Multimed Tools Appl (2017) 76:14151–14167
DOI 10.1007/s11042-016-3815-2

A high payload data hiding scheme based on modified AMBTC technique

Aruna Malik [1] · **Geeta Sikka** [1] · **Harsh K. Verma** [1]

Received: 30 September 2015 / Revised: 24 July 2016 / Accepted: 28 July 2016 /
Published online: 10 August 2016

Abstract In this paper, we propose a data hiding scheme which uses our modified AMBTC compression technique for embedding the secret data. Our modified AMBTC technique converts the one bit plane into two bit plane which helps in achieving better quality compressed image as well as high capacity. In this scheme, we first apply the original AMBTC technique on the given cover image then identify the smooth and complex blocks using a user defined threshold value. In case of the smooth blocks, it converts the one bit plane into two bit plane using mean value of the block and replaces all the bits of the bit plane with the secret data bits. It calculates four quantization levels in place of two old quantization levels. In case of complex blocks, it converts the one bit plane into two bit plane but here only the first LSBs of the newly constructed bit plane is replaced by the secret data bits. The four new quantization levels are calculated using the resultant bit plane. Thus, this scheme is able to embed 2 bits into each pixel of the smooth blocks and one bit in each pixel of complex blocks. It provides good quality stego image because the introduced error during the secret data embedding is reduced by having four quantization levels. Experimentally, our scheme is superior to the existing AMBTC based data hiding schemes in terms of both data hiding capacity and image quality. In fact, the proposed scheme hides approximately two times more secret data than the existing schemes with better image quality.

Keywords Data hiding · Quantization level · Secret data · Stego image · Absolute moment block truncation coding

✉ Aruna Malik
arunacsrke@gmail.com

Geeta Sikka
sikkag@nitj.ac.in

Harsh K. Verma
vermah@nitj.ac.in

[1] Department of Computer Science & Engineering, National Institute of Technology, Jalandhar, India

1 Introduction

Due to the rapid advancement of networking technology, the digital information is widely distributed over the internet. Peoples can easily access and interchange various multimedia information like audio, video, and images more conveniently. However, distribution of these types of data has enormously increases the chance of interception, forgery, and fraud. Thus, protection of the sensitive information is highly demanded. Data hiding, also known as information hiding, plays an important role in information security. It embeds the secret information into cover media such that an unintended observer will not be aware of the existence of the hidden messages [18]. In the field of data hiding, watermarking and steganography are the two main techniques that have raised intensive concern among researchers. As far as watermarking is concerned, it alters a cover object, either imperceptibility or perceptibility to embed a message about the cover objects. Generally, it is used for copyright protection, transaction tracking, and broadcast monitoring. In contrast, steganography is mainly used for secret communication. A steganographic method undetectably alters a cover object to hide a secret message. Thus, steganographic methods can hide the very presence of covert communications [5]. There exist two main parameters that affect data hiding scheme, namely visual quality and embedding capacity (or payload). A data hiding scheme, having low image distortion is generally preferred, as it does not raise suspicion among invaders or attackers. On the other hand, data hiding scheme having a high embedding capacity is generally preferred, so that a large amount of secret data can be embedded in the cover image [1]. Data hiding techniques can be carried out in three domains [8] namely spatial, transform, and compressed domain. In the spatial domain, all the pixel values are directly modified to embed the secret data [13]. In case of transform domain, the cover image is firstly transformed into frequency coefficients by utilizing a frequency oriented mechanism such as discrete cosine transformation (DCT) [21] or discrete wavelet transformation (DWT) [20]. Subsequently, secret data are combined with the relative coefficients. Finally, the modified coefficients are used to reconstruct the frequency form image as a stego image. Due to the limited bandwidth of the communication channels, it is preferable to use the compressed or small sized media files. There have been many compression techniques to tackle this problem; however, the lossy image compression techniques like JPEG [9], vector quantization (VQ) [15], block truncation coding (BTC) [7] etc., are more preferable when the media is image or video. One of the most powerful compression techniques is BTC. Furthermore, the BTC has been improved using absolute moment block truncation coding (AMBTC) [10] in which instead of using the standard deviation the first absolute moment is preserved along with the mean. AMBTC is computationally simpler than BTC and also typically results in a lower mean squared error (MSE). Our proposed data hiding scheme is carried out in compression domain more specifically using AMBTC technique. In this paper, absolute moment block truncation coding (AMBTC) technique is modified to embed a large amount of the secret data. This technique also maintains the stego image quality. To validate this point, experimental results are presented in Section 4 which not only proves that the proposed scheme hides large amount of secret data while maintaining the stego image quality but also shows that the scheme is superior to other existing schemes in terms of both data hiding capacity and stego image quality.

The rest of the paper is organized as follows. In Section 2, we briefly introduce some related works of BTC and AMBTC based data hiding methods. The embedding and extraction process of the proposed scheme is introduced in Section 3. The experimental results and analysis are given in Section 4. Finally, in Section 5, the paper is concluded.

