the Table 1, and sort them in descending order according to the difference value between the consecutive pixel pairs.
Step 4. Construct the Stego tree from the $n-$LSBs of the stego pixels, which is filled up level by level, from top to bottom and left to right.
Step 5. Perform a bitwise Exclusive-Or from the root to the leaf nodes to retrieve the secret message.

4 Experimental Results

We implemented the proposed method in MatlabR2012b. To evaluate the proposed method, six 256 × 256 gray images were used ("Lena", "Baboon", "Peppers", "Camerman", "House" and "Barbara") as cover images, which are shown in Figure 2. Data capacity is used as one of the evaluation criteria, which is defined as the amount of bits that can be embedded into the cover image. The embedding capacity is computed using Eq. 1.

$$E = \frac{K}{WH}(bpp) \tag{1}$$

where K is the number of the data message bits, while W and H are the width and height of the cover image (both cover and stego images are of the same size. For the considered images, $W = H = 256$).

The visual quality of stego images can be calculated using the Peak Signal-to-Noise Ratio (PSNR), which is calculated as shown in Eq. 2. Higher PSNR indicates better quality.

$$PSNR = 20\log_{10}\left(\frac{255}{MSE}\right)(dB) \tag{2}$$

where MSE is the mean square error between cover and stego images, which is defined as:

$$MSE = \frac{1}{WH}\sum_{i=1}^{W}\sum_{j=1}^{H}(c_{ij} - s_{ij})^2 \tag{3}$$

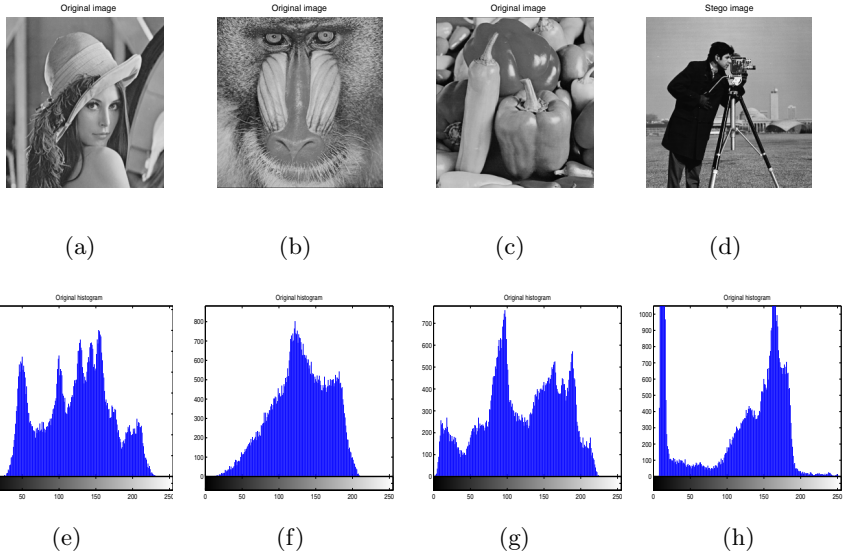


Fig. 2. The Cover Images (a) Lena,(b) Baboon (c) Pepperc, (d) Cameraman and (e) Barbara, (e-h) Corresponding histogram of the cover images

where c_{ij} and s_{ij} are the gray values of pixel (i, j) of the cover and stego images respectively. The disadvantage of the PSNR and MSE is that they simply measure how much change happened between the cover and stego image. However, they are not indicative of how the human visual framework (HVS) essentially would rate the resultant picture quality.

The weighted Peak signal-to-Noise Ratio ($wPSNR$) is an alternate measurement of imperceptibility. It utilizes an extra parameter called Noise Visibility function (NVF). $wPSNR$ is roughly equivalent to PSNR for flat areas because NVF is close to one in smooth regions. However, for regions with sharp contrasts, $wPSNR$ is higher than $PSNR$, because NVF is close to zero for complex regions. Hence, $wPSNR$ attempts to reflect how the HVS perceives images.

$$wPSNR = 10 \log_{10} \left(\frac{\max(C)^2}{\|NVF(S - C)\|^2} \right) (dB) \tag{4}$$

$$NVF = NORM \left\{ \frac{1}{1 + \delta^2} \right\} \tag{5}$$

Where δ is the luminance variance for the 8×8 block and NORM is the normalization function. In this paper, we will use $wPSNR$ as a measure of imperceptibility or quality of the produced stego images.

