http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet

Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 10 (7): 1421-1426, 2011
ISSN 1812-5638 / DOI: 10.3923/itj.2011.1421.1426

Reversible Data Hiding for Btc-compressed Images Based on Bitplane Flipping and Histogram Shifting of Mean Tables

[1]Cai-Hua Li, [2]Zhe-Ming Lu and [2]Yu-Xin Su
[1]Institute of Engineering Mechanics, China Earthquake Administration, Harbin 150080, China
[2]School of Aeronautics and Astronautics, Zhejiang University, Hangzhou 310027, China

Abstract: Recently, more and more attention have been paid to reversible data hiding techniques for compressed images based on JPEG, JPEG2000, Vector Quantization (VQ) and Block Truncation Coding (BTC). Exiting data hiding schemes in the BTC domain modified the BTC encoding stage or BTC-compressed data according to the secret bits and their embedding capacity was not high and might reduce the image quality. This study introduced the histogram shifting technique to BTC-compressed mean tables to further improve the hiding capacity while maintaining the BTC-compressed image quality. First, the original image was encoded by the BTC technique to obtain the BTC-compressed data which could be represented by a high mean table, a low mean table and a bitplane sequence. Then, the proposed reversible data hiding scheme was performed on the BTC-compressed data. Present hiding scheme contained two main steps, one was the bitplane flipping step that hid secret bits by swapping the high mean and low mean, the other was histogram shifting of the resulting mean tables after swapping. Experimental results showed that our scheme outperformed two existing BTC-based reversible data hiding works, in terms of capacity and efficiency.

Key words: Data hiding, reversible data hiding, block truncation coding, histogram shifting, bitplane flipping

INTRODUCTION

Data hiding (Rabah, 2004) is a technique to embed copyright information or secret information into images, audio, video or 3D meshes without perceivable degradation. By and large, data hiding schemes can be classified into two categories, irreversible data hiding and reversible data hiding. Irreversible data hiding schemes (Xiao *et al*., 2009; Luo *et al*., 2011) can make the hidden data imperceptible but the distortions induced in the host signal are inevitable and irreversible. However, in reversible data hiding schemes (Hong *et al*., 2009, 2010), the secret data are embedded in a lossless manner such that one can completely recover both the hidden data and the original host signal. In some special application fields such as military, medical and forensic areas, it is necessary to recover the host image with no distortion after the extraction of hidden data. Therefore, reversible data hiding has become a hot research branch now-a-days.

Reversible data hiding for images can be classified into spatial-domain schemes and compressed-domain schemes. The first reversible data hiding scheme in the spatial domain includes an encoding method that embeds the authentication stamp in the digital block and a decoding method that retrieves the meta-data from an authenticated digital block and allows restoration of the original data block if desired (Barton, 1997). The generalized LSB (Least Significant Bit) modification based reversible data hiding algorithm (Celik *et al*., 2002) can losslessly recover the cover signal by compressing distortion-susceptible portions of the signal and transmitting these compressed data as a part of the embedded payload. The famous difference expansion based reversible data embedding method (Tian, 2003) explores the redundancy in the digital content to achieve the reversibility. However, the main drawback is the unforeseen but necessary location map. The reversible watermarking scheme proposed by Thodi and Rodriguez (2004) utilizes the correlation inherent among the neighboring pixels, where data embedding is done by expanding the prediction-error values. The high-capacity reversible watermarking algorithm for colored images proposed by Alattar (2004) first calculates the differences between every two neighboring pixels in a quad in a predefined order and then hides triplets of bits in the difference expansion of quads of adjacent pixels.

Since most digital images are stored in compressed forms, such as JPEG, JPEG2000, Vector Quantization (VQ) and Block Truncation Coding (BTC), it is necessary to research and develop compressed-domain reversible data