2 Related works

In this section, we will briefly review several data hiding schemes for BTC and AMBTC compressed images proposed in last decade and so. Chuang et al. [4] introduced a data embedding scheme based on BTC technique. In their scheme, BTC technique is applied to compress gray scale cover image which results into two quantization values and one bitmap for each block. Then, a predefined threshold is used to classify each BTC encoded block as smooth or complex. Subsequently, secret data is embedded into the bitmap of the selected BTC encoded blocks. This scheme is limited to gray scale images only and the stego image quality is significantly degrades with the increase in threshold value thus increasing the suspicion from the attackers. Later, Chang et al. [2] introduced a reversible data hiding method for color images compressed by BTC. In this method, BTC is applied to each block of a cover image and three pairs of quantization levels and three bitmaps are obtained. To further improve the compression rate, a genetic algorithm is used to find an approximate optimal bitmap out of three bitmap. Chen et al. [3] discussed a data hiding scheme which uses the difference between quantized values to embed the secret data. This scheme provides better quality stego image. In 2011, Li et al. [11] suggested a reversible data hiding scheme for BTC compressed images using bitmap flipping and histogram shifting strategies for the high mean values and low mean values. Due to the lack of extra data in the stego bit stream, this achieves very low image distortion after data embedding and enhances the security of embedded data. However, this technique has low data hiding capacity. To further improve the data hiding capacity, Sun et al. [19] discussed a reversible data hiding scheme based on the joint neighbor coding technique to BTC compressed images. Using the joint neighbor coding technique, both the high mean and low mean table are used for embedding the secret data. It has been observed that this scheme obtained a capacity four times larger as the number of blocks in the cover image, however, it requires extra data to be included in the stego code stream. Later, Zhang et al. [22] proposed a reversible data hiding scheme for BTC compressed images by further losslessly encoding the BTC compressed data according to the secret bits. Initially, the BTC technique is applied to the original image to get the BTC compressed data that consists of a high mean table, a low mean table, and a bit plane sequence which is used to hide the secret data. However, this scheme needs extra information such as a key to generate the sequence, length of secret data, and the cover image size to achieve reversibility. Later, Lin et al. [14] presented an AMBTC compression based reversible data hiding method. In this technique, the redundancy of compressed blocks is firstly investigated, which is used to classify the blocks of AMBTC compressed blocks as embeddable or non embeddable. In order to embed the secret data in embeddable blocks, four disjoint sets are designed by using different combinations of the mean value and the standard deviation. Jeng et al. [17] introduced a reversible data hiding scheme for AMBTC compressed images using a reference matrix. The quantization levels of each AMBTC compressed image block are used to embed the secret data by using a reference matrix. For each image block, original quantization levels are changed into another stego message which is combined with a bitmap. Ou et al. [16] discussed an AMBTC based data hiding scheme having minimum distortion. In this technique, a threshold is defined to divide the blocks of the AMBTC compressed codes as smooth and complex, in which secret data are embedded. In case of smooth blocks, the bit planes are directly used to embed the data by replacing the bits with the secret data bits. Next, the two quantization levels are re calculated to reduce the caused distortion. For complex blocks, a proportion of secret bits are hidden by exchanging the order of two quantization levels with together toggling the bit plane. Huang et al. [6] discuss a hybrid secret hiding schemes based on absolute moment block truncation coding. It is an extension of Ou et al. [16] for embedding more secret data into the complex and smooth blocks. It utilizes two different hiding methods for embedding the secret data into the smooth and complex blocks. In case of smooth blocks, the small variation of the block is adopted to define the embedding rule to minimize the distortion after data embedding. As for the complex blocks, the large variation of the block is used to embed more secrets while maintaining good visual quality. Li et al. [12] discuss a bi-stretch reversible data hiding algorithm for absolute moment block truncation coding compressed images. In this scheme, the AMBTC compressed image is divided into non overlapped blocks first, after that, four feasible cases are employed to embed secret data, which takes advantage of the characteristics of the coefficients of the AMBTC compressed image and leads to very small distortion of the AMBTC compressed image. In this paper, we extend the work of Ou et al. [16] to embed the larger amount of secret data. Our aim in this paper is to provide both good quality stego image and high data hiding capacity at the same time.

The proposed scheme employs a user defined threshold value to classify the AMBTC compressed blocks as complex block and a smooth block. Here, each pixel of the image is represented by two bits in the bit plane and four quantization levels are calculated instead of two levels so that a large amount of secret data is embedded with better image quality. In the next section, the proposed scheme is discussed.

3 Proposed scheme

In this section, the proposed scheme which modifies the AMBTC compression technique to embed a large amount of secret data is discussed. It converts the one bit plane into two bit plane so that large amount of secret data is embedded and caused distortion can also be limited. It uses a user defined threshold value (which is adjustable as per the requirement) to categorize the AMBTC compressed blocks into smooth blocks and complex blocks. In case of smooth block, to embed the secret data, the bit plane is replaced with the secret data bits. Later, the four quantization levels are calculated using Eqs. (1)–(4) so that caused distortion is minimized. In case of complex blocks, the secret data is embedded into the first LSB of the bit plane and four new quantization levels are calculated using Eqs. (1)–(4) (like in the case of smooth blocks) so that the resultant block could be brought close to the original block to maintain the stego image quality. Thus, the proposed scheme is able to embed two bits into each pixel of the smooth blocks and one bit into every pixel of complex blocks. The algorithm of the proposed scheme is discussed in the following section.

3.1 Embedding algorithm

Input- I: original image of size N × N pixels, thr: threshold, S: secret data bit stream.
Output- AMBTC compressed stego codes
BEGIN
Step 1: Divide input image I into 4 × 4 non overlapping blocks in raster scan order.
Step 2: Process each block IB_i using AMBTC scheme to get two quantization levels a_i and b_i, mean value, and a bit plane B_i.
Step 3: Reprocess each block IB_i to get two bit plane instead of one bit plane so that caused noise due to the compression is reduced.

- If the original image pixel value of the block is less than the quantization level a_i then represent the corresponding pixel by '00' in the bit plane.

- Else if the original image pixel value is greater than or equal to the quantization level a_i and less than or equal to the mean value of the block, then represent the corresponding pixel by '01' in the bit plane.
- Else if the original image pixel value is greater than to the mean value of the block and less than to the quantization level b_i, then represent the corresponding pixel by '10' in the bit plane.
- Otherwise, the pixel is represented by '11' in the bit plane.

Step 4: Calculate absolute difference value D_i for the block IB_i, such that $D_i = |a_i - b_i|$.
Step 5: If $D_i \le thr$, means IB_i is a smooth block. Replace bits of the two bit plane $B_i^{'}$ with the 32 bits S_1 from secret data S for embedding the same, which in turn gives a new bit plane $B_i^{''}$ with embedded S_1. Subsequently, remove S_1 from S such that $S = S–S_1$.
Step 6: Calculate four new quantization levels as follows to reduce the distortion.

