A High Capacity Data Hiding Scheme Using Modified AMBTC Compression Technique

Aruna Malik, Geeta Sikka, and Harsh Verma
Department of Computer Science and Engineering, Dr B R Ambedkar National Institute of Technology, India

Abstract: *In this paper, a data hiding scheme is proposed which modifies the Absolute Moment Block Truncation Coding (AMBTC) technique to embed a large amount of secret data. This scheme employs a user-defined threshold value to classify the AMBTC compressed blocks as complex block and smooth block. In the case of smooth blocks, the bit plane is replaced with the secret data bits. Later, the quantization levels are re-calculated so that distortion is minimized. While for complex blocks, the bit plane is reconstructed in which every pixel is represented by two bits instead of just one bit. Now, the secret data is embedded into the first LSB of the bit plane. Finally, four new quantization levels are calculated for preserving the closeness of the resultant block to the original block. Thus, the proposed scheme is able to utilize each and every pixel of the cover image to hide the secret data while maintaining the image quality. This scheme achieves 1 bit per pixel data hiding capacity for every image. Experimental results show that our scheme is superior to the other existing schemes in terms of both hiding capacity and image quality.*

Keywords: *Data hiding, quantization level, secret data, stego-image, absolute moment block truncation coding.*

Received September 17, 2015; accepted January 11, 2016

1. Introduction

With the expeditious progress of computer technology and the widespread usage of the internet, it becomes more and more convenient for people to access and interchange all kinds of multimedia information like audio, video, and images. However, since everyone could access the data transmitted over the internet and further interfere them, it makes the received data capricious and thus made the people concern about the information security. Data hiding plays an important role in information security [6, 15]. It has been found that there exist two parameters that mainly affect the data hiding scheme, visual quality and embedding capacity (or payload) [5]. A data hiding scheme, having low image distortion is treated as more secure because it does not raise suspicion among invaders or attackers [1]. Data hiding techniques can be carried out in three domains [7] namely spatial, transform, and compressed domain. In the spatial domain scheme, all the pixel values are directly used and then modified to embed the secret data [11]. In transform domain scheme, the cover image is firstly transformed into frequency coefficients by utilizing a frequency oriented mechanism such as Discrete Cosine Transformation (DCT) [19] or Discrete Wavelet Transformation (DWT) [18]. 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 Joint Photographic Experts Group (JPEG) [17], Vector Quantization (VQ) [12], Block Truncation Coding (BTC) [2] etc., are more preferable when the media is image or video. One of the most effective compression techniques is BTC. Furthermore, the BTC has been improved using Absolute Moment Block Truncation Coding (AMBTC) [8] in which instead of using the standard deviation the first absolute moment is preserved along with the mean. It 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, 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 and Chang [4] developed a BTC based data embedding scheme. This scheme uses a predefined threshold to classify BTC-encoded blocks either in smooth or complex. It then embeds secret data into the bitmap of selected BTC-encoded blocks. This scheme is limited to gray scale images only and the achieved stego image is of low quality. Chang *et al*. [2] extend Chuang and Chang [4] method for color images compressed. It applies BTC to each block of a cover image to get three pairs of quantization levels and three bitmaps. 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] difference expansion based scheme to embed the secret data. This scheme provides better quality stego-image. Li *et al*. [9] 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. However, this technique has low data hiding capacity. To further improve the data hiding capacity, Sun *et al*. [16] discussed a joint neighbor coding based data hiding technique. It utilizes both high mean and low mean table for embedding the secret data. It has been observed that this scheme obtains a very high data hiding, however, it requires extra data to be included in the stego-code stream. Later, Zhang *et al*. [20] introduced a reversible data hiding scheme for BTC-compressed images by further losslessly encoding the BTC-compressed data according to the secret bits. 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. Lin *et al*. [10] 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. Pan *et al*. [14] 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 and Sun [13] 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. In this paper, we extend the work of Ou and Sun [13] to embed the larger amount of secret data. Our aim in this paper is to provide both good quality of 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. Secret data bits are equally embedded in both smooth and complex blocks resulting in improved embedding capacity and also able to maintain a better image quality as compared with other existing schemes Ou and Sun [13] and Chuang and Chang [4].