Corresponding Author: Zhe-Ming Lu, Teaching Building 5, Yuquan Campus, Zhejiang University, 38 ZheDa Road, Hangzhou 310027, P.R. China

hiding techniques. Here, the reversibility indicates that the compressed image can be completely recovered after the extraction of hidden data. That is, not the original image but its compressed version serves as the host signal. BTC is an efficient lossy image compression technique (Lu *et al*., 2002) that was first proposed by Delp and Mitchell (1979). It divides the original images into blocks and then uses a quantizer to reduce the number of grey levels in each block whilst maintaining the same mean and standard deviation. Another variation of BTC is Absolute Moment Block Truncation Coding (AMBTC) (Hong *et al*., 2008) that was first proposed by Lema and Mitchell (1984) in which instead of using the standard deviation the first absolute moment is preserved along with the mean. In the original AMBTC method, the M×N sized 256-grayscale image X is first divided into many non-overlapping m×n-sized blocks, i.e., $X = \{x^{(ij)}, 1 \le i \le M/m, 1 \le j \le N/n\}$. The pixels in each block are individually quantized into two-level outputs, where the mean value of pixels in each block $x^{(ij)} = \{x^{(ij)}_{uv}, 1 \le u \le m, 1 \le v \le n\}$ is taken as the one-bit quantizer threshold t_{ij}, i.e.,:

$$t_{ij} = \frac{1}{m \times n} \sum_{u=1}^{m} \sum_{v=1}^{n} x^{(ij)}_{uv} \tag{1}$$

The two output quantization levels can be calculated as follows:

$$l_{ij} = \begin{cases} \dfrac{1}{m \times n - q_{ij}} \sum\limits_{x^{(ij)}_{uv} < t_{ij}} x^{(ij)}_{uv} & q_{ij} < m \times n \\ t_{ij} & q_{ij} = m \times n \end{cases} \tag{2}$$

$$h_{ij} = \frac{1}{q_{ij}} \sum_{x^{(ij)}_{uv} \ge t_{ij}} x^{(ij)}_{uv} \tag{3}$$

where, l_{ij} and h_{ij} denote the low mean and the high mean of block $x^{(ij)}$, respectively q_{ij} stands for the number of pixels that are not less than the mean value t_{ij}. If $q_{ij} = m \times n$, we have $l_{ij} = h_{ij} = t_{ij}$. Then a two-level quantization is performed for all pixels in the block to form a bitplane p_{ij} such that 0 is stored for the pixels whose values are less than the mean and the rest of the pixels are presented by 1. The image is reconstructed at the decoding phase from the bitplane by assigning the value l_{ij} to 0 and h_{ij} to 1. Thus a compressed block appears as a triple (l_{ij}, h_{ij}, p_{ij}), where l_{ij}, h_{ij} and p_{ij} denote the low mean, the high mean and the bitplane of block $x^{(ij)}$, respectively. Figure 1 gives an example of encoding and decoding an image block based on AMBTC. In fact, if we gather all high means of all blocks, we can obtain a matrix that is called high mean table $H = \{h_{ij}, 1 \le i \le M/m, 1 \le j \le N/n\}$. Similarly, we can obtain the low mean table $L = \{l_{ij}, 1 \le i \le M/m, 1 \le j \le N/n\}$ by gathering all low means while all bitplanes can construct a bitplane sequence $P = \{p_{ij}, 1 \le i \le M/m, 1 \le j \le N/n\}$. AMBTC is computationally simpler than BTC. Although the original BTC is a fast encoding scheme, the bit rate (typically 2 bpp) is much higher than JPEG and VQ. In order to reduce the bit-rate many techniques have been introduced in BTC, such as median filtering, adaptive coding (Nasiopoulos *et al*., 1991), DCT-BTC (Wu and Coll, 1991) and VQ (Mohamed and Fahmy, 1995).

Original $x^{(ij)}$

$$\begin{bmatrix} 2 & 9 & 12 & 15 \\ 2 & 11 & 11 & 9 \\ 2 & 3 & 12 & 15 \\ 3 & 3 & 4 & 14 \end{bmatrix}$$

$t_{ij} = 7.84$

Bitplane p_{ij}

$$\begin{bmatrix} 0 & 1 & 1 & 1 \\ 0 & 1 & 1 & 1 \\ 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$q_{ij} = 9$

Reconstructed

$$\begin{bmatrix} 3 & 12 & 12 & 12 \\ 3 & 12 & 12 & 12 \\ 3 & 3 & 12 & 12 \\ 3 & 3 & 3 & 12 \end{bmatrix}$$

$l_{ij} = 3$
$h_{ij} = 12$

Fig. 1: An example of encoding a block $x^{(ij)}$ by the triple (l_{ij}, h_{ij}, p_{ij})