In order to have a comprehensive comparison, we implemented two versions of PVD and the proposed algorithm. In the first version, we used 1 bit per pixel, i.e., a LSB implementation of the two algorithms. Please note that the original TBPC algorithm was also a LSB-based algorithm. In the second version, we considered

Table 2. Comparisons using 1 bpp between PVD, TBPC and the Proposed Edge Adaptive-TBPC

		1 bpp PVD			TBPC		Adaptive Edge TBPC (1 bpp)	
Image	*Embedding Rate*	30%	50%	80%	30%	50%	30%	50%
Lena	PSNR	54.252	52.515	50.787	57.4206	55.3404	57.287	55.354
	wPSNR	68.476	67.415	64.202	69.023	67.451	71.254	68.007
	SSIM	0.9994	0.9982	0.9969	0.9991	0.9987	0.9994	0.9988
	Avg. Difference	0.1845	0.2901	0.4456	0.1178	0.1942	0.1154	0.1818
Baboon	PSNR	55.535	52.236	50.299	57.398	55.307	57.126	55.138
	wPSNR	94.853	86.584	77.203	81.936	79.555	93.129	81.071
	SSIM	0.9998	0.9993	0.9986	0.9997	0.9995	0.9998	0.9995
	Avg. Difference	0.1818	0.2808	0.4644	0.1184	0.1916	0.1176	0.1871
Cameraman	PSNR	54.683	53.039	51.315	57.399	55.316	57.411	55.317
	wPSNR	69.2814	66.909	62.455	66.963	65.327	70.746	66.011
	SSIM	0.9991	0.9978	0.9962	0.9987	0.9983	0.9991	0.9984
	Avg. Difference	0.1795	0.2808	0.4305	0.1176	0.1912	0.1146	0.1805
Peppers	PSNR	54.244	52.495	50.792	57.427	55.39	57.333	55.388
	wPSNR	72.752	67.635	65.116	70.258	68.497	71.591	68.849
	SSIM	0.9992	0.9984	0.9969	0.9993	0.9988	0.9994	0.9988
	Avg. Difference	0.1852	0.2915	0.4468	0.1176	0.1880	0.1148	0.1814
Barbara	PSNR	54.438	52.423	50.676	57.385	55.352	57.144	55.221
	wPSNR	68.860	66.826	65.886	70.313	68.426	72.898	69.314
	SSIM	0.9994	0.9989	0.9979	0.9995	0.9991	0.9996	0.9992
	Avg. Difference	0.1833	0.2925	0.4472	0.1187	0.1896	0.1179	0.1954
House	PSNR	54.745	52.907	51.031	57.366	55.301	57.44	55.41
	wPSNR	63.541	61.918	61.231	65.574	65.05	65.88	65.134
	SSIM	0.9983	0.9972	0.9959	0.9988	0.9983	0.9989	0.9984
	Avg. Difference	0.1786	0.2819	0.4372	0.1192	0.1919	0.1134	0.1815

n bits per pixel, and hence the known PVD algorithm and our proposed Edge Adaptive-TBPC.

Table 2 presents the obtained values of $wPSNR$ and average difference for the 1 bpp implementation of the three algorithms. These results indicate that the proposed Edge Adaptive-TBPC produced higher quality stego images compared to those obtained using the other two methods. The second best algorithm is found to be the TBPC, as it has a higher embedding efficiency due to it coding capability. However, the embedding rate of the original TBPC and Edge Adaptive-TBPC cannot exceed 50% of the cover image pixels. According to Eq. 5, sharp regions is the appropraite embedding locations because weighting of the modification in high contrast regions is smaller than in smooth regions. The aim of using adaptive edge is to maintain the texture of the LSB plane. As shown in