$$a_i^{'} = \frac{1}{q_{00}} \sum\nolimits_{x_i \in G_{00}^{'}} x_i \tag{1}$$

$$d_i^{'} = \frac{1}{q_{01}} \sum\nolimits_{x_i \in G_{01}^{'}} x_i \tag{2}$$

$$c_i^{'} = \frac{1}{q_{10}} \sum\nolimits_{x_i \in G_{10}^{'}} x_i \tag{3}$$

$$b_i^{'} = \frac{1}{q_{11}} \sum\nolimits_{x_i \in G_{11}^{'}} x_i \tag{4}$$

where, q_{00} , q_{01} , q_{10} and q_{11} are the number of 00, 01, 10, and 11in the bit plane, respectively.
Step 7: If $\left|a_i^{'}–b_i^{'}\right| \le thr$, indicates that new quantization levels can maintain the smoothness property of the block IB_i, then add $\left\{a_i^{'}, b_i^{'}, c_i^{'}, d_i^{'}, B_i^{''}\right\}$ into I_S (stego image). If $\left|a_i^{'}–b_i^{'}\right| > thr$ indicates that the block is violating the smoothness property. So, to maintain its smoothness, add the old quantization levels a_i and b_i in place of $a_i^{'} and\ b_i^{'}$ into I_S such that the compressed code is $\left\{a_i, b_i, c_i^{'}, d_i^{'}, B_i^{''}\right\}$. Finally, go back to the Step 3 to process the next block.
Step 8: If $D_i > thr$, the block IB_i is classified as a complex block. Embed the 16 bits S_1 from S by replacing the first LSB of the newly constructed two bit plane with the S_1 bits. After the replacement, a new bit plane $B_i^{''}$ with embedded S_1 is obtained. Subsequently, remove S_1 from S such that $S = S–S_1$.
Step 9: According to the new bit plane $B_i^{''}$ four new quantization levels $a_i^{'}$, $d_i^{'}$, $c_i^{'}$ and $b_i^{'}$ are re-calculated using Eqs. (1)–(4), respectively, to reduce the distortion.
Step 10: If $\left|a_i^{'}–b_i^{'}\right| > thr$, indicates that new quantization levels can maintain the complexness property of the block IB_i, then add $\left\{a_i^{'}, b_i^{'}, c_i^{'}, d_i^{'}, B_i^{''}\right\}$ into I_S. If $|a_i^{'}–b_i^{'}| \le thr$ indicates that the block is violating the complexness property. So, to maintain its complexness, add the old quantization levels a_i and b_i in place of $a_i^{'}$ and $b_i^{'}$ into I_S such that the compressed code is $\left\{a_i, b_i, c_i^{'}, d_i^{'}, B_i^{''}\right\}$. Finally, go back to the Step 3 to process the next block.
Step 11: Repeat the steps 3 to 10 to obtain all the compressed image codes in I_S.
END

3.2 Illustration of embedding process

An illustration of the proposed embedding process for stego image construction is shown in Fig. 1. Let the secret data bit stream S is $(101001010010111\ 000101110111001011110010100001010)_2$ and the threshold (thr) is 15. Further assume, we have two different blocks each of size 4×4 pixels where the first block IB_1 contains pixel values 100, 90, 85, 82, 76, 75, 72, 77, 80, 81, 80, 81, 82, 81, 79, and 79 and second block IB_2 contains pixel values 186, 211, 212, 205, 181, 208, 211, 202, 185, 205, 203, 193, 194, 211, 202, and 190. The original blocks IB_1 and IB_2 are compressed using AMBTC compression technique and compressed codes $\{77, 85, B_1\}$ and $\{188, 207, B_2\}$ are obtained for the blocks IB_1 and IB_2, respectively. The mean values are 81 and 200 for the blocks IB_1 and IB_2, respectively.

The bit planes for both the blocks IB_1 and IB_2 contain only two values either 0 or 1 for a corresponding pixel. For the block IB_1, absolute difference value D_1 is calculated, such that $D_1 = |a_1 - b_1| = |77 - 85| = 8 < thr$, which is less than the defined threshold value. It means the block IB_1 is a smooth block. The block IB_1 is processed to get a two bit plane B_1' instead of one bit plane B_1 as detailed in Section 3.1. To embed the secret data in smooth block, the bits of the bit plane B_1' are replaced with the 32 bits of the secret data i.e. $S_1 = (10100101001011100010111011100101\ )_2$, which in turn gives the new bit plane B_1''. Subsequently, S_1 from S is removed such that $S = S - S_1$. Now, the new quantization levels a_1', b_1', c_1', and d_1', are calculated so that caused distortion is reduced with respect to the new bit plane B_1'' using Eqs. (1)–(4). As, the absolute difference $|a_1' - b_1'| = |78 - 78| = (0 < thr)$, that means smoothness property of the block $IB_1$is maintained so new stego compressed code is $\{a_1', b_1', c_1', d_1', B_1''\} = \{78, 78, 83, 81, B_1''\}$. For the block IB_2, absolute difference value D_2

