Jestr

Journal of Engineering Science and Technology Review 10 (5) (2017) 61 - 67

Research Article

JOURNAL OF
Engineering Science and Technology Review

www.jestr.org

Distortionless Data Hiding for Encrypted Images: A DCT Approach

Subash Nemani[1,*], Jayachandra Prasad Talari[2] and Sumalatha Vangala[3]

[1]Department of ECE, RGMCET, JNTUA College of Engineering, Ananthapuramu, Andhra Pradesh-515002, India
[2] Department of ECE, RGMCET, Nandyal, Kurnool Dist, A.P
[3]Department of ECE, JNTUA College of Engineering, Ananthapuramu, AP

Received 21 July 2017; Accepted 15 September 2017

Abstract

An enhanced image quality for Distortionless Data Hiding (DDH) by Discrete Cosine Transform in images encrypted was implemented in this work. The DDH is commonly called reversible data hiding (RDH), referred as technique recovering input image with no distortion from stego image by separating the concealed information. Information can be implanted into the encrypted pictures by creating vacant space after encryption or before encryption. The former has the disadvantage of image degradation by data separation as well as on image recovery. To overcome the disadvantage, creating a vacant space before encryption such that it can attain real reversibility i.e. information can be separated from the cover picture with no degradation. Experiments result in a larger amount of data up to 1,31,072 bits embedded with better PSNR value in comparison to existing technique.

Keywords: Compression, Distortionless Data Hiding, DCT, Image Encryption, Steganography.

1. Introduction

In the present era, with the speedy advancement of technology, an increasing number of the amount of data and snapshots are available on the net. So there is a necessity to provide confidentiality, integrity and authenticity for data/images, one such method is Steganography. It hides information, photos, audio or video formats within another message, photograph, audio or video document. One of the important applications of steganography is Reversible Data Hiding (RDH). The data hiding method intent to insert covert bits within cover medium by varying the least significant bits for copyright security or secret transmission. The DDH scheme will be estimated using the three factors:

1. Embedding capacity: It refers to the number of bits hidden in cover medium. It is denoted by bits per pixel (bpp).
2. Imperceptibility: It refers to inability of human eyes to distinguish the presence of the concealed information directly. The amount of distortion is calculated using peak signal-to-noise ratio (PSNR).
3. Security: Is referred as impossibility in successful attacks to detect hidden information.

In a realistic scenario, many researchers in state-of-the-art have developed several RDH techniques. They are broadly classified into four categories: 1) Difference Expansion (DE) 2) Histogram Shifting (HS) schemes 3) Prediction Methods and 4) Interpolation Algorithms. In 2003 Tian introduced a DE technique [1] to obtain high capacity, low distortion RDH by calculating the differences of neighboring pixel values and estimating the embedding area i.e. identifying the insignificant bits and insert data into it. The main drawback is the size of the binary map and lack of embedding capacity control capability. Another capable approach is Histogram shift [3] (HS), which finds the zero and peak points and make minor modifications to pixel gray scale values to embed data results in higher capacity with better PSNR. The disadvantage is prior knowledge of histogram minimum and maximum values should be known at the receiver in order to extract the data from cover medium. Hwang *et al.* [4] solved this problem by proposing a new RDH scheme to store the reversible data based on a location map by selecting the minimum point to be recorded and change one bit position in it to increase the embedding capacity.

Additionally, the prediction-based method is another technique of RDH. The state-of-art methods [2, 4] mainly focus on location map which are large and could not embed more data into the cover medium. Sachnev *et al.* [5] solved this problem by proposing a rhombus prediction scheme with high embedding size. Azzoni [6] has implemented a novel multi layer structure RDH scheme by combining the difference expansion and prediction error, and achieved high embedding capacity by preserving the image quality. The drawbacks with this method are, the correlation between the pixels is insignificant and Complexity of the prediction is based on characteristics of the image.