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 uses a user-defined threshold value (which is adjustable as per the requirement is employed,) 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 two quantization levels are re-calculated using Equations (1) and (2) so that caused distortion is minimized. While for complex blocks, the bit plane is reconstructed in which every pixel is represented by two bits instead of representation by one bit (as done usually). Here, the secret data is embedded into the first LSB of the bit plane. Finally, four new quantization levels are calculated using Equations (3), (4), (5), and (6) 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 the equal amount of secret data into the complex blocks as in the smooth block which in turn increases the embedding capacity without deteriorating the image quality. The algorithm of the proposed scheme is discussed in the following section.

3.1. Embedding Algorithm

Algorithm 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_1 *and* b_1*, and a bit plane* B_i *.*

Step 3: Calculate absolute difference value D_i *, such that* $D_i = |a_i - b_i|$ *.*

Step 4: If $D_i \le thr$ *, means* IB_i *is a smooth block. Replace bits of the bitplane* B_i *with the 16 bits* S_i *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 5: Calculate two new quantization levels $a_i^{'}$ *and* $b_i^{'}$ *with respect to new bitplane* $B_i^{'}$ *using Equations (1) and (2) respectively, so that smoothness of the block is maintained.*

$$a_i^{'} = \frac{1}{M-q} \sum_{x_i \varepsilon G_0} x_i \tag{1}$$

$$b_i^{'} = \frac{1}{q} \sum_{x_i \varepsilon G_1} x_i \tag{2}$$

Where, M *is the total number of pixels in the block* IB_i *, q is the number of 1's in the bit plane.*

Step 6: If $|a_i^{'} - b_i^{'} \le thr|$ *indicates that the smoothness of the block is maintained. Then add* $\{a_i^{'}, b_i^{'}, B_i^{'}\}$ *into* I_S *. If* $|a_i^{'} - b_i^{'}| > thr$ *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* $\{a_i, b_i, B_i^{'}\}$ *Go back to the Step 3 and process the next block.*

Step 7: If $D_i > thr$*, the block* IB_i *is classified as a complex block, which is again processed to get four quantization levels instead of just two so that noise 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 bitplane.*
- *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 8: Embed the next 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: Calculate four new quantization levels as follows to reduce the distortion.

$$a_i^{''} = \frac{1}{q_{00}} \sum_{x_i \varepsilon G_{00}} x_i \tag{3}$$

$$d_i^{''} = \frac{1}{q_{01}} \sum_{x_i \varepsilon G_{01}} x_i \tag{4}$$

$$c_i^{''} = \frac{1}{q_{10}} \sum_{x_i \varepsilon G_{10}} x_i \tag{5}$$

$$b_i^{''} = \frac{1}{q_{11}} \sum_{x_i \varepsilon G_{11}} x_i \tag{6}$$

Where, $q_{00}, q_{01}, q_{10},$ *and* q_{11} *are the number of 00, 01, 10, and 11 in the bit plane, respectively.*

Step 10: If $|a_i^{''} - b_i^{''}| > thr$ *, indicates that new quantization levels can maintain the complexness property of the block* IB_i *, then add* $\{a_i^{''}, b_i^{''}, c_i^{''}, d_i^{''}, B_i^{''}\}$ *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* $\{a_i, b_i, c_i^{''}, d_i^{''}, B_i^{''}\}$*. 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 Figure 1. Let the secret data bit stream *S* is $(10100101001011100101000010 10)_2$ 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, 81, 76, 75, 70, 77, 80, 81, 80, 81, 81, 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 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. To embed the secret data in smooth block, the bits of the bit plane IB_1 are replaced with the 16 bits of the secret data i.e., $S_1=(1010, 0101, 0010, 1110)_2$, 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 and b_1 are calculated so that caused distortion is reduced with respect to the new bit plane $B_1^{'}$ using Equations (1) and (2). As, the absolute difference $|a_1^{'} - b_1^{'}| = |80-82| = (2 < thr)$, that means smoothness property of the block IB_1 is maintained so new stego compressed code is $\{a_1^{'}, b_1^{'}, B_1^{'}\} = \{80, 82, B_1^{'}\}$ For the block IB_2, absolute difference value D_2 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=(1110,0101,0000,1010)$_2$, 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 $a_2^{''}, d_2^{''}, c_2^{''}$, and $B_2^{''}$ with respect to new bit plane $B_2^{''}$ are calculated so that the caused distortion is limited using Equations (3), (4), (5), and (6). 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 $\{a_2^{''}, b_2^{''}, c_2^{''}, d_2^{''}, B_2^{''}\} = \{187, 207, 207, 190, B_2^{''}\}$