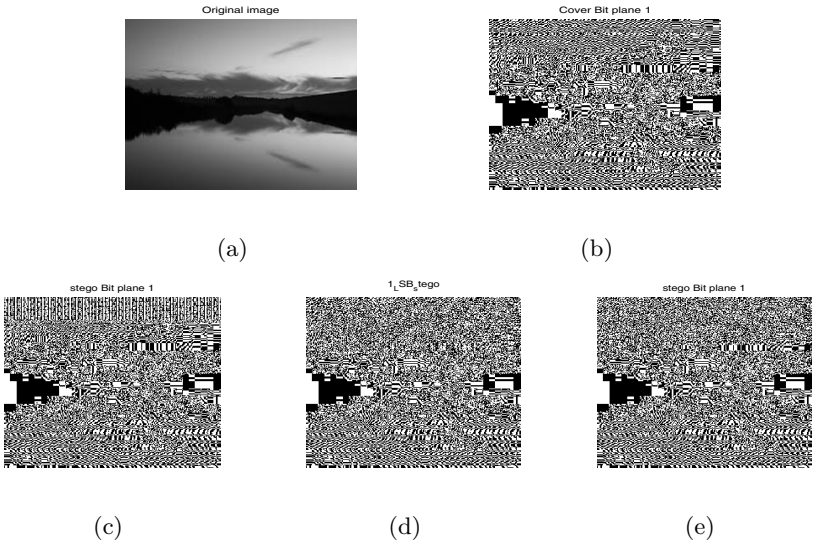


Fig. 3. (a) Cover image (b) LSB plane of the cover image (c−e) LSB planes of the stego images using using 1 bpp PVD, TBPC and Edge Adaptive-TBPC respectively with 20% embedding rate

Figure 3, any change in smooth area of the image may affect the LSB value of the pixels.

The second version of the proposed steganographic (n bits) begins with embedding the secret message in the sharp regions first according to the size of the secret data. This means the distortions will be less detectable by HVS because the modifications in edge areas. In addition, number of embedding bits on each block is based on the difference value between the adjacent pixels. However, PVD embeds the secret message in sequential order with different number of embedding bits on each block. The proposed method can reach 100% embedding rate with high visual quality.

Figures 4(a)−4(d) show the stego images when the embedding rate is 50% and the correspondence stego histograms are shown in Figure 4(e)−4(h). The produced images gave a high degree of similarity where it is quite hard to find visual differences between the cover and stego images. Figures 4(i)−4(l) show the stego images when the embedding rate is 80%. The stego histograms shown in Figure 4(m)−4(p). It is clear that the quality of the stego images decreased slightly by increasing the embedding rate.

The result of our proposed method and the original PVD are summarized in table 3. The proposed Edge Adaptive-TBPC achieved noticeably better wPSNR and average difference results compared to the PVD algorithm for all considered six images, which indicate that better imperceptibility for the same embedding capacity.