In recent times, interpolation algorithms solved the drawbacks of the prediction based method and improved capacity and image quality in terms of payload and Imperceptibility. Interpolation is the technique of obtaining an improved image resolution from the degraded-resolution images. Jung *et al.* [7] implemented a novel interpolation scheme for hiding data by scaling-up neighbor mean interpolation (NMI) to enhance embedding capacity up to 2,00,868 bits with better image quality, i.e. high PSNR. The drawback with this method is the pixel values are

[*] E-mail address: nemani.470@gmail.com

ISSN: 1791-2377
doi:10.25103/jestr.105.07

recalculated before the data is extracted from the cover image. Luo *et al.* [8] suggested error expansion method with large capacity and less distortion. This technique is efficient because interpolation-errors can easily decor relate pixels and overhead information. Jan *et al.* [9] recommended a new way of pixel calculation, which results in improved PSNR compared, to existing methods. Chang *et al.* proposed the enhanced neighbor mean interpolation [10] (ENMI), which uses a multistage embedding method to achieve a higher payload with the embedding capacity of 2, 47,095 bits which is higher than Jung *et al*.

Later providing security for images has also become a challenging task in RDH. One way of attaining security is through encryption as it converts the relevant and significant content to inconceivable form. In encrypted images, data can be hidden in two approaches: one way is encrypting the input image with an encryption key by creating a vacant space to hide the data called as Vacating room after encryption [15-17]. The Second way is creating some vacant space before encryption, then encrypting the input image and hiding information in the space created. Zhang [15] proposed a method for encrypted images by first encrypting input image with encryption key later by data concealing key information hider hides extra data into the images encrypted and again with the encryption key inserted data has been extracted. The input image will be retrieved using data hiding key which is free from errors. The drawback to the Zhang's method is it fails to calculate the smoothness of each block and pixel correlation for the neighboring blocks at the borders is not considered. Chen *et al.* [16] suggested modifications to overcome the drawback in Zhang's method, by smoothing the border of recovered blocks using side match scheme. The advantage with this method is it reduces the error rate for smaller block sizes when compared to the Zhang's method.

In this paper we proposed DDH for images encrypted by compressing the original image with DCT and then creating the vacant space before encryption and conceal the compressed image with LSB Positions for embedding information. Here apart from the extraction of information from the encrypted images we also provide excellent performance (in PSNR) and computational complexity when compared to existing techniques.

The remaining paper is categorized as: In Section 2, we explain different least significant bit (LSB) encoding methods. In Section 3, Proposed Method is explained. Results and Analysis are presented in Section 4. At the end, we conclude with conclusions in Section 5.

2. Related Work

Here, we briefly describe the basic concept of the least significant bit (LSB)-based reversible data hiding techniques. LSB steganography is the most popular in spatial domain among all embedding schemes. The different types of the LSB are: LSB Replacement (LSBR), LSB Matching (LSBM) and N-bit LSB Encoding.

2.1 LSB Replacement (LSBR)

It is an outstanding method for steganography, where we replace the LSB's of the original image with concealed information bits to get stego image [11]. The merits of LSBR are: high embedding capacity, good imperceptibility and ease of implementation. If the original image pixels contain even values, then the bits are either unchanged or incremented by 1, in case of odd values the bits are unmodified or decremented by 1. Due to the asymmetry, bits are detected easily even for lower embedding rates [12]. To overcome this asymmetry problem, change of the least significant bits is randomized.

2.2 LSB Matching (LSBM)

It is the modified form of LSBR. If LSB's of the input image are not equivalent to the secret message bit, one is randomly inserted/deleted from the original pixel value and is known as ±1 embedding [4]. The advantage of LSB matching is, the bits are randomly changed to avoid asymmetry so that it is difficult to detect. Further to improve the LSB matching, Mielikainen proposed a scheme based on the fewer pixel modifications to the cover image. It reduces the expected changes in pixel up to 0.375 [13] at embedding rate of 1bpp i.e. this method has improved the embedding efficiency when compared to the existing methods.

2.3 N-bit LSB Encoding

This method is used for inserting secret or private information in digital images with N=2, 3, 4 etc. In LSB algorithm, information is concealed in the least values in a binary number of the cover image which is not visible when viewed with the human eye. The covert information which is inserted in the cover image can be an image or a text. As the type and size of secret message vary, the payload varies accordingly. In order to increase the security, secret information bits are encrypted and then inserted into the LSB's of the cover.