In the past ten years, several data hiding schemes for BTC compressed gray-level images have been proposed. The first work was proposed by Lu *et al.* (2002), where the robust watermark is embedded by modifying the VQ-BTC encoding process according to the watermark bits. In study of Lin and Chang (2004), a data hiding scheme for BTC compressed images was proposed by performing LSB substitution operations on BTC high and low means and performing the minimum distortion algorithm on BTC bitplanes. In study of Chuang and Chang (2006), a hiding scheme was proposed to embed data in the BTC bitplanes of smooth regions. However, above three works are not reversible. The first reversible data hiding scheme for BTC-compressed gray-level images was proposed by Hong *et al.* (2008). As we know, if we exchange l_{ij} and h_{ij} in the compressed triple of block $x^{(ij)}$, we only need to flip the bitplane p_{ij} into $\bar{p}_{ij}$ in order to obtain the same reconstructed block. Based on this idea, the embedding process of Hong *et al.* (2008) scheme can be illustrated as follows: Firstly, each image block is compressed by AMBTC resulting in the compressed codes (l_{ij}, h_{ij}, p_{ij}). Secondly, all embeddable blocks with $l_{ij}<h_{ij}$ are found, that is to say, the block with $l_{ij} \le h_{ij}$ is non-embeddable. Thirdly, for each embeddable block, if the bit to be embedded is 1, then the compressed code is changed from (l_{ij}, h_{ij}, p_{ij}) into (h_{ij}, l_{ij}, $\bar{p}_{ij}$). Otherwise, if the bit to be embedded is 0, then no operation is required. In other words, the secret bit 0 corresponds to the code (l_{ij}, h_{ij}, p_{ij}) and the secret bit 1 corresponds to the code (h_{ij}, l_{ij}, $\bar{p}_{ij}$). The secret data extraction process is very simple. Assume we receive the code (l'_{ij}, h'_{ij}, p'_{ij}), we only need to judge the relationship between l'_{ij} and h'_{ij}. If $l'_{ij}>h'_{ij}$, then the secret bit is 1; Else if $l'_{ij}<h'_{ij}$, the secret bit is 0; Otherwise, no secret bit is embedded. Although Hong *et al.* (2008) scheme is reversible, it doesn't consider hiding data in the blocks with $l_{ij} = h_{ij}$. To overcome this problem, an improved reversible hiding scheme (Chen *et al.*, 2010) has been proposed recently. To embed secret data in the blocks with $l_{ij} = h_{ij}$, Chen *et al.* (2010) introduced Chuang and Chang's bitplane replacement idea (Chuang and Chang, 2006) to deal with the case $l_{ij} = h_{ij}$ in Hong *et al.*'s bitplane flipping scheme. The embedding process of Chen *et al.* (2010) scheme can be illustrated as follows: Each image block is compressed by AMBTC resulting in the compressed codes (l_{ij}, h_{ij}, p_{ij}). For each block with $l_{ij}<h_{ij}$, if the bit to be embedded is 1, then the compressed code is changed from (l_{ij}, h_{ij}, p_{ij}) into (h_{ij}, l_{ij}, $\bar{p}_{ij}$). Otherwise, if the bit to be embedded is 0, then no operation is required. For the block with $l_{ij}=h_{ij}$, since the bitplane has no use in the reconstruction process, the whole bitplane can be replaced with m×n secret bits. The secret data extraction process is also very simple. Assume we receive the code (l'_{ij}, h'_{ij}, p'_{ij}), we only need to judge the relationship between l'_{ij} and h'_{ij}. If $l'_{ij}>h'_{ij}$, then the secret bit is 1; Else if $l'_{ij}<h'_{ij}$, the secret bit is 0; Otherwise, all m×n bits in the bitplane p'_{ij} are extracted as the secret bits.

From above, we can see that, existing data hiding schemes in the BTC domain have low capacity and may reduce the image quality. To both increase the hiding capacity and achieve the reversibility, this study has proposed an improved reversible data hiding for BTC-compressed images by further introducing the histogram shifting technique to BTC compressed data. Experimental results and a comparison with Hong *et al.* (2008) reversible hiding algorithm and Chen *et al.* (2010) reversible hiding algorithm have demonstrated the superiority of our proposed scheme.