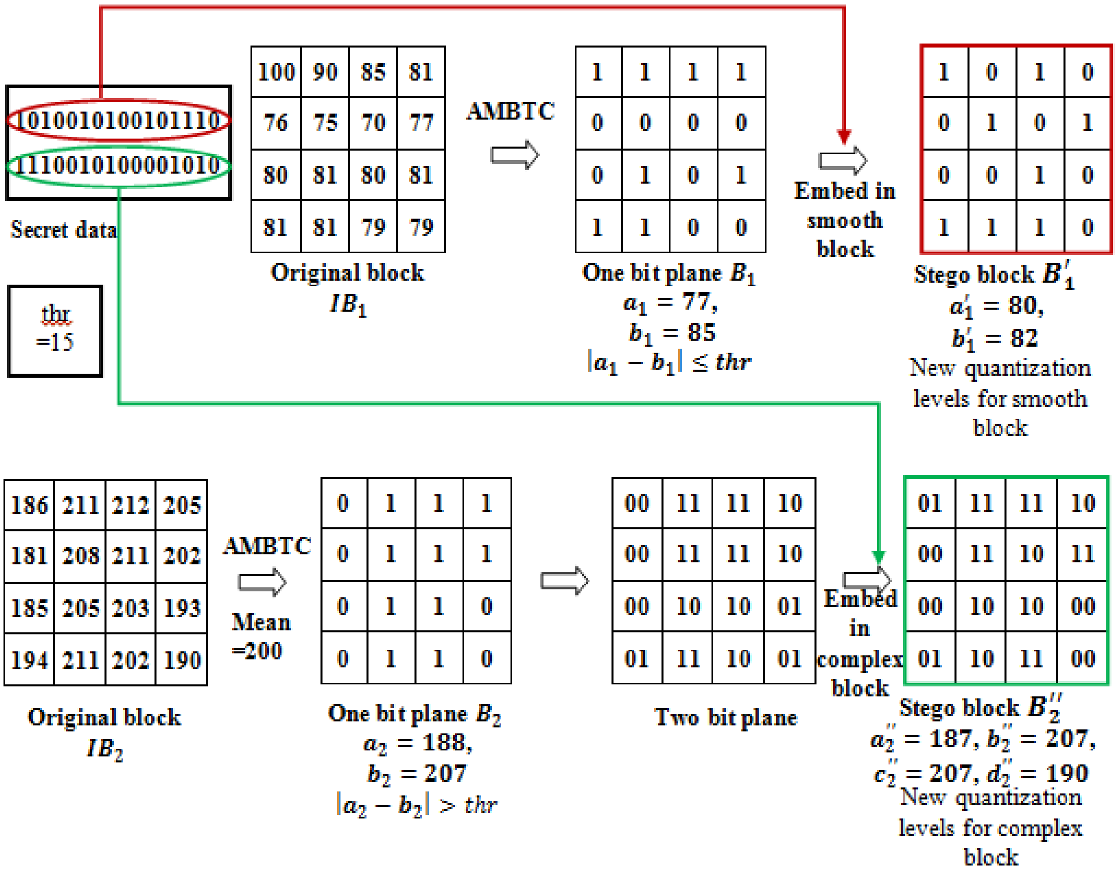


Figure 1. An example to explain the proposed embedding process.

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.

Algorithm 2: Extraction Algorithm

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 16 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 bitplane $B_2^{''}$ as S_1 and add the S_1 into S_D.

Step 4: Go to step 1 until all the codes are processed.