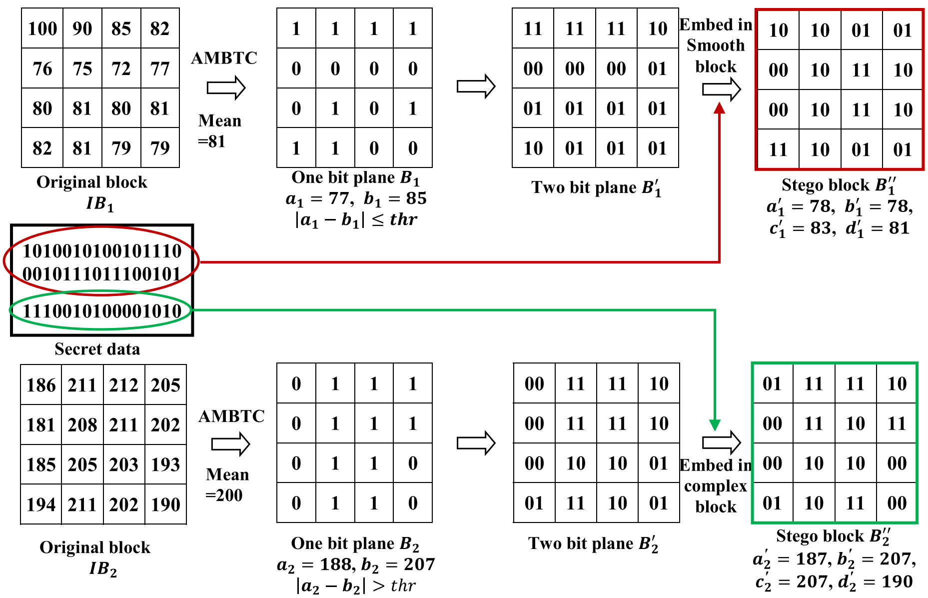


Fig. 1 An example to explain the proposed embedding process

is calculated, such that $D_2 = |a_2 - b_2| = |188 - 207| = 19 > thr$, which means the block IB_2 is a complex block. In case of complex block, the block is processed to get a two bit plane instead of one bit plane as detailed in Section 3.1. Now, the next 16 bits of the secret data i.e., $S_1 = (1110010100001010)_2$ from S are embedded into the first LSB of the block B_2. After the bit plane replacement, a newer bit plane (B_2'') with embedded S_1 is obtained. Subsequently, S_1 is removed from S such that $S = S - S_1$. Now, four new quantization levels a_2', b_2', c_2', and d_2' are calculated with respect to new bit plane B_2'' so that the caused distortion is limited. As, the absolute difference $|a_2' - b_2'| = |187 - 207| = (20 > thr)$, that means complexness of the block B_2'' is maintained and the value of old quantization levels a_2 and b_2 can be replaced by the new quantization levels a_2', b_2', c_2', and d_2',. Finally, the obtained stego compressed code will be as $\left\{ a_2', b_2', c_2', d_2', B_2'' \right\} = \left\{187, 207, 207, 190, B_2''\right\}$.

3.3 Extraction phase

In the extraction phase, the secret data is extracted from the AMBTC compressed codes for which a detailed algorithm is explained below.
Input- AMBTC compressed stego codes, thr: threshold
Output- S_D: Secret data bit stream
Step 1: Calculate the difference between a_i' and b_i' in I_S, such that $D_i = |a_i' - b_i'|$.
Step 2: If $D_i \leq thr$, the 32 bits of the bit planes are extracted as the S_1 which are added into S_D. Then Go to Step 4.
Step 3: If $D_i > thr$, then extract first LSB of every pixel from the bit plane B_2'' as S_1 and add the S_1 into S_D.
Step 4: Go to step 1 until all the codes are processed.
Thus proceeding, we can extract the complete secret data bit stream S_D from the AMBTC compressed stego codes.

3.4 Illustration of extracting process

In this section, an example is taken for illustrating the process of data extraction. We recall the example given in Section 3.2 for extraction of the secret data bits stream from the compressed codes. For the first compressed code, the absolute difference value D_1 of two quantization levels a_1' and b_1' is calculated, such that $|a_1' - b_1'| = |78 - 78| = (0 < thr)$, which is less than the defined threshold value. It means that the block is a smooth block. To extract the secret data in case of smooth block, the bits of the bit plane B_1'' are extracted. Thus, 32 bits ($S_1 = (10100101001011100010111011100101)_2$) is obtained from the bit plane B_1'' and these 32 bits are added into S_D. For the second compressed code, the absolute difference value D_2 of two quantization levels a_2' and b_2' is calculated, such that $|a_2' - b_2'| = |187 - 207| = (20 > thr)$, which is greater than the threshold value (i.e., 15) means the corresponding block is a complex block. To extract the secret data in case of complex block, the first LSB of every pixel from the two bit plane B_2'' is extracted. Thus, 16 bits ($S_1 = (1110010100001010)_2$) is obtained from the bit plane B_2'' and these 16 bits are added into S_D. This process can be continues for all the stego blocks to extract the secret data bit stream.

4 Experimental results and discussions

This section discusses the comparative analysis of our proposed scheme with other existing schemes. For exhaustive analysis, we have taken nine gray scale images, namely "Lena", "Baboon", "Plane", "Peppers", "Boats", "Barb", "House", "Houses", and "Zelda" each of size 512 × 512 pixels as shown in Fig. 2. The scheme is implemented in MATLAB running on Intel (R) Core (TM) i5 processor 3.20-GHz with 4-GB RAM hardware platform. The secret data used in the experiment is a stream of random bits, generated by a pseudo random number generator (PRNG). For comparison, three parameters, namely hiding capacity, peak signal to noise ratio (PSNR), and compression ratio are used to evaluate the performance of existing schemes and the proposed scheme.

