Data Hiding based on Color Image Compression Technique

Pragati Pal
Dept. of Electronics and Telecom Engineering,
Mumbai University
Mumbai, India

Sukanya Kulkarni
Dept. of Electronics and Telecom Engineering,
Mumbai University
Mumbai, India

ABSTRACT
In recent years use of multimedia is increased astronomically for various purposes. Therefore the files which contains the color images need lots of storage space in computers hard disk. Thus it has become essential to reduce the size of color images by applying some compression techniques so that the transfer of color images over network becomes easier and it takes less storage space. In this theses we present a color block truncation coding along with data hiding. Block Truncation Coding is one of the lossy compression technique which is basically used to reduce the size of digital image. In this method the computational involved is very simple. The compressed file obtained by BTC is further used to hide the secret data by bit reversal method.

Keywords
btc; psnr; mean; standard deviation; bit plane; data hiding

1. INTRODUCTION
Nowadays, internet has become very popular for data transmission. In today's era of multimedia the amount of image data are increasing day by day. Hence large storage space and bandwidth are needed to store and transmit the images, which is quite costly. Therefore it has become essential to compress the image data. Image compression techniques can be classified as lossy image compression and loss less image compression. Loss less compression technique has less compression ratio but has high quality of digital images; it is mostly used in the medical application where the loss of image data is not tolerable. While lossy image compression technique has high compression ratio but with degraded quality of digital images. Image compression is used to save the cost of image storing and transmission.

In recent years most of the images are in color and color image contain lot of redundancy and require a large amount of storage space. The present work investigates color image compression using block truncation coding. Block Truncation Coding BTC is one of the lossy compression techniques that can significantly reduce the size of digital images with acceptable visual quality. The computation complexity involved in this method is very simple. This technique is used globally in many online as well as graphical applications.

With the rapid growth in telecommunication technology, more and more digital data are being distributed and transmitted over the internet. Thus it has become necessary to secure these exposed data over public channel. One of the most widely used techniques to secure the data is encryption and decryption. But these encrypted codes may attract the intruder to incur attacks. To avoid the suspect of illegal users other technique like steganographic method came in light to protect the data. This is another method of hiding the secret data in the cover medium in such a way so that only the intended user can detect and reveal the secret data.

Data hiding techniques can be classified into three different domain spatial domain, frequency domain and compression domain. In spatial domain secret data is usually embedded into the least significant bits LSBs of each pixel in the cover image \[1]. But the main drawback of this is that the stego image is easier to discover peculiar in the images by using some statistics analysis tool. In the frequency domain method \[2] the cover image is first transformed into frequency domain by using some of the transformation methods such as DWT, DCT etc. Secret data is then embedded into the selected coefficients. However this method has better visual quality but embedding capacity is less than the spatial domain method and also computation complexity is large and hence time consuming. In compression domain the cover image is first compressed by BTC and data is hidden into the compression code by changing the order of quantization level\ [3],[4].

This paper introduces a new approach for data hiding which is based upon compression code obtained by Block Truncation Code.

2. RELATED WORK
Conventional steganography usually embeds the secret data into the least significant bits LSBs of each pixel in the cover image \[6],[7]. Such an approach is called the LSBs substitution method, which only slightly changes the original values without obvious perceptible distortion. Although a number of variants of the LSBs method have been proposed \[1],[5],[8],[10]. such tiny distortions are unacceptable to some sensitive applications, such as military and medical data. Therefore, how to design a loss less hiding method becomes a crucial issue. The loss less steganographic method is a method by which the original cover image can be recovered completely from the stego image after extraction of secret data.

\[11] proposed a reversible data hiding method by compressing one of the LSB planes of a cover image in order to vacate space for secret data\[12] further improved the embedding capacity by quantizing the cover image and using arithmetic coding to compress the residual values. \[13] applied a circular interpretation of bi objective transforms and the patchwork algorithm to achieve reversible watermarking and reduce digital artifacts. \[14] proposed a difference expansion method that enlarged the difference between each pair of pixels to allow for hiding secrets. This approach was improved by [15] to enhance the embedding capacity, in which the difference expansion of vectors is used instead of pixel pairs. However the above method are built for raw image data rather than for compressed image formats that are usually used across the internet today. The reversible stegnography used in the compressed images not only prevents the quality of the compressed image from deteriorating again, it also can augment uses of the compression format.

There are many popular compression methods for digital images, such as discrete cosine transformation DCT\ [16] discrete wavelet transformation DWT \[17], vector quantization VQ [18], and block truncation coding BTC [19]. The first two methods are based on the frequency domain with great compression results, but they require many complicated computations. On the contrary, the last two methods compress images in the spatial domain using simple computations to obtain acceptable compression rates. The spatial domain technique is to modify the selected pixels in a cover image for implying secret data \[20].

In this theses we have proposed a reversible data hiding technique for color images that are compressed by BTC, which means the compressed BTC image is never degraded again while embedding secret data. Before compression, the image is divided into non-overlapping blocks. Each color image block that is compressed by BTC usually contains three bit maps and three pairs of quantization levels for R,G and B, respectively. The reversible hiding scheme is based on the order of these quantization levels.