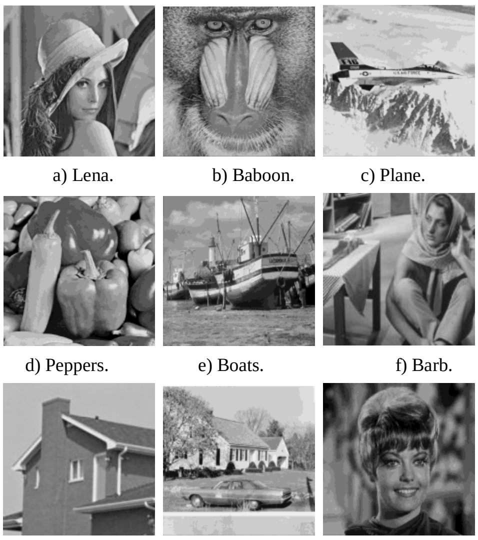
Thus proceeding, we can extract the complete secret data bit stream S_D from the AMBTC compressed stego codes.

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 $|a_1^{'} - b_1^{'}| = |80-82| = (2 < thr)$, which is less than the defined threshold value. It means that the block $B_1^{'}$ is a smooth block. To extract the secret data in case of smooth block, the bits of the bitplane $B_1^{'}$ are extracted. Thus, 16 bits (S_1=(1010,0101,0010,1110)$_2$) is obtained from the bit plane of the block $B_1^{'}$ and these 16 bits are added into S_D. For the second compressed code, the absolute difference value D_2 of two quantization levels $a_i^{''}$ and $b_i^{''}$ is calculated, such that $\left|a_1^{''} - b_1^{''}\right| = |187 - 207| = (20 < thr)$, which is greater than the threshold value (i.e., 15) means the block $B_2^{''}$ is a complex block. To extract the secret data in case of complex block, the first LSB of every pixel from the two bitplane $B_2^{''}$ is extracted. Thus, 16 bits (S_1=(1110,0101,0000,1010)$_2$) 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 [4, 13]. 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 Figure 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 Mean Structural Similarity Index Measure (MSSIM) are used to evaluate the performance of existing schemes and the proposed scheme.



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

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

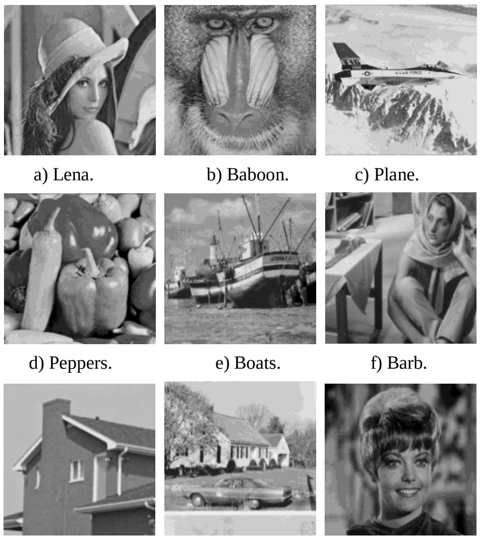
The hiding capacity refers to the number of secret data bits hidden into the cover image. The PSNR and MSSIM are the image quality metrics. The PSNR is measured in dB and is given by:

$$PSNR = 10\log_{10}\left[\frac{255\times 255}{\frac{1}{N\times N}\sum_{i=1}^{N}\sum_{j=1}^{N}(x_{ij}-y_{ij})^2}\right] \tag{7}$$

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 $N\times N$ pixels, respectively. The MSSIM also measures the similarity between the cover image and the stego-image. The value of MSSIM always varies between the interval [0, 1]. The value 0 indicates that both the images are totally unrelated and 1 indicates that the both are exactly the same. It is given as:

$$MSSIM = \frac{1}{N}\sum_{j=1}^{N}\frac{(2\mu_{x_j}\mu_{y_j}+c_1)(2\sigma_{x_jy_j}+c_2)}{(\mu_{x_j}^{\ 2}+\mu_{y_j}^{\ 2}+c_1)(\sigma_{x_j}^{\ 2}+\sigma_{y_j}^{\ 2}+c_2)} \tag{8}$$

where, μ_{xj} and $\sigma_{x_j}^{\ 2}$ are the average and variance of cover image x_j; μ_{y_j} and $\sigma_{y_j}^{\ 2}$ are the average and variance of stego-image y_j; $\sigma_{x_jy_j}$ is the covariance of x_1 and x_j; $c_1=(k_1L)^2$ and $c_2=(k_2L)^2$ are two variables to stabilize the division with weak denominator; L is the dynamic range of the pixel-values; K_1=0.01 and K_2=0.03 are constants.



