A Reversible Data Hiding Scheme Based on Histogram Modification in Integer DWT Domain for BTC Compressed Images

Shun Zhang, Tiegang Gao, Liang Yang
(Corresponding author: Shun Zhang)

College of Software, Nankai University
Weijin Road 94#, Nankai District, Tianjin, China
(Email: shentengvip@gmail.com)
(Received May 30, 2014; revised and accepted Jan. 12 & Apr. 22, 2015)

Abstract

Reversible data hiding is an efficient way for embedding additional data into cover media, which can reversibly recover the original cover media when the additional data is extracted. It has been widely utilized in secure communication and copyright protection. A reversible data hiding scheme imposed on the quantized coefficients of compressed images based on block truncation coding (BTC) is proposed in this paper. Through rearranging the quantized coefficients of BTC images into matrix form, two sampled images are constructed. A histogram modification based reversible data hiding scheme in the integer discrete wavelet transform (integer DWT) domain is adopted on the constructed images. Additional data is embedded into the middle and high frequency sub-bands of the constructed image after integer DWT. Experimental results and analysis have demonstrated that, both higher embedding capacity and lower distortion have been achieved with the proposed scheme compared with existing reversible data hiding schemes for BTC compressed images.

Keywords: Reversible data hiding, Integer DWT, Histogram modification, Block Truncation Coding

1 Introduction

Data hiding is the process that embeds additional data into the cover media while causes distortion as little as possible to cover media. There are two main applications of data hiding. One is secure communication, which is always called steganography. It is considered much safer than traditional encryption, because it conceals the existence of secure communication. The other one is copyright protection and authentication for the cover media, which is often called digital watermarking. Reversible data hiding is the data hiding that can reversibly recover the original cover media after the additional data is extracted. Due to the reversibility, it can be used in a lager field, such as medical image surgery, military imagery, and remote sensing imagery and so on.

Many reversible data hiding schemes have been proposed in recent years [1, 3, 7, 8, 9, 14, 17, 18, 19, 20, 21, 22, 24]. There are there ways to achieve the reversibility. The first one is lossless compressing the cover image to make room for data hiding [7]. The second one is to expand the differences between adjacent pixels to embed the additional data [1, 20]. The last one is to shift the histogram of the cover media and to embed the additional data into the gap of the shifted histogram [3, 9, 14, 18, 19, 21, 22, 24]. Reversible data hiding based on compression makes use of the redundancy of cover image. Therefore, the characters of the cover image limit the capacity and quality of these schemes. Difference expansion based reversible data hiding scheme [20] hides one bit data by extending the difference between two neighbor pixels. And the embedding capacity is improved by extending n-1 pairs of neighbor pixel differences to hide n-1 bits information in [1]. However, the quality of the cover image drops quickly when the embedding capacity increases. Schemes based on histogram modification cause less distortion. However, the obvious drawback of histogram modification based schemes is that the embedding capacity is limited to the peak point of the histogram [18]. Two measures can be applied to increase the embedding capacity: raising the peak points height or increasing the number of peak points of the histogram. Many improved schemes based on the two measures were proposed. Lin et al. [14] adopted a multi-level embedding strategy to increase the capacity. Some prediction difference expansion based schemes were also proposed to generate the histogram for data hiding, which increased the height of the histogram. Tsai et al. [21] proposed a prediction model to get the prediction errors for the histogram modification, which explored the similarity of neighbor pixels. Kim et al. [9] sampled the original image to get a predicted image based on the sampled images. The differences between the predicted image and these sampled images were calculated. Then the histograms of the difference images were generated for the data embedding. Recently, a reversible data hiding method based on histogram modification was proposed in [3]. It divided the cover image into the smooth blocks and complex blocks. Additional data was embedded into the smooth blocks for a higher embedding capacity and lower distortion. A reversible information hiding scheme suitable for embedding small amounts of data was proposed in [17]. It offered flexible embedding capacity and low overhead.