3. Proposed Method

As found from the existing methods of RDH in encrypted domain, i.e. vacating room after encryption (VRAE) has some limitations that, it is not possible to recover input image and extract concealed data, i.e. free from errors in all cases and errors will occur due to increase in embedding capacity beyond a limit at the data extraction end. To overcome these limitations, the vacant space can be created after compression at the sender side which further increases the PSNR.

From Figure 1, we compress the original image using DCT and then create sufficient space in the image compressed thereafter transforming to encrypted form using encryption keys. Proposed method consists of four stages: 1) Compression using DCT 2) Generating encrypted images 3) Hiding data in images encrypted 4) Image recovery after data retrieval.

3.1 Compression using DCT

Transformation of image from one domain, i.e. spatial to another, i.e. frequency domain is referred as DCT. Main purpose of DCT is it is applied for image transformation and compression. It is a real transform which results in better computational efficiency.

Algorithm for image compression using DCT:

Two steps are performed for compression:
A) Encoding and B) Decoding.

A) Encoding:

The following are the steps followed for encoding or compressing image:

1. Divide the input image into N*N segments of horizontal and vertical pixels.
2. DCT is applied to each block, in a Zigzag manner (i.e. left to right, from bottom to top) with increasing frequency.
3. Apply Quantization for each block element to be compressed.
4. The energy is stored in all blocks compressed in free space.
5. Repeat Step 2 to step 4 till all the blocks are completed.

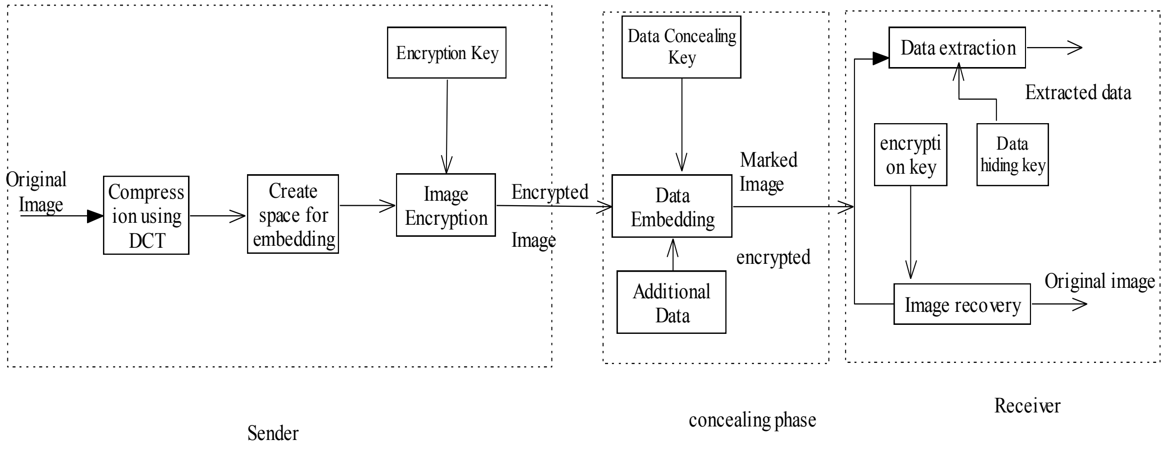


Fig. 1. Framework of Proposed Method.

B) Decoding:

Decoding is the exact reverse process of encoding. The following are the steps to be performed:

1. Load the image compressed.
2. Divide the pixels of the image compressed into N*N blocks of the same order of N as in encoding.
3. By applying reverse process of quantization, each block is de-quantized.
4. Now apply inverse DCT on each block and then join all these blocks to recover the input image.

3.2 Generating Encrypted Images

In order to get an encrypted image, we divide the compressed image into two regions X and Y as shown in Figure 2, such that the LSB's of X are reversible embedded into Y using DDH algorithm. The LSB planes of X are used to store information bits.