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

Figure 3. Stego-images of the proposed scheme.

To critically analyze the performance of our proposed algorithm, we have compared the performance with some of the most similar work carried out in literature i.e., Ou and Sun [13] and Chuang and Chang [4]. Furthermore, the results are taken for both the best quality (thr1=10) and the maximum capacity (at thr2=50) for all the methods. Basically, these two thresholds i.e., thr1=10, and thr2=50, give the best quality stego-image and highest data hiding capacity, respectively, for traditional schemes. Our scheme is an exception to the standard trade-off of data hiding because it has fixed data hiding capacity i.e., its maximum capacity, irrespective of the cover image characteristics and user-defined threshold value. It gives best quality stego-image in terms of PSNR and MSSIM. It provides maximum capacity at thr 10 whereas Ou and Sun [13] and Chuang and Chang [4] methods only give the best quality which leads to approximately minimum capacity. To get maximum capacity, Ou and Sun [13] and Chuang and Chang [4] methods increase the threshold value (thr) which in turn deteriorates the stego-image quality; however, our scheme gives the best quality stego-image as shown in Figure 3 even with the highest data hiding capacity. The embedding capacity, PSNR, and MSSIM for each image with a threshold (*thr*=10) and (*thr*=50) are given in Table 1. From the Table I, it is clearly evident that our scheme is superior to Ou and Sun [13] and Chuang and Chang [4] methods in terms of both image quality and data hiding capacity, irrespective of the cover image. For the noisy images like baboon which have more complex blocks than the smooth blocks, the Ou and Sun [13] method crumbles as it can only hide 50374 bits having only 26.92 dB PSNR at thr=10, whereas, the proposed scheme is able to hide the same amount of secret data as in the smoother images i.e., 262144 bits, with 27.76 dB PSNR and MSSIM 0.9989 (which is very close to 1). Therefore, it achieves approximately 420% increase in the embedding capacity with 3.12 and 0.22% increase in PSNR and MSSIM respectively. At thr=50, the approximate increase in the embedding capacity, PSNR, and MSSIM is 28.30, 15.13, and 8.12% respectively. In case of smooth images like Lena, our scheme is able to hide approximately 50% more secret data bits with an approximate increase of 2.60% in the PSNR value (for some image it might be much more). The proposed scheme is able to achieve minimum 3.52% and 28.30% increase in data hiding capacity, 0.77 and 7.52% in PSNR value and maximum 53% and 420% increase in data hiding capacity, approximately 11% & 15% increase in PSNR value for smooth and complex images respectively. Thus, we can say that the proposed scheme is acceptable in every scenario whereas Ou and Sun [13] method might fail to perform when the cover image is noisy. The proposed scheme provides higher data hiding capacity because it hides one secret data bit into every pixel of the image whereas the Ou and Sun [13] method only hides one bit in every pixel of the smooth blocks, leaving the complex blocks almost untouched. Furthermore, the capacity of the Ou and Sun [13] can never be more than the proposed scheme even if the image contains only the smooth blocks, its capacity will only be equal to the proposed scheme. As far as image quality is concerned, our scheme provides better stego-image quality because in case of complex blocks it minimizes the difference between the quantization levels by having four levels instead of just two so that distortion is reduced. Thus, our scheme performs much better than the existing schemes in terms of both the data hiding capacity and the image quality.