With the development of information technology, more and more multimedia is being produced. The image is one of the most important one. The storage and processing of the raw images are space consuming and resources consuming. Therefore, images are compressed before they are stored and processed in advance. The compression strategies include transformed domain methods and the spatial domain methods. The JPEG compression based on discrete cosine transform (DCT) and the JPEG2000 compression based on discrete wavelet transform (DWT) are transformed domain methods. The vector quantization (VQ) compression [15] and block truncation coding (BTC) compression [6, 11] are the spatial domain methods. Reversible data hiding for compressed images are sometimes of greater importance compared with reversible data hiding on the raw images. The first reversible watermarking scheme for JPEG compressed images was proposed in [7]. After that, a differential energy watermarking (DEW) algorithm for JPEG/MPEG streams, which embedded label bits by selectively discarding high frequency DCT coefficients in certain image regions, was proposed in [10].

BTC [6] compression is a block based lossy image compression technique. It represents the image with many quantized coefficients and one bitmap. Reversible data hiding schemes for BTC compressed images generally utilize the coefficients and the bitmap as the cover media. A genetic algorithm was adopted in [2] to generate an optimal common bitmap. The original three bitmaps of the colored BTC compressed images were replaced with the common bitmap. Additional data was embedded into the common bitmap and the orders of the quantized coefficients. An improved common bitmap was constructed in [5], which improved the embedding capacity. The difference expansion strategy was adopted in [4] to hide the secret data reversibly. Additional data was embedded into the BTC compressed image by swapping sequence of the high mean value and the low mean value in the compressed code in [12]. A histogram constructed from the bitmap of the BTC compressed images was shifted to embed additional data in [13]. The histogram shifting technique was also employed in [16] to embed the secret data into the quantization levels of the compressed codes. However, the embedding capacity is rare in [13, 16]. To increase the embedding capacity and reduce the distortion to the cover image, a well-designed histogram modification based reversible data hiding scheme for BTC compressed images is proposed in the followings.

The scheme utilizes the quantized coefficients of the BTC compressed images to achieve the reversible data hiding. The high mean values and low mean values of every block in the BTC compressed stream can construct two matrixes that are just like two sampled prediction image of the original image. Therefore, some traditional data hiding schemes can be utilized in the design of data hiding scheme for BTC compressed images. This paper imposes the integer DWT on the constructed images, and embeds additional data into the histograms of the middle and high frequency sub-bands in the integer DWT domain. The scheme has achieved high embedding capacity and low distortion in the experiments. Besides, compared with some existing schemes, better performances have been achieved with the proposed scheme.

The rest of the paper is organized as follows. Some related techniques are introduced in Section 2. The main algorithm is proposed in Section 3. Section 4 demonstrates the experimental results and the corresponding analysis, and Section 5 draws the conclusions.

2 Related Works

2.1 Absolute Movement Block Truncation Coding

Block Truncation Coding (BTC) is a simple and efficient way for lossy image compression. Different from those transform domain compression schemes, such as JPEG compression and JPEG 2000, BTC compression is imposed in the spatial domain. It is less time-consuming and more suitable for those real-time applications with low computational ability. The BTC compression transforms an image into a set with two vectors and one bitmap (H, L, BM), where H and L are two vectors with quantized high mean values and low mean values, and BM is a bitmap that indicates which quantized value should be selected in the compressed image. An improved image compression scheme based on BTC, which was called Absolute Movement Block Truncation Coding (AMBTC) was proposed in [11]. Similar to BTC, the AMBTC firstly blocks images into non-overlapping blocks with size$k \times k$. Then every block in the image is represented by a high mean value h, a low mean value l, and a bitmap bm. Suppose an image I with size $m \times n$ is blocked into $k \times k$ sized blocks. For every block $X_i = \{x_j, j = 1, 2, \cdots, k \times k\}, i = 1, 2, \cdots, (m \times n)/(k \times k)$, calculate the mean value:

$$\bar{x}_i = \frac{1}{k \times k} \sum_{j=1}^{k \times k} x_j, \tag{1}$$

where x_j is the j^{th} pixel of the block. Then the low mean value l_i and high mean value h_i are calculated with:

$$l_i = \frac{1}{k \times k - q} \times \sum_{x_j < \bar{x}_i} x_j, \qquad (2)$$

$$h_i = \frac{1}{q} \times \sum_{x_j \geq \bar{x}_i} x_j, \qquad (3)$$

where q is the number of pixels greater or equal to the mean value $\bar{x}_i$ of the block. Then the bitmap of the block bm_i is calculated with:

$$bm_i = \left\{ \begin{array}{ll} 1 & \text{if } x_j \geq \bar{x}_i \\ 0 & \text{otherwise} \end{array} \right\} \qquad (4)$$