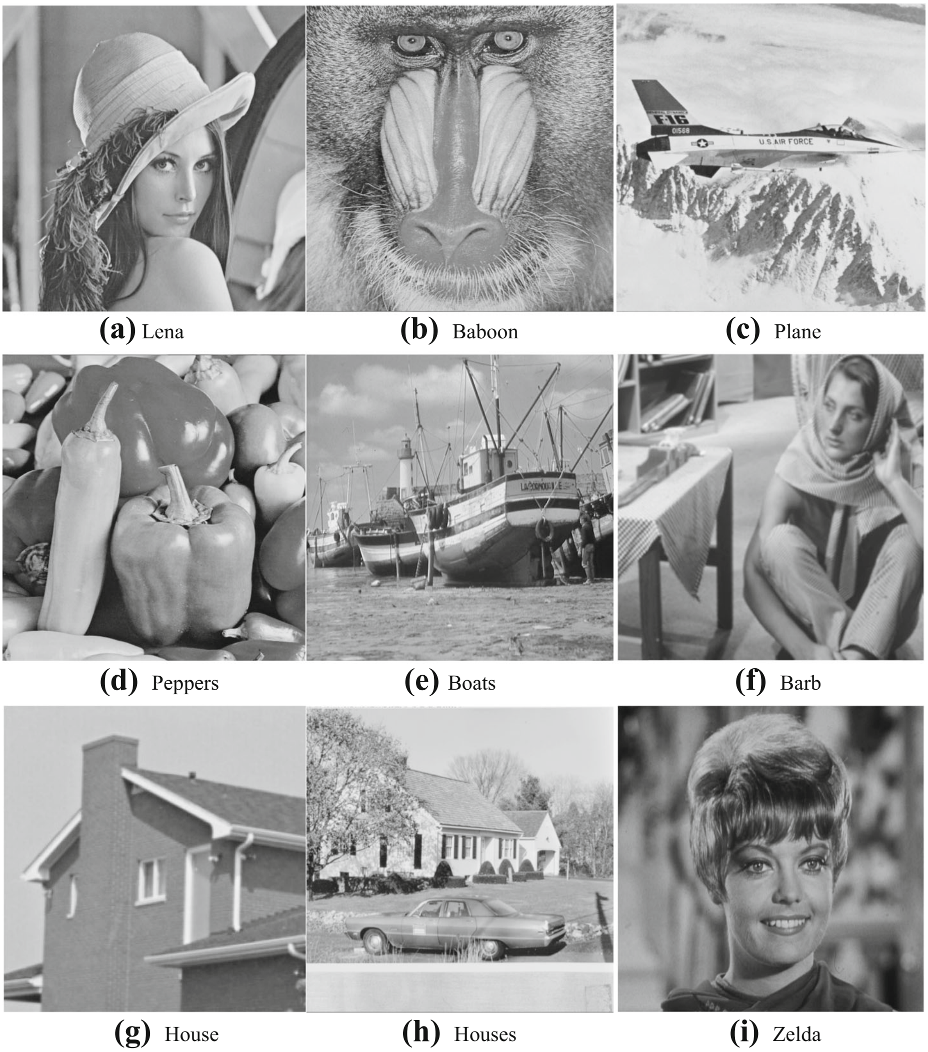
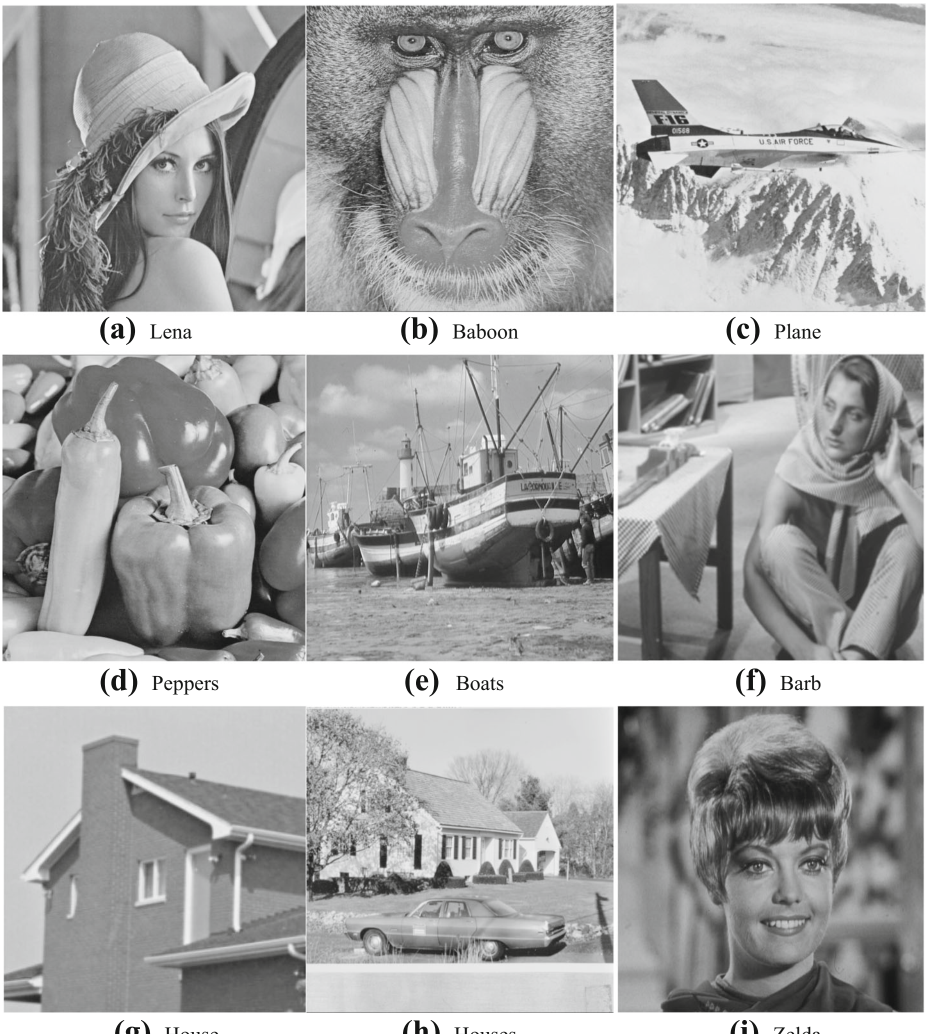


Fig. 2 Cover images, each of size 512 × 512

The hiding capacity refers to the number of secret data bits hidden into the cover image. The visual quality of the image is measured by the peak signal to noise ratio (PSNR) defined as:

$$\mathrm{PSNR} = 10\log_{10}\left[\frac{255*255}{\frac{1}{N\times N}\sum_{i=1}^{N}\left(x_{ij}-y_{ij}\right)^2}\right] \tag{5}$$

where, x_{ij} and y_{ij} are the pixels located at the i^{th} row and j^{th} column of cover image x and stego image y, each of size N × N pixels, respectively. The stego images of the proposed scheme are shown in Fig. 3.



(a) Lena **(b)** Baboon **(c)** Plane **(d)** Peppers **(e)** Boats **(f)** Barb **(g)** House **(h)** Houses **(i)** Zelda

Fig. 3 Stego images of the proposed scheme

The compression ratio is defined as the ratio between the uncompressed size and compressed size:

$$\text{Compression ratio} = \frac{\text{Compressed size}}{\text{Uncompressed size}} \tag{6}$$

As discussed in proposed algorithm, the threshold value is used to determine whether the block is smooth or not. The value of the threshold certainly affects the hiding capacity and the image quality. As increasing the threshold value, the image quality is deteriorated but the hiding capacity is increased. The proposed scheme with different threshold values is implemented by using the cover images sized 512 × 512 pixels as shown in Fig. 2(a–i). The experimental results are illustrated in Table 1, where the PSNRs, hiding capacity, and compression ratio are shown. The trade off among the hiding capacity and the visual quality of the image can vary from the application to application by setting different values of the threshold. The value of the threshold can also be set according to the requirements of the application.