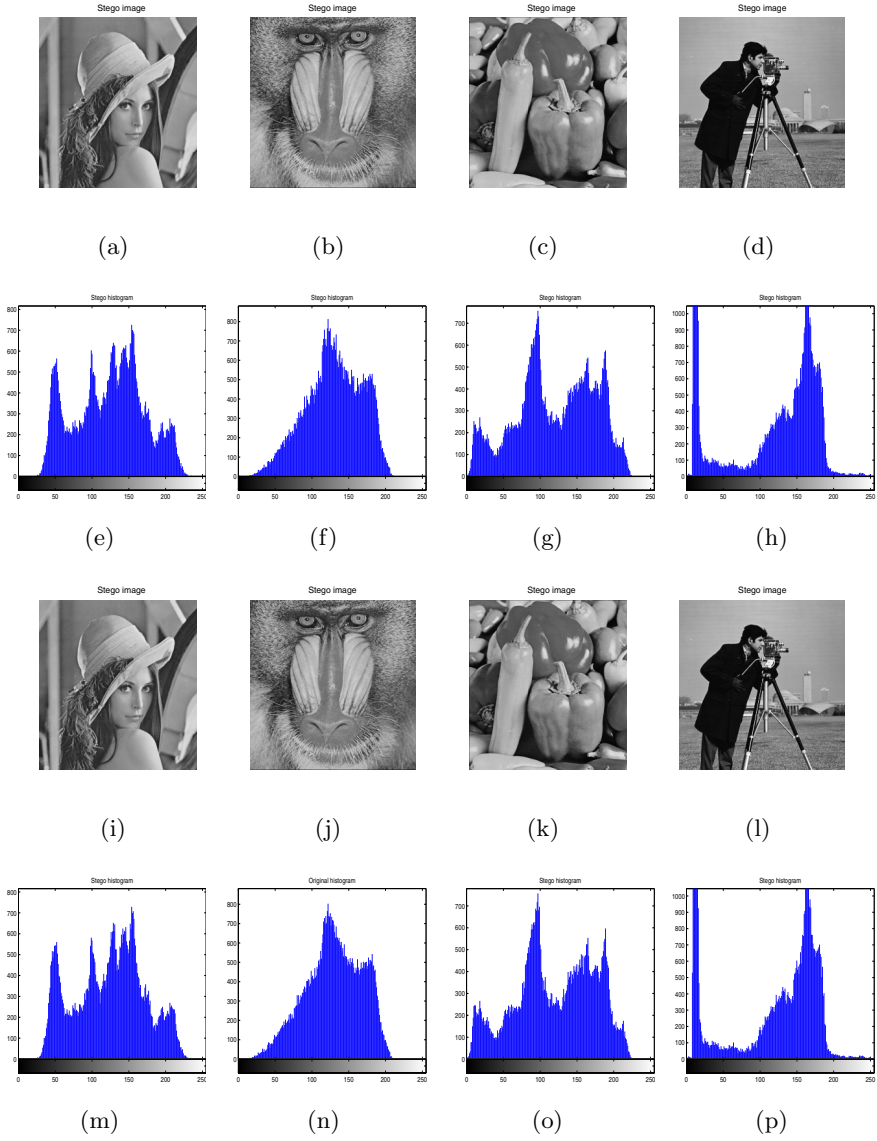


Fig. 4. (a − d) The Stego Images with embedding rate 50%, (e − h)) The histogram of the stego image with embedding rate 50%, (i − l) The Stego image with embedding rate 80% and (m − p) The histogram of the stego image with embedding rate 80%

Table 3. Comparisons of the visual quality and the embedding rate

		PVD			Proposed		
Image	*Embedding Rate*	30%	50%	80%	30%	50%	80%
Lena	wPSNR	63.5569	62.4805	60.8132	89.9022	73.8984	67.5874
	SSIM	0.9974	0.9961	0.9947	0.9979	0.9971	0.9967
	Avg. Difference	0.1892	0.3223	0.5315	0.1887	0.2561	0.4082
Baboon	wPSNR	89.1094	84.3872	76.0297	92.3605	88.5796	83.9867
	SSIM	0.9992	0.9987	0.9979	0.9974	0.9962	0.9953
	Avg. Difference	0.2019	0.3896	6303	0.1801	0.3217	0.5392
Cameraman	wPSNR	65.3993	62.5937	59.6606	87.4737	73.2688	66.8596
	SSIM	0.9967	0.9948	0.9929	0.9982	0.9976	0.9967
	Avg. Difference	0.2189	0.3937	0.5943	0.2158	0.3512	0.5661
Peppers	wPSNR	65.6801	63.8660	62.3860	87.5087	86.8021	72.0155
	SSIM	0.9981	0.9970	0.9959	0.9986	0.9979	0.9973
	Avg. Difference	0.2304	0.3814	0.5921	0.1924	0.3384	0.5478
Barbara	wPSNR	66.3192	64.1648	62.7694	91.3775	86.37	70.5923
	SSIM	0.9974	0.9963	0.9955	0.9986	0.9977	0.9963
	Avg. Difference	0.1776	0.3026	0.4892	0.1517	0.2822	0.4650
House	wPSNR	58.9912	57.8211	57.2199	67.5922	63.7642	62.8509
	SSIM	0.9964	0.9948	0.9929	0.9982	0.9973	0.9963
	Avg. Difference	0.1776	0.3020	0.4892	0.1517	0.2822	0.4050