Finally, block X_i is represented with (h_i, l_i, bm_i), and all the blocks constitutes the set (H, L, BM). When decompressing the compressed image, every 1 in the bitmap bm_i is replaced by the grey value h_i and every 0 in the bitmap bm_i is replaced by the grey value l_i. An example that compresses one block from image Lena is presented in Figure 1. The block is extracted from pixel values of $(128 : 131, 128 : 131)$ in the standard Lena image. In Figure 1, (a) is the original image block; (b) is the bitmap of the block; (c) is the reconstructed compressed block.

2.2 Reversible Data Hiding in Integer-DWT Domain Based on Histogram Modification

The reversible data hiding scheme proposed in [23] embeds data in the integer-DWT domain, which has achieved both high data embedding capacity and low distortion to the cover image. The histograms of the middle and high frequency sub-bands (LH, HL, HH) after integer DWT are of Laplacian-like distributions [24], which is beneficial to histogram modification based data hiding. Therefore, they are shifted to generate the gap for data hiding. The structure of the image Lena is presented in Figure 2. An example of the histogram modification based data hiding method, which embeds data into the LH subbands of the constructed image, is presented in Figure 3. The generated histogram of sub-band LH is depicted in Figure 3 (a). Then the histogram is shifted to both sides by an embedding strength (Figure 3 (b)). At last, data is embedded by expanding the histogram between and, and the histogram after embedding is as Figure 3 (c).

2.2.1 Reversible data embedding

The histograms of LH, HL, HH sub-bands are generated and data is embedded into the coefficients by histogram modification as presented in Figure 3. For every coefficient in the sub-bands, given an embedding strength parameter q. If $C \geq q$, then C is shifted to $C + q$; else if $C \leq -q$, then C is shifted to $C - q + 1$; else $C \leftarrow 2 \times C + B$, where B is the data to be embedded. The embedding strength parameter q is encoded as the key for data extraction.

2.2.2 Data extraction and reversible recovery of the matrix before embedding

Generate the histograms of middle and high frequency sub-bands and shift these histograms to extract the hidden data. The original coefficients matrixes are reversibly recovered with the following steps. For every coefficient C of LH, HL, HH sub-bands, given an embedding strength parameter q. If $C \geq 2 \times q$, then C is shifted to $C - q$; else if $C \leq -2 \times q + 1$, then C is shifted to $C + q - 1$; else $C \leftarrow floor(C/2)$, and data is extracted: $B = \text{mod}(C, 2)$. All the coefficients of sub-bands LH, HL, HH are reversibly recovered and the extracted B is the data embedded before.

3 Proposed scheme

The BTC compression divides the original image into blocks, and then quantizes the blocks into the high mean values and the low mean values and a bitmap that indicates the quantized values. The quantized high mean values and low mean values of the blocks just construct two sampled images, which are utilized for reversible data hiding. The original image Lena and the sampled image constructed with its high mean values and low mean values after BTC compression are presented in Figure 4. The block size is. Sub-image (a) is the original image Lena with size 512×512; sub-image (b) is the constructed image from the high mean values; and sub-image (c) is the constructed image from the low mean values. Obviously, the size of the two constructed image is 128×128.

The two constructed images is similar to the original image only with a smaller size according to the human visual system. Therefore, traditional reversible data hiding schemes can also be utilized on the constructed images to achieve the reversible data hiding on the BTC compressed images. The efficient reversible data hiding scheme based on the histogram modification in the integer DWT domain is imposed on the constructed images. Suppose the original image I with size $m \times n$ is compressed for the reversible data hiding. Detailed steps of the reversible data hiding scheme for BTC compressed images are as follows:

1) Divide I into $k \times k$ sized blocks $X = \{X_i, i = 1, 2, \cdots, (m \times n)/(k \times k)\}$;

2) Calculate the coefficients of compressed image (H, L, BM) with method given in Section 2.1, where H and L are vectors with size $(1, (m \times n)/(k \times k))$, BM is a binary matrix with size (m, n);

3) Construct the sampled images, denoted as I_h and I_l, respectively with the high mean values vector H and the low mean values vector L;

4) Impose the integer DWT on I_h and I_l to get the four sub-bands LL, LH, HL, HH for data hiding;