THE PROPOSED SCHEME

we can see that Hong *et al.* (2010) bitplane flipping scheme and Chen *et al.* (2010) scheme can embed 1 bit information per block if there is no block with $l_{ij} = h_{ij}$. In order to embed more secret bits, after using Chen *et al.* (2010) scheme, we can further compose all high/low means as a high/low mean table and then introduce the histogram shifting technique to high and low mean tables. The proposed algorithm consists of three stages, i.e., the AMBTC compression stage, the data hiding stage and the data extraction and image recovery stage which can be illustrated as follows.

The AMBTC compression stage: This stage is a preprocessing stage before data hiding. The aim of this stage is to obtain the two mean tables and the bitplane sequence for later use. Given an M×N-sized 256 grayscale original image X, this stage can be expressed as follows:

- **Step 1:** The original image X is divided into non-overlapping m×n-sized blocks $x^{(ij)}$, I = 1, 2, ..., M/m, j = 1, 2, ..., N/n
- **Step 2:** Encode each image block $x^{(ij)}$ by AMBTC, obtaining the compressed triple (l_{ij}, h_{ij}, p_{ij})
- **Step 3:** Compose all high means and low means to obtain the high mean table H = {h_{ij} |i = 1,2,..., M/m; j = 0,1,..., N/n} and the low mean table L = (l_{ij} |i = 1,2,..., M/m; j = 0,1,..., N/n), respectively. Similarly, compose all bitplanes to obtain the bitplane sequence P = {p_{ij} |i = 1,2,..., M/m; j = 0,1,..., N/n}

The data hiding stage: With the mean tables H and L and the bitplane sequence P in hand, now we can introduce our reversible data hiding algorithm. Assume the secret bit sequence to be embedded is W = {w_1,w_2,...}. Before embedding, we perform the permutation operation on the secret bit sequence in order to enhance the security. Our embedding method is performed in two main steps, i.e., bitplane flipping and histogram shifting which can be illustrated detailedly as follows:

- **Step 1:** We first perform the bitplane flipping technique on H, L and P to embed the first part of secret bits. For each image block with $l_{ij}<h_{ij}$, if the bit to be embedded is 1, then the element h_{ij} in H and the element l_{ij} in L are swapped and the bitplane p_{ij} in P is replaced by $\bar{p}_{ij}$. Otherwise, if the bit to be embedded is 0, then no operation is required. For the block with $l_{ij} = h_{ij}$, since the bitplane has no use in the reconstruction process, the whole bitplane p_{ij} in P can be replaced with m×n secret bits. After this step, we get the modified mean tables and bitplane sequence denoted as H', L' and P', respectively
- **Step 2:** We further perform the histogram shifting technique on H' and L', respectively to embed the second part of secret bits. The detailed sub-steps can be illustrated as follows:
- **Step 2.1:** Generate the histogram from H'
- **Step 2.2:** In the histogram, we first find a zero point and then a peak point. A zero point corresponds to the grayscale value v which doesn't exist in the given image. A peak point corresponds to the grayscale value u which has the maximum number of pixels in the given image
- **Step 2.3:** The whole mean table is scanned in a sequential order. Assume u<v, the grayscale value of pixels between u (including u) and v (including v) is

incremented by 1. This step is equivalent to shifting the range of the histogram, (u, v), to the right-hand side by 1 unit, leaving the grayscale value u empty

- **Step 2.4:** The whole mean table is scanned once again in the same sequential order. Once a pixel with grayscale value of u is encountered, we check the secret bit to be embedded. If this bit is binary 1, the pixel value is incremented by 1. Otherwise, the pixel value remains intact
- **Step 2.5:** After above sub-steps, we can get the final marked high mean table H''. Similarly, perform above sub-steps on the low mean table L' to get the final marked low mean table L'. Note that we should record the u, v values for H and L, respectively as the overhead information

The decoding and extracting stage: Present data hiding scheme is reversible because we can recover the original mean tables and the bitplane sequence after data extraction and thus the original BTC-compressed image can be losslessly recovered. Given the marked mean tables H^w and L^w and the marked bitplane sequence P^w, our purpose is to extract the secret bit sequence and recover the original BTC compressed image, the extraction process as follows:

- **Step 1:** Perform the reverse histogram shifting technique on H^w and L^w to extract the second part of secret bits and get the intermediate result H^r and L^r, respectively. The detailed sub-steps can be illustrated as follows:
- **Step 1.1:** Scan the marked table H^w in the same sequential order as that used in the embedding procedure. If a pixel with its grayscale value v + 1 is encountered, the secret bit 1 is extracted. If a pixel with its value v is encountered, a secret bit 0 is extracted
- **Step 1.2:** Scan the image again, for any pixel whose grayscale value is in the interval (u, v), the pixel value is subtracted by 1
- **Step 1.3:** After above sub-steps, we can get the intermediate high mean table H^r. Similarly, perform above sub-steps on the marked low mean table L^w to get the intermediate low mean table L^r
- **Step 2:** Perform the reverse bitplane flipping technique on H^r, L^r and P^w to extract the first part of secret bits and recover the BTC-compressed image data. For each triple (l^r_{ij}, h^r_{ij}, p^w_{ij}), we only need to judge the relationship between l^r_{ij} and h^r_{ij}. If $l^r_{ij}>h^r_{ij}$, then the secret bit is 1 and l^r_{ij} and h^r_{ij} are swapped and the bitplane p^w_{ij} is replaced by $\bar{p}_{ij}$; Else if $l^r_{ij}<h^r_{ij}$, the secret bit is 0; Otherwise, all m×n bits in the bitplane p^w_{ij} are extracted as the secret bits, replace the bitplane p^w_{ij} with m×n '1' s

Based on above two steps, we can extract all secret bits and reconstruct the original BTC-compressed image, thus the whole reversible process is realized. Obviously, the improved scheme can increase the payload compared with Chen *et al.* (2010) scheme.

RESULTS AND DISCUSSION

To evaluate the proposed scheme, we use six test images, Lena, Peppers, Bridge, Boat, Goldhill and Jet-F16, of the same size 512×512 with 256 grayscales, as shown in Fig. 2 (a-f). Comparisons among our algorithm, Hong *et al.* (2009) algorithm and Chen *et al.* (2010) algorithm are performed under the same block size 4×4. Table 1 lists the number of image blocks whose low mean equals its high mean (i.e., l_{ij} = h_{ij}) for different images. From Table 1, we can see that, for Lena, Boat, Goldhill and Jet-F16 images, there is no image block with l_{ij} = h_{ij}, thus both Hong *et al.* (2009) algorithm and Chen *et al.* (2010) algorithm can embed 16384 bits in each 512×512 image. While for the Bridge image and the Peppers image, since there are some blocks with l_{ij} = h_{ij}, Chen *et al.* (2010) algorithm can embed more bits while Hong *et al.* (2009) algorithm can embed less bits.

To show the superiority of our proposed algorithm, we compare it with Hong *et al.* (2009) scheme and Chen *et al.* (2010) scheme. Three aspects of performance are adopted in the experiments to evaluate a data hiding scheme, i.e., the capacity representing the maximum number of secret bits that can be hidden, the Peak Signal to Noise Ratio (PSNR) representing the quality of the stego-image and the embedding efficiency indicating the number of embedded secret data when a bit of the binary code stream has been transmitted. Obviously, the embedding efficiency can be calculated as Capacity/(Bitrate×M×N). As shown in Table 2, the PSNRs of stego images based on present, Hong *et al.* (2010) and Chen *et al.* (2010) schemes are exactly the same as those of the original AMBTC-compressed images, the reason is that these three schemes are reversible.

Table 1: The number of 4×4 pixel blocks whose high mean equals its low mean (denoted by EB)

Image	MB
Lena	0
Peppers	285
Bridge	54
Boat	0
Goldhill	0
Jet-F16	0

Table 2: Comparisons of the proposed, Hong *et al*. (2010) and Chen *et al*. (2010) schemes (m×n = 4×4)