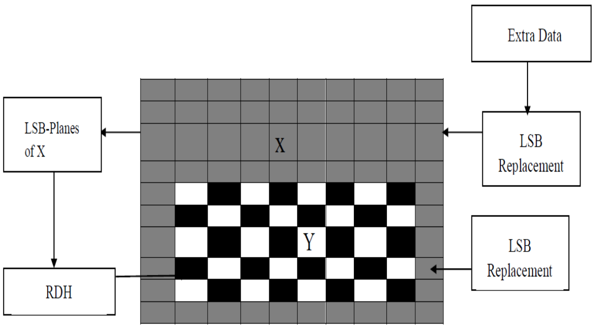


Fig. 2. Representation of partitioning Image.

Consider the compressed image I is 8-bit gray scale image of U x V dimension with pixels ranging from 0 to 255. Let the embedded message size is denoted as E and the content owner extracts many overlapping blocks based on this embedded message size. Each block contains u rows with $u = E/V$ and total blocks are calculated as $v = U - u + 1$. For an image along the rows, every pixel value is overlapped by previous or neighboring blocks. In order to find the smoothness of individual block with neighboring pixel elements a mathematical equation is given as:

$$f = \sum_{u=2}^{m}\sum_{v=2}^{N-1}\left| I_{u,v} - \frac{I_{u+1,v} + I_{u-1,v} + I_{u,v+1} + I_{u,v-1}}{4} \right| \quad (1)$$

A high value of f leads to the more complex textures in the blocks. For that reason the content owner selects the higher values of f for particular blocks to be in X and lower texture values in rest part of Y as shown in Figure 2. To embed the information on to the partitioned image, content owner chooses two LSB-planes of X into Y because in this paper, we are compressing the input image and then creating the vacant space for that compressed image as a result the single or higher LSB-planes of X causes decrease in size with decrease in PSNR. Therefore, in this paper, we are taking only two LSB-planes of X and compare these results with the state-of-art methods experimentally in the next section.

In order to insert the LSB-planes of X into Y we follow self-embedding process. As we know that the region used for embedding is Y. The pixels in the remaining portion of Y are grouped as white and black regions with m and n as coordinates satisfying: (m + n) mod2 = 0 for white pixels and (m + n) mod2=1 for black pixels from Figure 2. The white pixels $Y_{m,n}$ in each block are estimated using interpolation with surrounding four black pixels as:

$$Y_{m,n}' = w_1 Y_{m-1,n} + w_2 Y_{m+1,n} + w_3 Y_{m,n-1} + w_4 Y_{m,n+1} \quad (2)$$

Here weight w_i for $1 \le w_i \le 4$ are calculated as proposed in [8]. The rest of the process of self embedding is same as in [14]. After all the self embedding process is done, we get an image denoted by R this image can be encrypted using stream cipher and that encrypted image is denoted by Z. Let $R_{m,n}$ be a gray scale image denoted by

8-bits, i.e. $R_{m,n,0}, R_{m,n,1}.....R_{m,n,7}$ ranging from 0 to 255. Here the pixel position is indicated by (m, n) whose gray values are $p_{m,n}$. Therefore

$$R_{m,n,k} = \left\lfloor \frac{p_{m,n}}{2^k} \right\rfloor \bmod 2 \qquad (3)$$

Where $0 \le k \le 7$

$$p_{m,n} = \sum_{k=0}^{7} R_{m,n,k}.2^k \qquad (4)$$

By using Exclusive-OR operation, encrypted bits $Z_{m,n}$ can be calculated as:

$$Z_{m,n} = R_{m,n} \oplus r_{m,n} \qquad (5)$$

With the help of a stream cipher, $r_{m,n}$ will be created from the encryption key. Once the image is encrypted, either the information hider or any other potential attacker may not acquire the original content in the absence of encryption keys, as a result image sender is secured with any malicious attacks.

3.3 Hiding Data in Images Encrypted

As the encrypted image is being accessed by the data hider, he will insert additional messages to it by changing a little amount of the data encrypted though he has no access to the original image. Figure 3 shows a Simple block representation of data hiding process.