Table 1. Comparison of hiding capacity, PSNR and MSSIM 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					
Chuang and Chang [4]	Max PSNR	32.03	26.85	31.54	32.37	31.04	29.01	33.11	28.78	32.98
	Capacity	166608	36256	172496	178288	141040	112400	175436	128416	167849
	Max MSSIM	0.9812	0.9937	0.9834	0.9772	0.9898	0.9895	0.9845	0.9721	0.9787
Ou and Sun [13] (A)	Max PSNR	32.67	26.92	31.91	33.36	31.32	29.22	33.07	29.88	34.25
	Capacity	172579	50374	178099	183529	148609	121759	181451	136774	171315
	Max MSSIM	0.9899	0.9967	0.9905	0.9888	0.9947	0.9942	0.9897	0.9875	0.9910
Proposed method (B)	PSNR	33.52	27.76	32.41	33.62	31.56	29.91	35.56	31.40	35.25
	Capacity	262144	262144	262144	262144	262144	262144	262144	262144	262144
	MSSIM	0.9971	0.9989	0.9948	0.9904	0.9976	0.9986	0.9954	0.9908	0.9965
Percentage increment ((B-A)/A) %	PSNR	**2.60**	**3.12**	**1.56**	**0.77**	**0.76**	**2.36**	**7.52**	**5.08**	**2.91**
	Capacity	**51.89**	**420.39**	**47.19**	**42.83**	**76.39**	**115.29**	**44.47**	**91.66**	**53.01**
	MSSIM	**0.72**	**0.22**	**0.43**	**0.16**	**0.29**	**0.44**	**0.57**	**0.33**	**0.55**
					Threshold *thr*=50					
Chuang and Chang [4]	PSNR	26.80	22.00	26.93	27.66	25.58	24.17	29.98	26.21	29.07
	Max Capacity	251488	200464	239280	251168	240480	220960	247802	244687	248654
	MSSIM	0.9280	0.8633	0.9457	0.9400	0.9149	0.9049	0.9258	0.9158	0.9158
Ou and Sun [13] (C)	PSNR	29.30	24.11	29.20	30.29	28.00	26.31	30.04	27.05	30.24
	Max Capacity	252154	204319	240709	251854	241834	223534	251022	245847	253215
	MSSIM	0.9580	0.9238	0.9687	0.9673	0.9488	0.9436	0.9478	0.9445	0.9486
Percentage increment ((B-C)/C) %	PSNR	**14.40**	**15.13**	**10.99**	**10.99**	**12.71**	**13.68**	**18.37**	**16.08**	**16.56**
	Capacity	**3.96**	**28.30**	**8.90**	**4.08**	**8.39**	**17.27**	**4.43**	**6.62**	**3.52**
	MSSIM	**4.08**	**8.12**	**2.69**	**2.38**	**5.14**	**5.82**	**5.02**	**4.90**	**5.04**

5. Conclusions

In this paper, we have proposed a high capacity data hiding scheme which modifies the AMBTC compression technique. The scheme identifies smooth and complex blocks and hides secret data equally in both the blocks. It increases the smoothness of the complex block so that caused distortion is reduced. The existing schemes [4, 13] suffer in terms of data hiding capacity when there is noisy cover image like baboon. However, the characteristic of the cover image does not have any impact on the hiding capacity of the proposed scheme. Thus, the proposed scheme provides higher data hiding capacity and better image quality at the same time than the existing techniques.

References

[1] Chang C. and Kieu T., "A Reversible Data Hiding Scheme Using Complementary Embedding Strategy," *Information Sciences*, vol. 180, no. 16, pp. 3045-3058, 2010.

[2] Chang C., Lin C., and Fan Y., "Lossless Data Hiding for Color Images Based on Block Truncation Coding," *Pattern Recognition*, vol. 41, no. 7, pp. 2347-2357, 2008.

[3] Chen J., Hong W., Chen T., and Shiu C., "Steganography for BTC Compressed Images Using no Distortion Technique," *The Imaging Science Journal*, vol. 58, no. 4, pp. 177-185, 2010.

[4] Chuang J. and Chang C., "Using a Simple and Fast Image Compression Algorithm to Hide Secret Information," *International Journal of Computer and Application*, vol. 28, no. 4, pp. 329-333, 2006.

[5] Cox I., Miller M., Bloom J., Fridrich J., and Kalker T., *Digital Watermarking and Steganography*, Morgan Kaufman, 2007.

[6] Khan Z., Shah M., Naeem M., Mahmood T., Khan S., Amin N., and Shahzad D., "Threshold-based Steganography: A Novel Technique for Improved Payload and SNR," *The International Arab Journal of Information Technology*, vol. 13, no. 4, pp. 380-386, 2016.