We have compared the performance of the proposed scheme with other popular AMBTC based reversible data hiding schemes of Chuang et al. [4], Ou et al. [16], Lin et al. [14], Huang et al. [6], and Li et al. [12]. The proposed scheme is more closer to the Ou et al. [5] scheme that's why to critically analyse the performance the results of the proposed scheme are compared with [16] for K = 2, 3, and 4, where K is the dimension of block size. Therefore, the results are taken at 4 × 4, 3 × 3, and 2 × 2 sized blocks for Ou et al. [16]. From Table 1, it is evident that the proposed scheme is performing better than the Chuang et al. [4], Ou et al. [16] at K = 4, Lin et al. [14], Huang et al. [6], and Li et al. [12] for both data hiding capacity and PSNR. However, its image quality performance i.e., PSNR is comparable to Ou et al. [16] at K = 3 and providing approximately more than two times better data hiding capacity for all the images. For K = 2, the Ou et al. [16] scheme provides much better PSNR than the proposed scheme, however its hiding capacity is approximately 5 times lower for all the images. Overall, we can say the proposed scheme is having much better data hiding capacity than the other contemporary AMBTC based data hiding schemes, however, its performance can be comparable for the PSNR in some cases like K = 3 of Ou et al. [16]. The compression ratio for K = 2, 3 and 4 of Ou et al. [16] and the proposed scheme is also provided in Table 1. Figure 4 illustrates the results of performance in terms of embedding capacity (bits) and PSNR (dB) for some of the typical standard images like Lena, Peppers, Boats, and Baboon to cover the thorough analysis. From Fig. 4, it is clearly evident that our scheme achieves more than two times hiding capacity while maintaining image quality of stego-image with respect to Lin et al. [14], Ou et. [16], Huang et al. [6], and Li et al. [12] schemes because it hides two secret data bits into every pixel of the smooth blocks and one secret data bit into the each pixel of complex blocks for all the images. It is also clearly evident from the Fig. 4((a)–(d)) the image quality of our method is better than that of Lin et al. [14], Ou et al. [16] at K = 3 and 4, Huang et al. [6], and Li et al. [12] methods because it reduces the caused distortion of the AMBTC compressed image by having four quantization levels. Hence, the quality of the stego images is maintained as shown in Fig. 3(a–i). Based on the experimental results and comparisons, it can be revealed that our proposed scheme outperforms the other methods, However, when the performance is compared with Ou

Table 1 Comparison of hiding capacity, PSNR, and compression ratio for different images between the proposed scheme and other AMBTC and BTC based schemes

Methods	Metrics	Lena	Baboon	Plane	Peppers	Boats	Barb	House	Houses	Zelda
Threshold thr = 10										
Proposed method	PSNR	33.45	27.66	32.31	33.52	31.66	29.80	35.61	31.32	35.52
	Capacity	428,752	298,400	434,640	440,432	403,184	374,544	438,215	390,560	427,404
	Compression ratio	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Ou et al. [16] K = 4	PSNR	32.67	26.92	31.91	33.36	31.32	29.22	33.07	29.88	34.25
	Capacity	172,579	50,374	178,099	183,529	148,609	121,759	181,451	136,774	171,315
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Ou et al. [16] K = 3	PSNR	34.61	30.34	34.34	35.40	32.84	34.31	36.98	33.52	37.91
	Capacity	179,703	59,508	185,202	193,383	107,028	141,489	212,346	141,057	212,220
	Compression ratio	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
Ou et al. [16] K = 2	PSNR	42.11	35.63	42.10	41.88	39.71	43.54	41.04	46.62	46.14
	Capacity	85,688	24,488	87,676	77,044	22,904	80,076	70,036	102,336	104,840
	Compression ratio	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625
Chuang et al. [4]	PSNR	32.03	26.85	31.54	32.37	31.04	29.01	33.11	28.78	32.98
	Capacity	166,608	36,256	172,496	178,288	141,040	112,400	175,436	128,416	167,849
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Threshold thr = 20										
Proposed method	PSNR	32.72	27.49	31.97	32.76	30.79	29.64	34.71	30.96	34.82
	Capacity	475,632	358,304	468,448	484,656	449,376	414,048	469,677	451,979	480,515
	Compression ratio	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Ou et al. [16] K = 4	PSNR	31.78	26.58	31.34	32.42	30.62	28.23	32.85	29.04	33.89
	Capacity	216,529	106,534	209,794	224,989	191,914	158,794	210,946	194,354	221,107
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Ou et al. [16] K = 3	PSNR	33.58	29.63	33.61	34.39	31.44	33.00	35.98	32.47	36.34
	Capacity	222,624	130,014	216,126	230,472	200,898	196,632	236,349	193,518	245,421
	Compression ratio	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
Ou et al. [16] K = 2	PSNR	40.41	34.85	40.76	39.82	36.38	40.89	39.33	44.42	44.12

Table 1 (continued)

Methods	Metrics	Lena	Baboon	Plane	Peppers	Boats	Barb	House	Houses	Zelda
	Capacity	106,124	58,384	102,224	106,412	78,864	103,616	92,960	111,032	114,460
	Compression ratio	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625
Chuang et al. [4]	PSNR	30.43	26.11	30.39	30.78	29.63	28.19	32.45	28.07	31.22
	Capacity	213,488	96,160	206,304	222,512	187,232	151,904	209,864	190,254	217,009
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Threshold thr = 30										
Proposed method	PSNR	31.98	27.04	31.48	32.09	30.11	28.97	33.92	29.12	32.78
	Capacity	494,272	396,048	482,576	499,568	476,416	445,072	490,821	478,645	499,349
	Compression ratio	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Ou et al. [16] K = 4	PSNR	30.91	26.02	30.75	31.64	29.69	28.09	31.92	28.75	32.01
	Capacity	234,004	141,919	223,039	238,969	217,264	187,879	230,769	219,354	238,764
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Ou et al. [16] K = 3	PSNR	32.65	28.45	32.78	33.58	30.52	31.44	34.99	31.32	35.33
	Capacity	238,518	179,937	230,409	241,920	227,304	228,906	246,321	221,040	254,925
	Compression ratio	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
Ou et al. [16] K = 2	PSNR	39.24	33.71	39.54	38.90	34.66	39.10	37.58	42.42	43.17
	Capacity	111,836	78,136	107,600	111,480	104,644	111,736	104,512	114,876	116,436
	Compression ratio	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625
Chuang et al. [4]	PSNR	29.06	25.04	29.34	29.58	28.05	26.83	31.04	27.45	30.83
	Capacity	232,128	133,904	220,432	237,424	214,272	182,928	227,896	216,504	231,040
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Threshold thr = 40										
Proposed method	PSNR	31.05	26.15	30.77	31.45	29.41	27.74	32.96	28.02	32.07
	Capacity	506,608	430,240	493,552	508,368	491,648	466,432	507,909	496,772	507,647
	Compression ratio	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Ou et al. [16] K = 4	PSNR	30.02	25.17	29.97	30.88	28.83	27.22	30.87	27.65	31.79
	Capacity	245,569	173,974	233,329	247,219	231,544	207,904	246,789	236,348	246,543