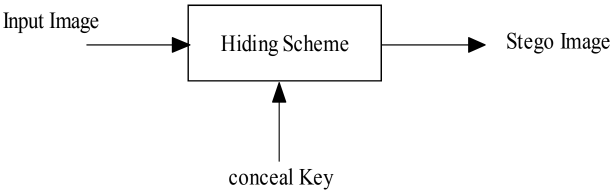


Fig. 3. Information hiding scheme

At First encrypted image will be segmented by data hider into smaller blocks of size q x q such that it satisfies the conditions:

$$(u-1).q + 1 \le m \le u.q \qquad (6)$$

$$(v-1).q + 1 \le n \le v.q \qquad (7)$$

Where u and v are positive integers. By doing in this manner at least one extra bit is used to store in every block. Now divide q^2 pixels of each block into two sets q_0 and q_1 pseudo-randomly based on data-hiding key. If binary bit 0 is to be inserted, then change pixels in q_0 by three LSB positions.

$$Z_{m,n,k}{}' = \overline{Z}_{m,n,k}, (m,n) \in q_0; for 0 \le k \le 2 \qquad (8)$$

Similarly, if 1 is to be inserted then change pixels in q_1 by three LSB positions.

$$Z_{m,n,k}{}' = \overline{Z}_{m,n,k}, (m,n) \in q_1; for 0 \le k \le 2 \qquad (9)$$

Rest of the bits encrypted remains unchanged.

3.4 Image recovery after data retrieval

We can extract data from encrypted images or from decrypted images. In former case the receiver checks the server using data hiding key and decrypts the least significant bits from encrypted images denoted as Z and extract hidden information. This process is more secure than previously because the operation is based on encryption. In the later case the owner first decrypts the image, and then based on the requirement extracts the information from the decrypted images. All the changes in encrypted images are made in the encryption side.

4. Results and Analysis

The results of the proposed method are presented by taking the reference images as: “Lena”, “Airplane”, “Barbara”, “Baboon”, “Peppers” and “Boat” [18] from database were shown in Figure 4.

Fig. 4. Cover Images used in experiment

The size of all the images is 512 x 512. As from Figure 2, we have divided the original image into X and Y, in which size of X can be determined by the length of the embedded message and the number of LSB-planes, embedded reversible in Y. We can enlarge the size of Y by increasing the embedding capacity and embedding process is applied only once to Y to accommodate the LSB-planes of X such that the distortion can be reduced. The parameter used to measure the distortion is PSNR in decibels. Higher PSNR value leads to better visual quality in images. PSNR value can be computed by first calculating the mean square error as:

$$MSE = \frac{1}{MXN} \sum_{M=0}^{M-1} \sum_{N=0}^{N-1} \left[(I_1 - I_1')^2 \right] \qquad (10)$$

Where

- Original Image,
- $I_1^{'}$ - Stego image and
- M, N - No. of Rows and columns.

PSNR is now computed as:

$$PSNR = 20\log_{10}\left(\frac{255}{MSE}\right) \quad (11)$$

In the proposed method we measure embedding rate (ER) as the amount of concealed information per pixel denoted as bits per pixel (bpp). The encrypted, data embedded at 0.1bpp and recovered image of Lena are given in Figure 5.

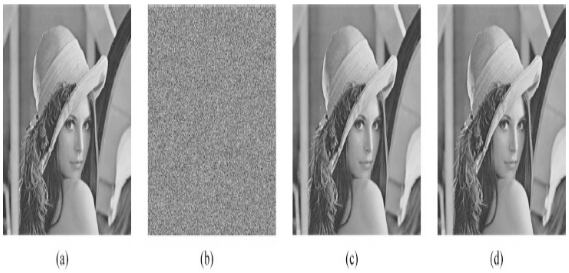


Fig. 5. a) Input Image, b) encrypted version, c) data embedded at embedding rate 0.1bpp, d) Image recovered.

Unlike like the previous methods our method is also free from errors to all the images tested and the image quality has been improved to 3-7dB. PSNR versus embedding rates for different images are presented in Table.1. Plot for embedding rate versus PSNR for various images are shown in Figure 6 (a) - (f). Average PSNR between existing method [14] and the proposed method are compared from Table 2 for various images and observed that its value is ≥ 44.5 dB, average computational time is reduced and the bar graph representations are shown in Figure 7 and Figure 8.