Algorithm	Performance	Lena	Peppers	Bridge	Boat	Goldhill	Jet-F16
AMBTC	PSNR (dB)	32.041	31.595	28.585	31.151	33.163	31.033
	Bit rate (bpp)	2.000	2.000	2.000	2.000	2.000	2.000
Proposed	PSNR (dB)	32.041	31.595	28.585	31.151	33.163	31.033
	Capacity (bits)	16684	21969	17548	17061	16770	17375
	Efficiency	0.032	0.042	0.034	0.033	0.032	0.033
Hong *et al*. (2010)	PSNR (dB)	32.041	31.595	28.585	31.151	33.163	31.033
	Capacity (bits)	16384	16099	16330	16384	16384	16384
	Efficiency	0.031	0.030	0.031	0.031	0.031	0.031
Chen *et al*. (2010)	PSNR (dB)	32.041	31.595	28.585	31.151	33.163	31.033
	Capacity (bits)	16384	20659	17194	16384	16384	16384
	Efficiency	0.031	0.039	0.033	0.031	0.031	0.031

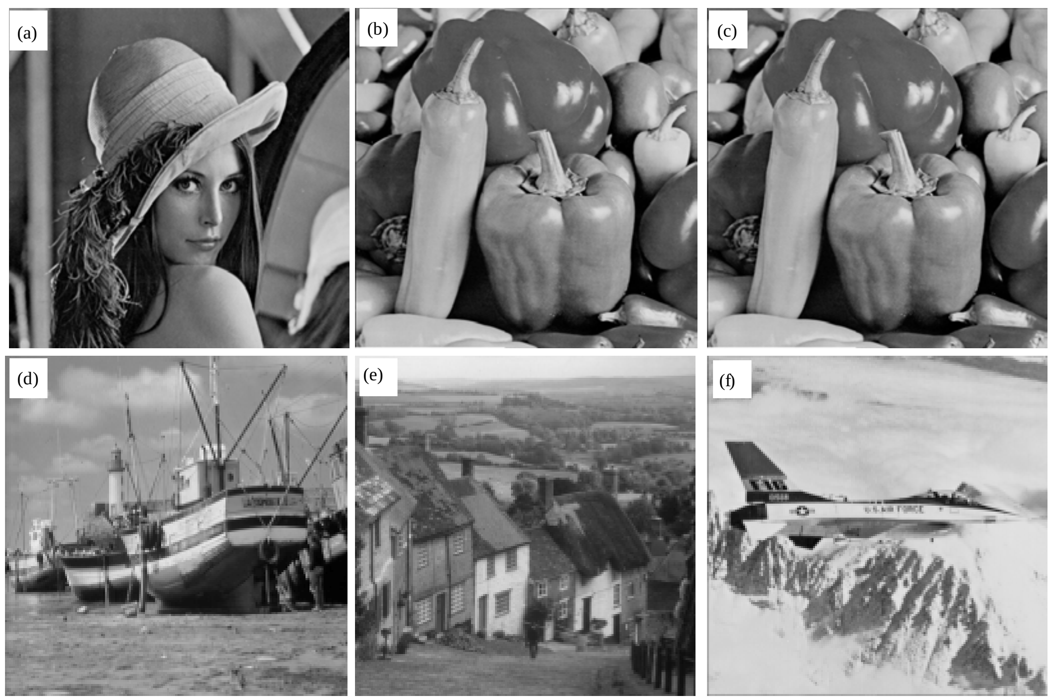

Fig. 2: (a-f) Six test images

With respect to the embedding capacity, the proposed scheme achieves the highest embedding capacity. Compared with other two schemes, our scheme can embed more secret bits because we adopt the histogram shifting technique in addition to Chen *et al*. (2010) scheme. The capacity of Hong *et al*. (2010) and Chen *et al*. (2010) schemes is normally 1 bits in each 4×4 pixel block. However, if there are some blocks with the same high and low mean, then Hong *et al*. (2010) scheme will have less capacity and Chen *et al*. (2010) scheme will have more capacity. As shown in Table 1 and 2, for the Peppers image, because EB = 285, Hong *et al*. (2010) scheme can only embed 16384-285 = 16099 bits while Chen *et al*. (2010) scheme can embed 16384-285 + 285×4×4 = 20659 bits.

In terms of embedding efficiency, present scheme has the highest efficiency values. This means that our scheme can transmit the most number of embedded secret data when a bit of the binary code stream has been transmitted. Taking the above three attributes into comprehensive consideration, the proposed scheme is a more effective method for its high capacity, high PSNR and high embedding efficiency.