Table 1 (continued)

Methods	Metrics	Lena	Baboon	Plane	Peppers	Boats	Barb	House	Houses	Zelda
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Ou et al. [16] K = 3	PSNR	31.69	27.16	31.82	32.80	29.76	30.30	33.84	30.13	34.70
	Capacity	248,661	216,693	240,165	248,445	240,174	244,962	253,467	237,717	258,498
	Compression ratio	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
Ou et al. [16] K = 2	PSNR	38.44	32.55	38.41	38.33	34.05	38.09	36.35	41.22	42.72
	Capacity	114,324	90,596	110,744	113,336	109,628	114,756	110,076	116,548	116,876
	Compression ratio	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625
Chuang et al. [4]	PSNR	27.78	23.58	28.06	28.50	26.76	25.47	30.16	26.78	29.46
	Capacity	244,464	168,096	231,408	246,224	229,504	204,288	240,765	234,859	239,429
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Threshold thr = 50										
Proposed method	PSNR	29.85	25.11	30.07	30.84	28.70	26.97	32.10	27.11	31.62
	Capacity	513,632	462,608	501,424	513,312	502,624	483,104	512,425	506,905	514,764
	Compression ratio	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
Ou et al. [16] K = 4	PSNR	29.30	24.11	29.20	30.29	28.00	26.31	30.04	27.05	30.24
	Capacity	252,154	204,319	240,709	251,854	241,834	223,534	251,022	245,847	253,215
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25
Ou et al. [16] K = 3	PSNR	31.03	26.17	31.01	32.18	29.11	29.59	32.97	29.34	34.36
	Capacity	253,575	238,149	246,357	251,937	247,734	252,216	257,517	246,357	259,659
	Compression ratio	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375	0.375
Ou et al. [16] K = 2	PSNR	37.87	31.51	37.41	37.86	33.47	37.45	35.50	40.91	42.59
	Capacity	115,552	100,088	112,868	114,380	112,784	115,948	112,984	116,924	116,944
	Compression ratio	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625	0.625
Chuang et al. [4]	PSNR	26.80	22.00	26.93	27.66	25.58	24.17	29.98	26.21	29.07
	Capacity	251,488	200,464	239,280	251,168	240,480	220,960	247,802	244,687	248,654
	Compression ratio	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25	0.25

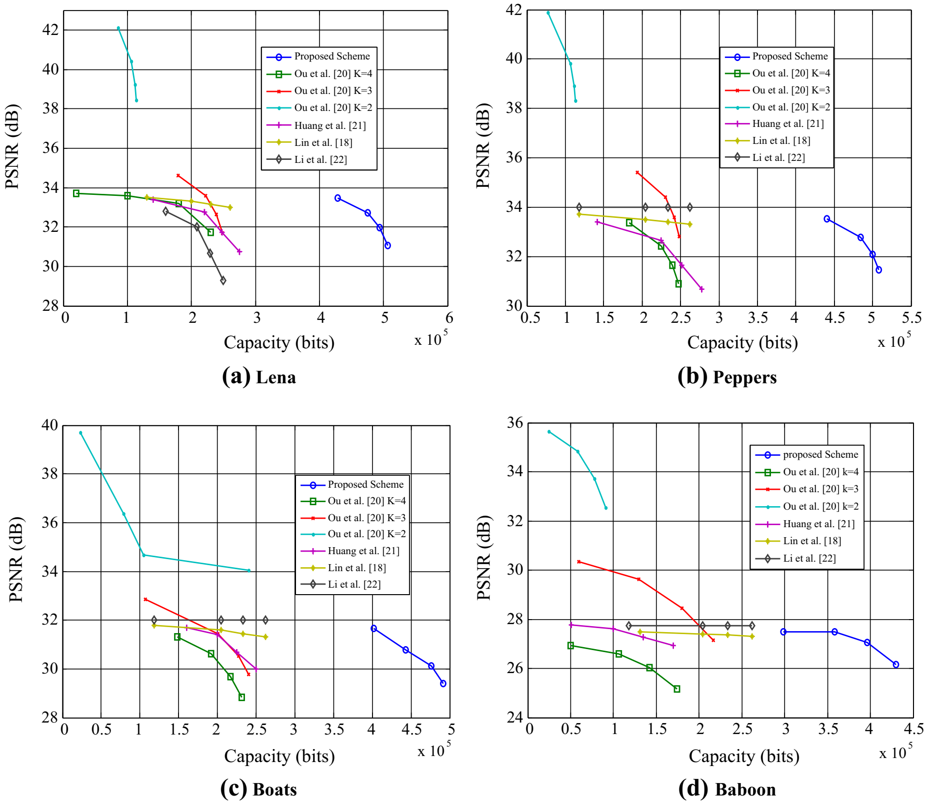


Fig. 4 Performance comparisons of the proposed scheme and other related schemes based on AMBTC compressed image for four typical images: **a** Lena, **b** Peppers, **c** Boats, and **d** Baboon

et al. [16] at K = 2, the proposed scheme is achieving approximately 5 times better hiding capacity but comparatively poor PSNR. Conclusively, we can say that the image quality of the Ou et al. [16] at K = 2 is better than the proposed scheme because [16] is able to hide very less number of secret data bits which helps in maintain image quality. To analyse the performance of the proposed scheme at different threshold with its parent method [16], the results are provided in the Table 1. From the Table 1, it is clearly evident that our scheme is superior to Ou et al. [16] and Chuang et al. [4] methods in terms of both image quality and data hiding capacity, irrespective of the cover image. However, it reduces the compression level of the AMBTC compression technique but is able to compensate it better visual quality and higher data hiding capacity.