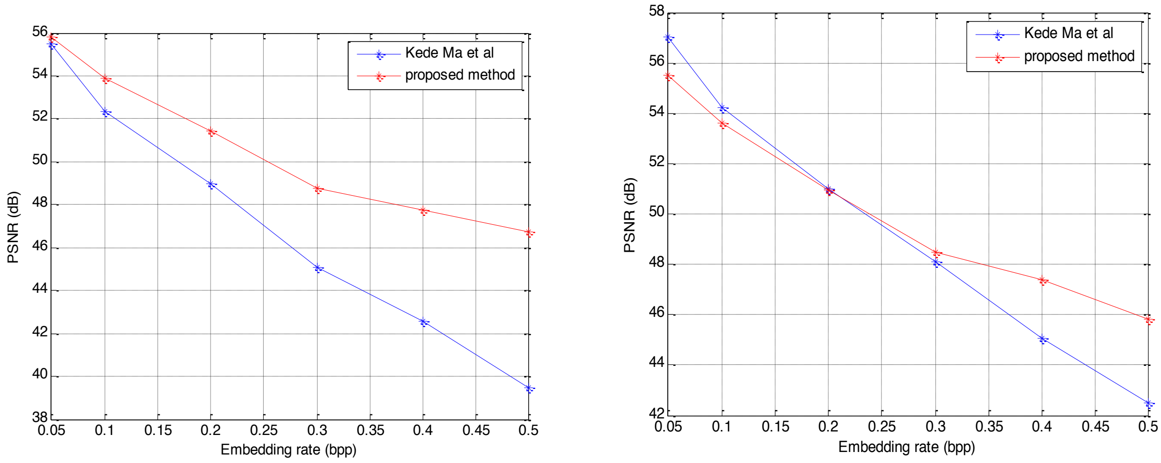
Table 1. PSNR versus embedding rates for different images

Original	***Embedding Rate (bpp)***					
Image PSNR (dB)	0.05	0.1	0.2	0.3	0.4	0.5
Lena	55.8	53.91	51.41	48.78	47.75	46.75
Airplane	55.53	53.63	50.95	48.49	47.39	45.84
Barbara	52.81	49.86	45.32	42.48	39.59	37.24
Baboon	53.24	50.05	45.58	42.44	39.75	37.29
Peppers	54.39	52.01	49.39	46.18	43.98	41.44
Boat	55.07	52.87	50.42	47.93	46.29	44.43

Table 2. Average PSNR and computational time Comparison between Proposed Method & Existing Method