192	187	162	162
195	173	172	171
188	171	170	150
185	184	159	140

(a) Original

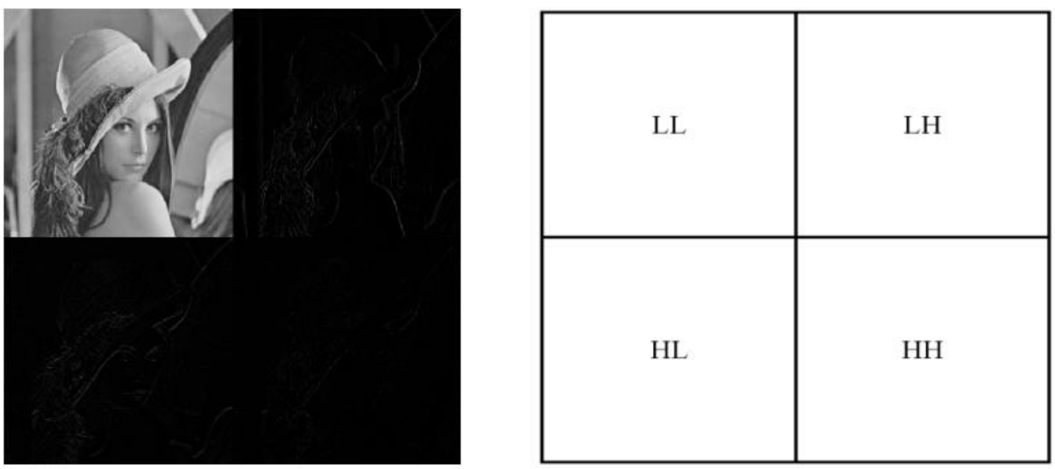
1	1	0	0
1	1	0	0
1	0	0	0
1	1	0	0

(b) Bitmap

186	186	162	162
186	186	162	162
186	162	162	162
186	186	162	162

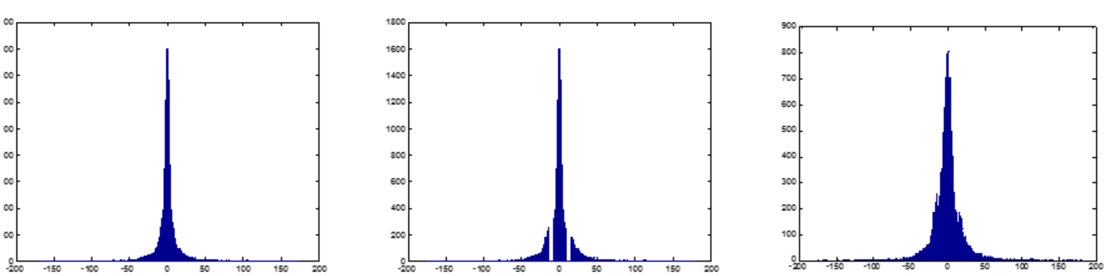
(c) Compressed

Figure 1: An example of AMBTC compression



(a) Image 'Lena' after one level integer DWT (b) The structure of the sub-bands of DWT

Figure 2: Structure of the image Lena after one level integer DWT



(a) Original (b) Shifted (c) Embedded

Figure 3: An example of histogram modification and data hiding process

5) Select the LH, HL, HH sub-bands of I_h and I_l after integer DWT to embed data with the method proposed in Section 2.2.1;

6) Impose inverse integer DWT on the corresponding sub-bands to get the ${I_h}'$ and ${I_l}'$ that contains hidden data;

7) Scan ${I_h}'$ and ${I_l}'$ to reconstruct the coefficients vectors H' and L', and then (H', L', BM) is the BTC compressed image with hidden data. In fact, the LL sub-bands after integer DWT can also be utilized for the data hiding, which may increase the embedding capacity. In the receiving end, with the encoded (H', L', BM), the compressed image with hidden data is decoded. Besides, the hidden data is extracted, and the original BTC compressed image is reversibly recovered.

The data extraction process is the inverse process of data hiding. Detailed steps for extracting the hidden data and reversible recovery of the original BTC compressed image are presented as follows.