[7] Langelaar G., Setyawan I., and Lagendijk R., "Watermarking Digital Image and Video Data: a State-of-the Overview," *IEEE Signal Processing Magazine*, vol. 17, no. 5, pp. 20-46, 2000.

[8] Lema M. and Mitchell O., "Absolute Moment Block Truncation Coding and its Application to Color Images," *IEEE Transaction Communication*, vol. 32, no. 10, pp. 1148-1157, 1984.

[9] Li C., Lu Z., and Su Y.,"Reversible Data Hiding for BTC-compressed Images Based on Bitplane Flipping and Histogram Shifting of Mean Tables," *Information Technology Journal*, vol. 10, no. 7, pp. 1421-1426, 2011.

[10] Lin C., Liu X., Tai W., and Yuan S., "A Novel Reversible Data Hiding Scheme Based on AMBTC Compression Technique," *Multimedia Tools Appl*, vol. 74, no. 11, pp. 3823-3842, 2015.

[11] Lin I., Lin Y., and Wang C., "Hiding Data in Spatial Domain Images with Distortion Tolerance," *Computer Standards and Interfaces*, vol. 31, no. 2, pp. 458-464, 2009.

[12] Lu Z., Wang J., and Liu B., "An Improved Lossless Data Hiding Scheme based on Image VQ-Index Residual Value Coding," *Journal of Systems and Software*, vol. 82, no. 6, pp. 1016-024, 2009.

[13] Ou D. and Sun W., "High Payload Image Steganography with Minimum Distortion based on Absolute Moment Block Truncation Coding," *Multimedia Tools and Applications*, vol. 74, no. 21, pp. 9117-9139, 2015.

[14] Pan J., Li W., and Lin C., "Novel Reversible Data Hiding Scheme for AMBTC-Compressed Images by Reference Matrix," *in Proceedings of Multidisciplinary Social Networks Research*, Kaohsiung, pp. 427-436, 2014.

[15] Petitcolas F., Anderson R., and Kuhn M., "Information Hiding-A Survey," *in Proceedings of the IEEE*, vol. 87, no. 7, pp. 1062-1078, 1999.

[16] Sun W., Lu Z., Wen Y., Yu F., and Shen R., "High-performance Reversible Data Hiding for Block Truncation Coding Compressed Images," *Signal Image and Video Processing*, vol. 7, no. 2, pp. 297-306, 2013.

[17] Wang K., Lu Z., and Hu Y., "A High Capacity Lossless Data Hiding Scheme for JPEG Images," *Journal of Systems and Software*, vol. 86, no. 7, pp. 1965-1975, 2013.

[18] Xuan G., Shi Y., Yao Q., Ni Z., Yang C., and Gao J., "Lossless Data Hiding using Histogram Shifting Method based on Integer Wavelets," *International Workshop on Digital Watermarking*, *Lecture Notes in Computer Science*, Jeju Island, pp. 323-332, 2006.

[19] Yang B., Schmucker M., Funk W., Brush C., and Sun S., "Integer DCT-Based Reversible Watermarking for Images Using Companding Technique," *in Proceedings of International Journal of Electronics Communication*, San Jose, pp. 814-826, 2004.

[20] Zhang Y., Guo S., Lu Z., and Luo H., "Reversible Data Hiding for BTC-compressed Images based on Lossless Coding of Mean Tables," *IEICE Transactions on Communications*, vol. 96, no. 2, pp. 624-631, 2013.

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. Currently, she is pursuing her doctoral degree in Computer Science and Engineering, at National Institute of Technology, Jalandhar, Punjab, India. Her research lies in the area of text and image based steganogrography, digital watermarking and image processing.

Geeta Sikka, 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. She is presently working as Associate Professor in the Department of Computer Science and Engineering at National Institute of Technology, Jalandhar. Her research interests are Software Engineering, Databases and Data mining.

Harsh Verma, received his doctorate in Numerical Computing from Punjab Technical University Punjab, India. Currently, he is associate Professor in Computer Science and Engineering, National Institute of Technology, Jalandhar, India. His research interest includes in numerical computing, information security and computer networks. He has published various papers in national and international journal and conferences. He has attended various national and international workshop, training schools, and other technical activity during his academic carrier.