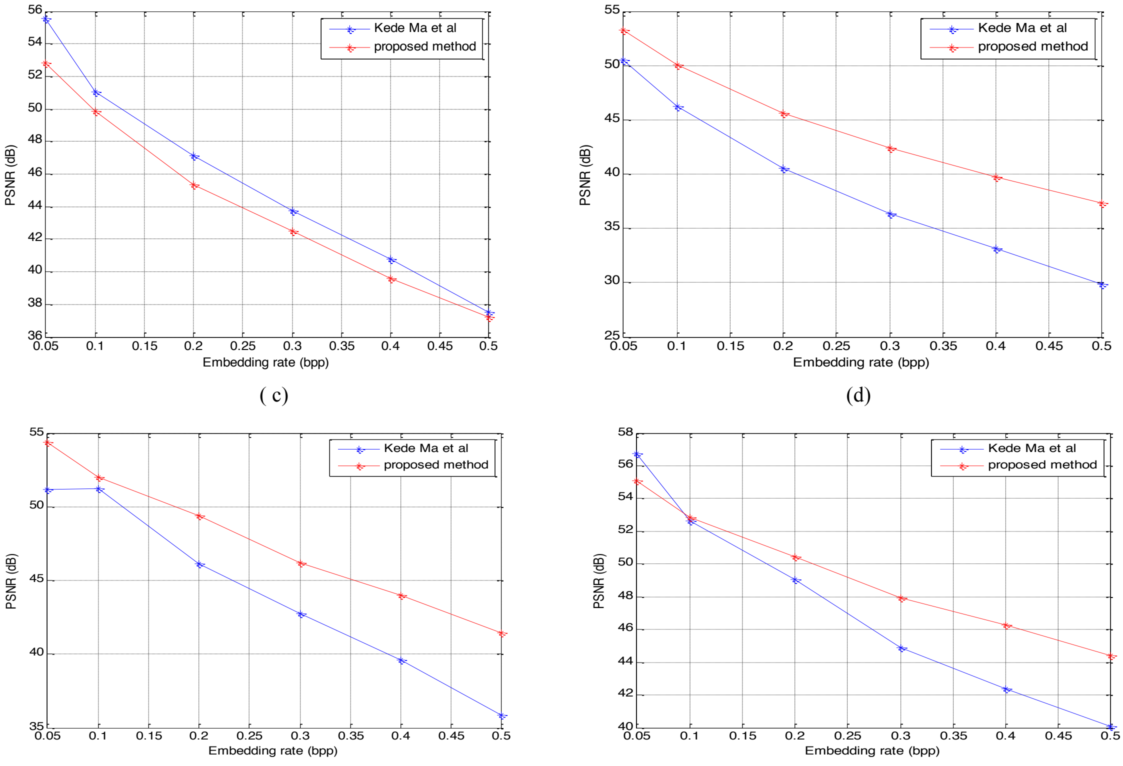
Cover Image	Average PSNR (dB) (Existing Method)	Average PSNR (dB) (Proposed Method)	Average Computational Time (Sec) (Existing Method)	Average Computational Time (Sec) (Proposed Method)
Lena	47.31	50.73	7.24	3.9
Airplane	49.64	50.30	5.96	5.18
Barbara	45.94	44.55	8.5	9.7
Baboon	39.44	44.72	18	10.44
Peppers	44.45	47.89	9.46	5.74
Boat	47.62	49.50	7.24	4.37

Table 2 shows that the average image quality has been improved for all the cover images than the existing method except for Barbara. The reason is Barbara image contains more rough regions so that when we extract data from the image and recovered the image we get low PSNR when compared to the state-of-art methods which is further need to be investigated to improve the quality of that image.



(a)

(b)



(c) (d)

(e) (f)

Fig. 6. Embedding rate versus PSNR for test images a) Lena b) Airplane c) Barbara d) Baboon e) Peppers f) Boat

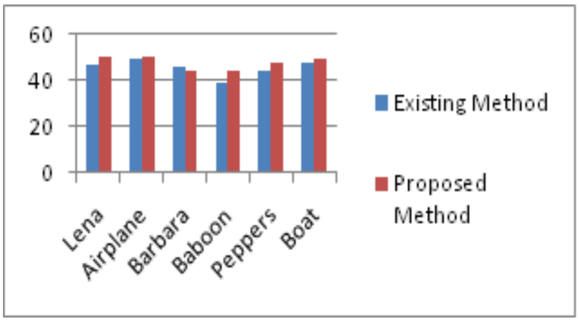


Fig. 7. Average PSNR values for Different images

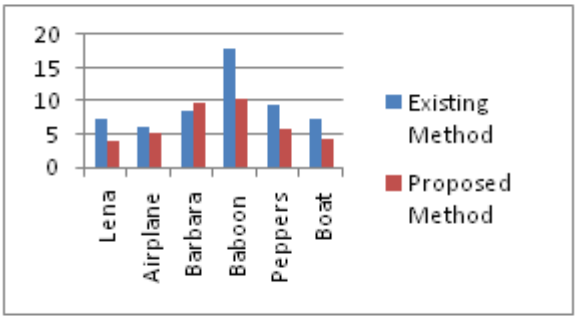


Fig. 8. Average Computational Time for different images

.

From the results it is observed that smaller the embedding rate better is the visual quality in terms of PSNR and further, we can notice for the cover image with smooth regions perform better in the extraction of the data and image recovery. It has been observed that the PSNR is greater than 37.24 dB for all the embedding rates in the proposed method. In Figure 6 red line indicates the method proposed with PSNR 3-7dB greater than state-of-art method. The advantages with the method proposed are 1) It reduces the average time required 2) Average PSNR values are high at various embedding rates tested from the database when compared to the existing method which indicates a wide range of applications in a practical scenario.