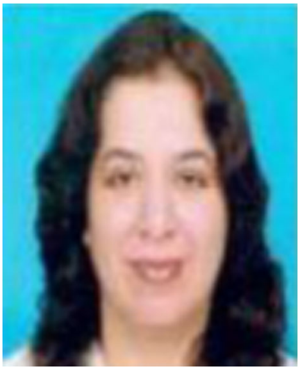
5 Conclusion

In this paper, we have proposed a data hiding scheme that uses our modified AMBTC compression technique to embed the secret data. Our modified AMBTC technique provides two bit plane unlike the one bit plane in original AMBTC which helps in generating four quantization levels so that the resultant image quality is improved. Furthermore, it increases the hiding capacity. This scheme embeds almost 2 times more secret data than the existing schemes with better visual quality. However, our scheme affects the AMBTC compression ratio but the amount of secret data embedded into the codes compensates this overhead. Experimentally, we have shown that our scheme is superior to the existing AMBTC based data hiding schemes in terms of both the data hiding capacity and image quality.

References

1. Chang C, Kieu T (2010) A reversible data hiding scheme using complementary embedding strategy. Inf Sci 180:3045–3058
2. Chang C, Lin C, Fan Y (2008) Lossless data hiding for color images based on block truncation coding. Pattern Recogn 41:2347–2357
3. Chen J, Hong W, Chen T, Shiu C (2010) Steganography for BTC compressed images using no distortion technique. Imaging Sci J 58(4):177–185
4. Chuang J, Chang C (2006) Using a simple and fast image compression algorithm to hide secret information. Int J Comput Appl 28(4):329–333
5. Cox I, Miller M, Bloom J, Fridrich J, Kalker T (2007) Digital watermarking and steganography. Morgan Kaufmann Publishers Inc., San Francisco ISBN 978-0-12-372585-1
6. Huang YH, Chang CC, Chen YH (2016) Hybrid secret hiding schemes based on absolute moment block truncation coding. Multimed Tools Appl. doi:10.1007/s11042-015-3208-y
7. Hu YC, Chang CC (1999) Quadtree-segmented image coding schemes using vector quantization and block truncation coding. Opt Eng 39(2):464–471
8. Langelaar G, Setyawan I, Lagendijk R (2000) Watermarking digital image and video data: a state -of-the overview. IEEE Signal Process Mag 17(5):20–46
9. Lee J, Chiou Y, Guo J (2013) A high capacity lossless data hiding scheme for JPEG images. J Syst Softw 86: 1965–1975
10. Lema M, Mitchell O (1984) Absolute moment block truncation coding and its application to color images. IEEE Trans Commun 32:1148–1157
11. Li C, Lu Z, Su Y (2011) Reversible data hiding for BTC-compressed images based on bit plane flipping and histogram shifting of mean tables. Inf Technol J 10(7):1421–1426
12. Li F, Bharanitharan K, Chang CC, Mao Q (2015) Bi-stretch reversible data hiding algorithm for absolute moment block truncation coding compressed images. Multimed Tools Appl. doi:10.1007/s11042-015-2924-7
13. Lin I, Lin Y, Wang C (2009) Hiding data in spatial domain images with distortion tolerance. Comput Stand Inter 31(2):458–464
14. Lin C, Liu X, Tai W, Yuan S (2013) A novel reversible data hiding scheme based on AMBTC compression technique. Multimed Tools Appl 74(11):3823–3842
15. Lu Z, Wang J, Liu B (2009) An improved lossless data hiding scheme based on image VQ-index residual value coding. J Syst Softw 82(6):1016–1024
16. Ou D, Sun W (2014) High payload image steganography with minimum distortion based on absolute moment block truncation coding. Multimed Tools Appl 74(21):9117–9139
17. Pan J, Li W, Lin C (2014) Novel reversible data hiding scheme for AMBTC-compressed images by reference matrix. Multidiscip Soc Netw Res 473:427–436
18. Petitcolas F, Anderson R, Kuhn M (1999) Information hiding – a survey". Proc IEEE 87(7):1062–1068
19. Sun W, Lu Z, Wen Y (2013) High- performance reversible data hiding for block truncation coding compressed images. SIViP 7(2):297–306
20. Xuan G, Shi Y, Yao Q, Ni Z, Yang C, Gao J (2006) Lossless data hiding using histogram shifting method based on integer wavelets. International Workshop on Digital Watermarking Lecture Notes in Computer Science 4823:323–332
21. Yang B, Schmucker M, Funk W, Brush C, Sun S (2011) Integer DCT-based reversible watermarking for images using companding technique. Proc Int J Electron Commun 65:814–826
22. Zhang Y, Guo S, Lu Z, Luo H (2013) Reversible data hiding for BTC-compressed images based on lossless coding of mean tables. IEICE Trans Commun 96(2):624–631

Aruna Malik, received her B.Tech. in Computer Science and Engineering from Uttar Pradesh Technical University, Lucknow, India and M.Tech. in Computer Science and Engineering from National Institute of Technology, Jalandhar, Punjab, India. She is pursuing her doctoral degree in Computer Science and Engineering, at National Institute of Technology, Jalandhar, Punjab, India. Her research areas lie in the area of Data hiding and Image processing.



Geeta Sikka is presently working as an Associate Professor in the Department of Computer Science and Engineering at National Institute of Technology, Jalandhar. She received her Ph.D in Computer Science and Engineering, from National Institute of Technology, Jalandhar, India. She did her Master's degree in Computer Science from Punjab Agricultural University, Ludhiana. Her research interests are Software Engineering, Databases, Data hiding and Data mining.

Harsh K. Verma is working as a Professor and Head of Computer Centre at National Institute of Technology, Jalandhar (India). He has done his Bachelor's degree in computer science and engineering in 1993 and Master's degree in Software Systems from Birla Institute of Technology, Pilani, in 1998. He received his Ph.D. degree from Punjab Technical University, Jalandhar (India) in 2006. He has many publications of international and national level to his credit. His research interests include Information security, Data hiding, Computer networks, Image processing and Scientific computing.