1) Construct the sampled image with hidden data ${I_h}'$ and ${I_l}'$ by scanning the coefficients vectors H' and L';

2) Impose the integer DWT on the ${I_h}'$ and ${I_l}'$ to get the four sub-bands LL, LH, HL, HH;

3) Extract the hidden data from the four sub-bands of ${I_h}'$ and ${I_l}'$, and then recover LL, LH, HL, HH with method proposed in Section 2.2.2;

4) Impose the inverse integer DWT on the four recovered sub-bands to get the recovered I_h and I_l;

5) Reconstruct the quantized coefficients vectors H and L, and then the (H, L, BM) is recovered.

The additional data is hidden into the quantized vectors through histogram modification operated on the integer DWT domain. The bit map remains unchanged throughout the data hiding and extraction processes. In fact, the bitmap can also be incorporated for the data hiding, which will increase the embedding capacity further. Besides, the arrangements between the high mean values and low mean values can be utilized to present some data, which will also increase the embedding capacity.

4 Experiments

The proposed scheme has been imposed on different images to testify the validity. The standard images selected from the USC-SIPI image database are adopted for the demonstration. Random bit streams are embedded into these images as the hidden data. All the experiments are performed on the MATLAB 2012a running on a personal computer with CPU of AMD Phenom (tm) X4 810 Processor 2.6GHz, memory of 4 GB, and the operating system of Windows 7 x64 Ultimate Edition.

The original image Peppers, the image after BTC compression and the BTC compressed image with hidden data are presented in Figure 5 respectively.

The blocking strategy with different sizes in the BTC compression produces different quantized coefficients vectors. Therefore, different sampled images with different sizes are constructed, which affect the embedding capacity. The distortion caused to the BTC compressed images can be measured by the peak signal-to-noise ratio (PSNR), which is calculated as follows:

$$PSNR = 10 \times \log_{10} \frac{255^2}{MSE}(dB), \tag{5}$$

where

$$MSE = \frac{1}{N_1 \times N_2} \sum_{1}^{N_1} \sum_{1}^{N_2} (I_{i,j} - {I'}_{i,j}). \tag{6}$$

Different images with different BTC compression parameters and different embedding strength parameters are tested. Detailed data is presented in the following figures. The embedding capacity and PSNR of images that are compressed by BTC with block size 2×2 are presented in Figure 6. The horizontal axes represents different embedding strength parameters, while the vertical axes represents the embedding capacity and PSNR respectively in plot (a) and plot (b). The embedding capacity and PSNR of images that are compressed by BTC with block size 4×4 are presented in Figure 7. The horizontal axes represents different embedding strength parameters, while the vertical axes represents the embedding capacity and PSNR respectively in plot (a) and plot (b). The embedding capacity and PSNR of images that are compressed by BTC with block size 8×8 are presented in Figure 8. The horizontal axes represents different embedding strength parameters, while the vertical axes represents the embedding capacity and PSNR respectively in plot (a) and plot (b).

It is clear that, the embedding capacity increases when the embedding strength increases. However, the PSNR decreases at the same time. The embedding capacity decreases when the block size in the BTC compression increases. That is because the constructed images become smaller when the block size become bigger. Moreover, there will be less pixels left for hiding additional data. Embedding data into the constructed images will enlarge the distortion caused to the compressed images in some degree. Larger blocks are easier to be affected because more pixels will be changed when the same amount of pixels are modified in the constructed image.

Comparisons with existing schemes are presented in Table 1. The embedding strength parameters are $q = 8$ and $q = 2$ respectively in the tow comparisons. The block size of BTC compression is 4×4.

As can be seen in the Table 1, better performances are got with the proposed scheme. In fact, more data can be