CONCLUSIONS

In this study, we present an efficient reversible data hiding algorithm for BTC-compressed images. Our scheme can maintain the same PSNR values as the original AMBTC technique. We embed secret bits in two mean tables and also more secret bits in bitplanes if there are some blocks whose high mean equals its low mean, thus our scheme can obtain higher capacity. Furthermore, our proposed scheme is separated into the AMBTC compression process for generating two mean tables together with one bitplane sequence and the data hiding process to embed secret data into the output codestream which facilitates the individual processing of the encoder and the watermark embedder and the controlling of the corresponding performance.

ACKNOWLEDGMENTS

This study was supported by the National Natural Scientific Foundation of China (Project Number: 61003255). This research project was conducted in China from January 2011 to December 2013.

REFERENCES

Alattar, A.M., 2004. Reversible watermark using difference expansion of quads. Proc. IEEE Int. Conf. Acoustics Speech Signal Process., 3: 377-380.

Barton, J.M., 1997. Method and apparatus for embedding authentication information within digital data. United States Patent, 5646997. http://www. freepatentsonline.com/5646997.html.

Celik, M.U., G. Sharma, A.M. Tekalp and E. Saber, 2002. Reversible data hiding. Proc. IEEE Int. Conf. Image Process., 2: 157-160.

Chen, J., W. Hong, T.S. Chen and C.W. Shiu, 2010. Steganography for BTC compressed images using no distortion technique. Imaging Sci. J., 58: 177-185.

Chuang, J.C. and C.C. Chang, 2006. Using a simple and fast image compression algorithm to hide secret information. Int. J. Comput. Appl., 28: 329-333.

Delp, E. and O. Mitchell, 1979. Image compression using block truncation coding. IEEE Trans. Commun., 27: 1335-1342.

Hong, W.C., T.S. Shiu and C. Wei, 2008. Lossless steganography for AMBTC compressed images. Proc. 1st Int. Congress Image Signal Process., 2: 13-17.

Hong, W., J. Chen and T.S. Chen, 2009. Blockwise reversible data hiding by contrast mapping. Inform. Technol. J., 8: 1287-1291.

Hong, W., T.S. Chen, K. Y. Lin and W.C. Chiang, 2010. A modified histogram shifting based reversible data hiding scheme for high quality images. Inform. Technol. J., 9: 179-183.

Lema, M.D. and O.R. Mitchell, 1984. Absolute moment block truncation coding and its application to color images. IEEE Trans. Commun., 32: 1148-1157.

Lin, M.H. and C.C. Chang, 2004. A novel information hiding scheme based on BTC. Proceedings of the 4th International Conference on Computer and Information Technology, Sept. 14-16, Taichung, Taiwan, pp: 66-71.

Lu, Z., C. Liu and S. Sun, 2002. Digital image watermarking technique based on block truncation coding with vector quantization. Chinese J. Electron., 11: 152-157.

Luo, H., Z. Zhao and Z.M. Lu, 2011. Joint secret sharing and data hiding for block truncation coding compressed image transmission. Inform. Technol. J., 10: 681-685.

Mohamed, S.A. and M.M. Fahmy, 1995. Image compression using VQ-BTC. IEEE Trans. Commun., 43: 2177-2182.

Nasiopoulos, P., R.K. Ward and D.J. Morse, 1991. Adaptive compression coding. IEEE Trans. Commun., 39: 1245-1254.

Rabah, K., 2004. Steganography-the art of hiding data. Inform. Technol. J., 3: 245-269.

Thodi, D.M. and J.J. Rodriguez, 2004. Prediction-error based reversible watermarking. Proc. IEEE Int. Conf. Image Process., 3: 1549-1552.

Tian, J., 2003. Reversible data embedding using a difference expansion. IEEE Trans. Circ. Syst. Video Technol., 13: 890-896.

Wu, Y. and D.C. Coll, 1991. BTC-VQ-DCT hybrid coding of digital images. IEEE Trans. Commun., 39: 1283-1287.

Xiao, X., X. Sun, X. Wang and L. Rao, 2009. DOSM: A data-oriented security model based on information hiding in WSNs. Inform. Technol. J., 8: 678-687.+