5 Conclusion

This paper has introduced an image steganography method that combines the edge adaptive and TBPC algorithms to enhance the payload and imperceptibility of the stego image. Our method introduced minimum possible distortion during the embedding process to minimize the probability of discovering the secret message data from unauthorized users. We embed four bits into seven pixel bits. Due to the incorporation of an efficient coding mechanism, the probability of modifying pixel bits is 0.285. To further minimize the visual artefacts, the selection of embedding blocks is decided based on the the length of the secret message. Levels of sharp regions are categorized. The proposed method hides the secret data into the sharpest edge regions first (highest level). Then based on the message length, it moves down to the lower edge levels. Experimental results obtained by applying the proposed method to different images indicate that the proposed steganography method outperformed both of the original PVD and TBPC algorithms.

References

1. Wu, D.C., Tsai, W.-H.: A steganographic method for images by pixel-value differencing. Pattern Recognition Letters 24(9), 1613–1626 (2003)
2. Verma, N.: Review of steganography techniques. In: Proceedings of the International Conference & Workshop on Emerging Trends in Technology, pp. 990–993. ACM (2011)
3. Bailey, K., Curran, K.: An evaluation of image based steganography methods. Multimedia Tools and Applications 30(1), 55–88 (2006)
4. Sokół, B., Yarmolik, V.: Cryptography and steganography: teaching experience. In: Enhanced methods in computer security, biometric and artificial intelligence systems, pp. 83–92. Springer (2005)
5. Mukherjee, R., Ghoshal, N.: Steganography based visual cryptography (SBVC). In: Satapathy, S.C., Udgata, S.K., Biswal, B.N. (eds.) Proceedings of Int. Conf. on Front. of Intell. Comput. AISC, vol. 199, pp. 559–566. Springer, Heidelberg (2013)
6. Cheddad, A., Condell, J., Curran, K., Kevitt, P.M.: Digital image steganography: Survey and analysis of current methods. Signal Processing 90(3), 727–752 (2010)
7. Shrivastava, G., Pandey, A., Sharma, K.: Steganography and its technique: Technical overview. In: Proceedings of the Third International Conference on Trends in Information, Telecommunication and Computing, pp. 615–620. Springer (2013)
8. Johnson, N.F., Jajodia, S.: Exploring steganography: Seeing the unseen
9. Luo, W., Huang, F., Huang, J.: Edge adaptive image steganography based on lsb matching revisited. IEEE Transactions on Information Forensics and Security 5(2), 201–214 (2010)
10. Fridrich, J., Goljan, M.: On estimation of secret message length in lsb steganography in spatial domain. In: Electronic Imaging 2004, pp. 23–34. International Society for Optics and Photonics (2004)
11. Fridrich, J., Kodovský, J.: Steganalysis of LSB replacement using parity-aware features. In: Kirchner, M., Ghosal, D. (eds.) IH 2012. LNCS, vol. 7692, pp. 31–45. Springer, Heidelberg (2013)
12. Luo, W., Huang, F., Huang, J.: A more secure steganography based on adaptive pixel-value differencing scheme. Multimedia Tools and Applications 52(2-3), 407–430 (2011)
13. Agrawal, S.S., Samant, R.M.: Data hiding in gray-scale images using pixel value differencing. In: Technology Systems and Management, pp. 27–33. Springer (2011)
14. Crandall, R.: Some notes on steganography. Posted on steganography mailing list (1998)
15. Fridrich, J., Soukal, D.: Matrix embedding for large payloads. In: Electronic Imaging 2006, pp. 60721W–60721W. International Society for Optics and Photonics (2006)
16. Li, R.Y., Au, O.C., Lai, K.K., Yuk, C.K., Lam, S.Y.: Data hiding with tree based parity check. In: 2007 IEEE International Conference on Multimedia and Expo, pp. 635–638. IEEE (2007)
17. Hou, C.L., Lu, C., Tsai, S.C., Tzeng, W.G.: An optimal data hiding scheme with tree-based parity check. IEEE Transactions on Image Processing 20(3), 880–886 (2011)
18. Liu, G., Liu, W., Dai, Y., Lian, S.: Adaptive steganography based on block complexity and matrix embedding. Multimedia Systems 20(2), 227–238 (2014)