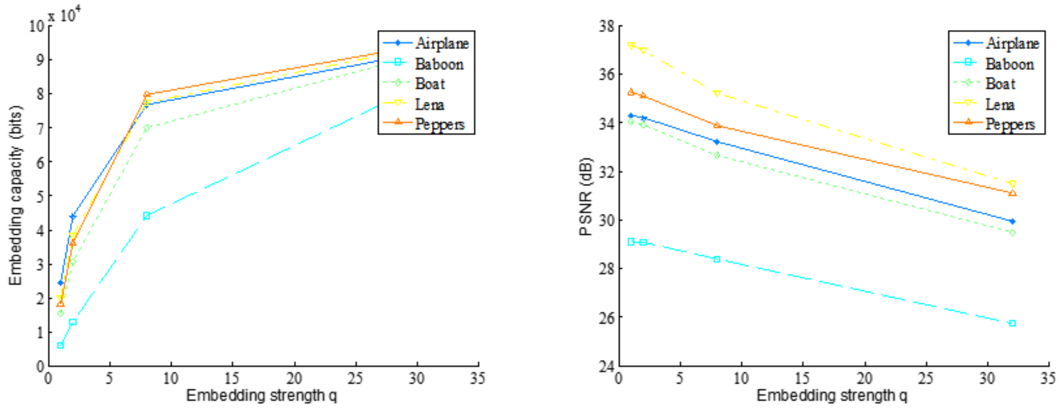
(a) Original (b) Constructed with high mean values (c) Constructed with low mean values