4. Conclusions

In this work, we have proposed a method by applying a compression technique using DCT to an input image and then creating a vacant space before encryption and achieved better performance in two aspects: one is in terms of higher PSNR at various embedding rates and the second in terms of low computational time. Our method achieves reversibility i.e., extracting data/messages from a stego image without losing its visual quality and improved the image quality with PSNR ≥ 37.24 dB with high embedding capacity up to 1, 31,072 bits.

References

1. Tian.J, Reversible data embedding using a difference expansion, IEEE Trans.Circuits Syst. Video Technol, Vol. 13, Aug. 2003, pp. 890-896.
2. Ni.Z, Shi.Y.N, Ansari and Wei.S, Reversible data hiding, IEEE Trans.Circuits Syst. Video Technol, Vol. 16, Mar. 2006, pp. 354-362.
3. Wen-Chung Kuo, Dong-Jin Jiang and Yu-Chih Huang, Reversible Data Hiding Based on Histogram, Springer-Verlag Berlin Heidelberg , 2007, pp.1152-1161.
4. Hwang.J, Kim.J and Choi.J, A reversible watermarking based on histogram shifting, Springer-Verlagin, 2006, pp. 348-361.
5. Sachnev.V, Kim.H. J, Nam.J, Suresh.S and Shi, Y. Q, Reversible watermarking algorithm using sorting and prediction, IEEE Trans.Circuits Syst. Video Technol, Vol. 19, Jul. 2009, pp. 989-999.
6. Azzoni.M, Boato.G.M, Carli and Egiazarian.K, Reversible watermarking using prediction and difference expansion, 2nd European workshop on visual information processing, Jul. 2010.
7. Jung.K. H and Yoo.K. Y, Data hiding methods using image interpolation, Computer standards and interfaces, Vol. 31, 2009, pp. 465-470.
8. Lixin Luo, Zhenyong Chen, Ming Chen, Xiao Zeng, Zhang Xiong, Reversible image watermarking using interpolation technique, IEEE Trans. Information forensics and security, Vol. 5, Mar. 2010, pp. 187-193.
9. Sen-Ren Jan, Steen J. Hsu, Chuan-Feng Chiu and shu-lin Chang, An improved data hiding method using image interpolation, seventh international conference on intelligent information hiding and multimedia signal processing, 2011.
10. Chang.Y. T, Huang.C. T, Lee.C. F and Wang. S. J, Image interpolating based data hiding in conjunction with pixel-shifting of histogram, The Journal of Supercomputing, Vol. 66, Sep. 2013, pp. 1093-1110.
11. Xiaolong Li, Bin Yang, Daofang Cheng, and Tieyong Zeng, A generalization of LSB Matching, IEEE signal process. Lett, Vol. 16, Feb. 2009.
12. Fridrich.J, Goljan. M and Du.R, Detecting LSB steganography in color and gray-scale images, IEEE Multimedia, Vol. 8, Dec. 2001, pp. 22-28.
13. Mielikainen.J, LSB matching revisited, IEEE signal process. Lett, Vol. 13, May. 2006, pp. 285-287.
14. Kede Ma, Weiming Zhang, Xianfeng Zhao, Nenghai Yu and Fenghua Li, Reversible data hiding in encrypted images by reserving room before encryption, IEEE Trans. Information forensics and security, Vol. 8, Mar. 2013, pp. 553-562.
15. Zhang. X, Reversible data hiding in encrypted image, IEEE signal process. Lett, Vol. 18, Apr. 2011, pp. 255-258.
16. Hong.W. T, Chen and Wu, H, An improved reversible data hiding in encrypted images using side match, IEEE signal process. Lett, 2012, pp. 199-202.
17. Zhang.X, Separable reversible data hiding in encrypted image, IEEE Trans. Information forensics and security, 2012, pp. 826-832.
18. Miscellaneous gray level Test images. http://decsai.ugr.es/cvg/dbimagenes/g512.php.