Figure 4: Lena image and it constructed sampled images

(a) Original (b) Compressed

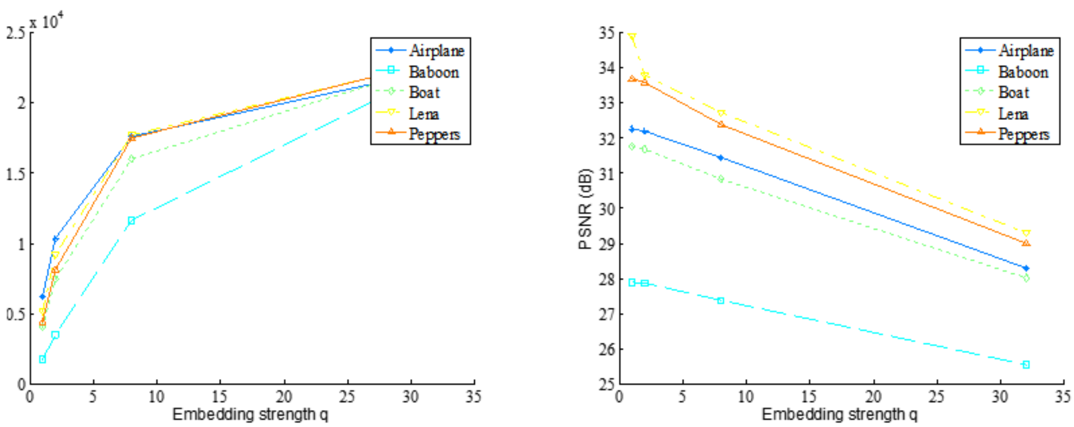
(c) Embedded (d) Recovered

Figure 5: Lena image and it constructed sampled images



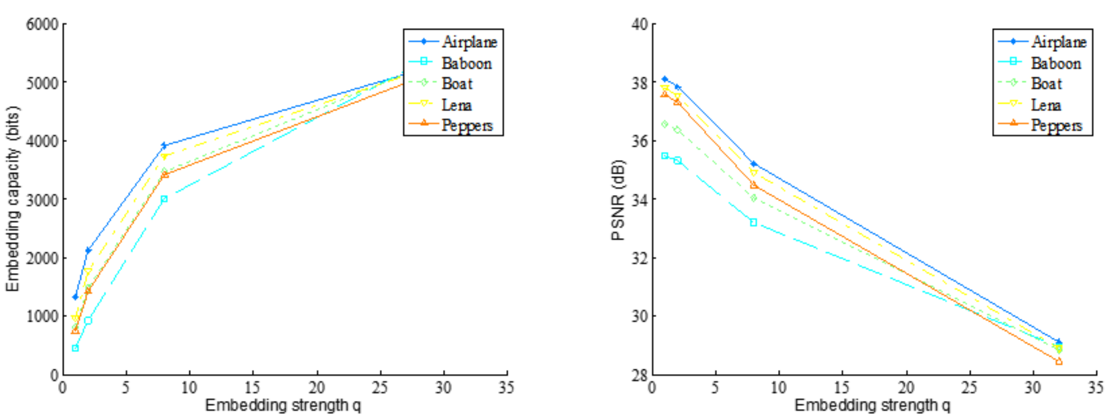
(a) Embedding capacity (b) PSNR

Figure 6: Embedding results with BTC block size 2×2



(a) Embedding capacity (b) PSNR

Figure 7: Embedding results with BTC block size 4×4



(a) Embedding capacity (b) PSNR

Figure 8: Embedding results with BTC block size 8×8

Table 1: Comparison with existing schemes

Image	Proposed with $q=8$		Li et al. [12]		Proposed with $q=2$		Lo et al. [16]	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Airplane	17600	31.4257	17375	31.033	10349	32.1805	7073	32.914
Baboon	11626	27.3667	-	-	3458	27.8537	-	-
Boat	16008	30.8236	17061	31.151	7418	31.6798	5387	31.751
Lena	17705	32.7076	16684	32.041	9174	33.7687	4025	33.364
Peppers	17487	32.3810	21969	31.595	8093	33.5590	4882	33.873

embedded into the BTC compressed images with higher embedding strength. The PSNR will decrease along with the embedding capacity increase, of course. Besides, the embedding strength can be decided adaptively according to the features of the cover media and the need of the actual applications. Different block sizes can be adopted in the process of BTC compression. The proposed scheme can achieve higher embedding capacity and cause lower distortion when a smaller block size is selected, which can be seen from Figure 6, Figure 7 and Figure 8. For example, if the block size in the BTC compression is 2×2, better results are got in Table 2. The embedding strength $q = 8$ in the experiments.

The reversible data hiding scheme adopted in the proposed scheme has larger embedding capacity compared with similar histogram modification based reversible hiding schemes. Besides, the histogram modification schemes imposed on the constructed images has better performance itself, compared with those data hiding schemes based on the bitmap, or on the pattern of the low mean vectors and high mean vectors in the BTC compressed images. The proposed scheme will never change the sizes of the BTC compressed images. Therefore, it keeps the compression rate of the original BTC compressed image, which will never reveal the existence of additional data.

5 Conclusion

A reversible data hiding scheme for BTC compressed images is proposed in this paper. Based on the high mean values and low values in the BTC compression, we constructed two sampled images for the data hiding process. A histogram modification based scheme in the integer DWT domain is utilized to achieve the high embedding capacity and low distortion. Through the proposed construction method of sampled images, traditional reversible data hiding schemes can be adjusted to realize the data hiding on the BTC compressed images. Besides, some other data hiding strategies were also mentioned in the paper to further improve the performances of the scheme.

Acknowledgments

The work described in this paper was supported by the Key program of National Science Fund of Tianjin, China (Grant NO. 11JCZDJC16000). The authors gratefully acknowledge the anonymous reviewers for their valuable comments.

References

[1] A. M. Alattar, "Reversible watermark using the difference expansion of a generalized integer transform," *IEEE Transactions on Image Processing*, vol. 13, no. 8, pp. 1147–1156, 2004.

[2] C. C. Chang, C. Y. Lin, and Yi H. Fan, "Lossless data hiding for color images based on block truncation coding," *Pattern Recognition*, vol. 41, no. 7, pp. 2347–2357, 2008.

[3] C. C. Chang, T. S. Nguyen, and C. C. Lin, "Reversible image hiding for high image quality based on histogram shifting and local complexity," *International Journal of Network Security*, vol. 16, no. 3, pp. 208–220, 2014.

[4] Z. F. Chen, Yu X. Su, and Z. M. Lu, "Reversible data hiding for btc-compressed color images using difference expansion," *ICIC Express Letters, Part B: Applications*, vol. 2, no. 5, pp. 1213–1218, 2011.

[5] Y. C. Chou and H. H. Chang, "A high payload data hiding scheme for color image based on btc compression technique," in *Proceedings of 4th IEEE International Conference on Genetic and Evolutionary Computing (ICGEC'10)*, pp. 626–629, 2010.

[6] E. J. Delp and O. R. Mitchell, "Image compression using block truncation coding," *IEEE Transactions on Communications*, vol. 27, no. 9, pp. 1335–1342, 1979.

Table 2: Optimized comparison with existing schemes

Image	Li et al. [12]		Proposed		Lo et al. [16]	
	Capacity	PSNR	Capacity	PSNR	Capacity	PSNR
Airplane	17375	31.033	76678	33.2088	7073	32.914
Baboon	-	-	44204	28.3893	-	-
Boat	17061	31.151	69998	32.6636	5387	31.751
Lena	16684	32.041	77337	35.2198	4025	33.364
Peppers	21969	31.595	79575	33.8742	4882	33.873

[7] J. Fridrich, M. Goljan, and R. Du, "Invertible authentication watermark for jpeg images," in *Proceedings of IEEE International Conference on Information Technology: Coding and Computing*, pp. 223–227, 2001.

[8] Li C. Huang, L. Yu Tseng, and M. S. Hwang, "The study on data hiding in medical images," *International Journal of Network Security*, vol. 14, no. 6, pp. 301–309, 2012.

[9] K. Su Kim, M. J. Lee, H. Y. Lee, and H. K. Lee, "Reversible data hiding exploiting spatial correlation between sub-sampled images," *Pattern Recognition*, vol. 42, no. 11, pp. 3083–3096, 2009.

[10] G. C. Langelaar and R. L. Lagendijk, "Optimal differential energy watermarking of dct encoded images and video," *Ieee Transactions on Image Processing*, vol. 10, no. 1, pp. 148–158, 2001.

[11] M. Lema and O. R. Mitchell, "Absolute moment block truncation coding and its application to color images," *IEEE Transactions on Communications*, vol. 32, no. 10, pp. 1148–1157, 1984.

[12] C. H. Li, Z. M. Lu, and Yu X. Su, "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.

[13] C. C. Lin and X. L. Liu, "A reversible data hiding scheme for block truncation compressions based on histogram modification," in *International Conference on Genetic and Evolutionary Computing*, pp. 157–160, 2012.

[14] C. C. Lin, W. L. Tai, and C. C. Chang, "Multilevel reversible data hiding based on histogram modification of difference images," *Pattern Recognition*, vol. 41, no. 12, pp. 3582–3591, 2008.

[15] Y. Linde, A. Buzo, and R. M. Gray, "An algorithm for vector quantizer design," *IEEE Transactions on Communications*, vol. 28, no. 1, pp. 84–95, 1980.

[16] C. C. Lo, Yu C. Hu, Wu L. Chen, and C. M. Wu, "Reversible data hiding scheme for btc-compressed images based on histogram shifting," *International Journal of Security and its Applications*, vol. 8, no. 2, pp. 301–314, 2014.

[17] Q. Mao, C. C. Chang, and T. F. Chung, "A reversible steganography suitable for embedding small amounts of data," *International Journal of Network Security*, vol. 16, no. 4, pp. 295–303, 2014.

[18] Z. Ni, Y. Q. Shi, N. Ansari, and W. Su, "Reversible data hiding," in *Proceedings of the 2003 IEEE International Symposium on Circuits and Systems (ISCAS'03)*, vol. 2, pp. II-912–II-915, 2003.

[19] W. L. Tai, C. M. Yeh, and C. C. Chang, "Reversible data hiding based on histogram modification of pixel differences," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 19, no. 6, pp. 906–910, 2009.

[20] J. Tian, "Reversible data embedding using a difference expansion," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, no. 8, pp. 890–896, 2003.

[21] P. Tsai, Yu C. Hu, and H. L. Yeh, "Reversible image hiding scheme using predictive coding and histogram shifting," *Signal Processing*, vol. 89, no. 6, pp. 1129–1143, 2009.

[22] C. H. Yang and M. H. Tsai, "Improving histogram-based reversible data hiding by interleaving predictions," *IET Image Processing*, vol. 4, no. 4, pp. 223–234, 2010.

[23] S. Zhang, T. Gao, and G. Sheng, "A joint encryption and reversible data hiding scheme based on integer-dwt and arnold map permutation," *Journal of Applied Mathematics*, vol. 2014, pp. 12, 2014.

[24] Z. Zhao, H. Luo, Z. M. Lu, and J. S. Pan, "Reversible data hiding based on multilevel histogram modification and sequential recovery," *International Journal of Electronics and Communications*, vol. 65, no. 10, pp. 814–826, 2011.

Shun Zhang born in 1986, is a Ph.D. candidate in in College of Software, Nankai University, China. His current research interests include watermarking, reversible data hiding, and multi-media security.

Tiegang Gao is a Prof. in College of Software, Nankai

University, China since 2006. His current research interests include information security, multimedia information processing and software engineering.

Liang Yang born in 1992, is a M.S. candidate in in College of Software, Nankai University, China. His current research interests include information security, reversible data hiding, and multimedia security.