Block Truncation Coding (BTC) is a well known compression scheme for the gray scale images \ [2][4].It is also known as moment preserving block truncation because it preserve the first and second moment of each image block. The original algorithm of BTC preserves the mean and standard deviation of each block. Mean and standard deviation are to be coded as a part of the block. The BTC algorithm involves the following steps:

- First the given image is divided into same size of non overlapping block.
- Then we calculate the mean and standard deviation value for each block as given by equation 1 and 2 respectively.

$$\overline{x} = \frac{1}{n}\sum_{i=1}^{n} Xi \qquad (1)$$

$$\sigma = \sqrt{\frac{(\overline{x}-x_i)^2}{n}} \qquad (2)$$

Where x_i represents the i^{th} pixel value of the image block and n is the total no of pixels in that block.

$$Low = \bar{x} - \sigma\left[\sqrt{\frac{p}{m-p}}\right] \qquad (3)$$

$$High = \bar{x} + \sigma\left[\sqrt{\frac{m-p}{p}}\right] \qquad (4)$$

where p represents the no of pixels in a block whose value are greater than or equal to the mean value of the image block and m is the total no of pixels in that block.

- Then binary bit plane is obtained by comparing each pixel value by the mean value of that block. If the current pixel value is greater than the mean value we assign it as 1 otherwise 0.
- At the decoder two quantizing level High and Low for each block is calculated as per equation 3 and 4 given below and replace all 1 by High and all 0's by Low value. The two quantizing levels High and Low and bit plane for each block are transmitted or stored for the compressed image. Thus by this process each block is reduced to a bit plane. For example, a block of 4x4 pixels will give a 32 bits compressed data, amounting to 2 bits per pixel (bpp).

3. PROPOSED METHOD

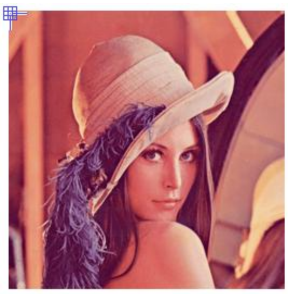
3.1 Color Block Truncation

In multimedia application most of the images are in color and color image contain lot of redundancy and require a large amount of storage space. Block Truncation Coding is basically applicable for gray images. Here we have enhanced it for color images. The color BTC applies the traditional BTC three times on the prime color plane i.e. red, green and blue color separately. Therefore for every compressed image requires three bit maps with BTC value.

3.2 CBTC Algorithms

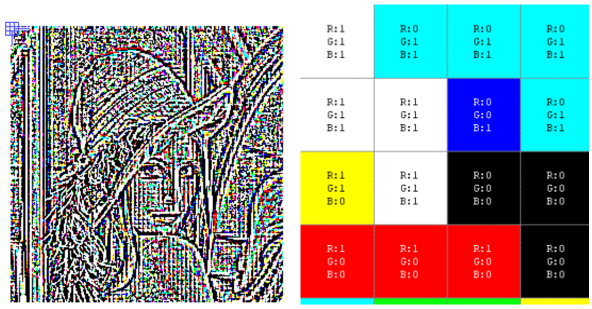
- First color image is divided into same size of non overlapping blocks.
- Then calculate the mean value and standard deviation of the entire three color plane for each block by using equation 1 and equation 2.
- Then each pixel value is compared with the mean value of the corresponding color plane.
- If the current pixel value is greater than the mean value, then we assign it as 1, otherwise 0.
- In this way we get three bitmap for each block.
- Then High and Low value is calculated for the entire three color plane for every block by using equation 3 and 4 respectively.
- After three bit maps have been made, then output the bit maps with the High and Low mean values to be the color BTC compression code.
- At the decoder end each 1 is replaced by High value and each 0 is replaced by Low value of the corresponding color plane.

(a)



R:222 G:134 B:124	R:221 G:135 B:124	R:217 G:132 B:121	R:218 G:134 B:121
R:223 G:133 B:121	R:224 G:136 B:123	R:217 G:131 B:118	R:220 G:133 B:119
R:224 G:132 B:117	R:224 G:133 B:118	R:219 G:130 B:114	R:221 G:131 B:114
R:224 G:130 B:114	R:222 G:129 B:111	R:223 G:130 B:112	R:221 G:128 B:110

(b)



(c)

Fig. 1. Test image lena, (a) original, (b) encoded & (c) decoded with their corresponding bitmaps for block size 4x4

As mentioned above, a true color image can be significantly reduced in size by using the color BTC technique.

3.3 Data Embedding Rule

Any data hiding technique tries to embed as many secret data as possible in the cover. The rules for embedding the data are as follow:

- First we check whether the block is embeddable or not. The block is embeddable if the value of equation 3 and equation 4 are not same.
- If the block is embeddable then in every block we can embed 3 secret bit by changing the order of the quantization level.
- For embedding 0 we do not change the order and keep the order as BM‖H‖L, Where BM is the bit map of corresponding plane and H and L are the corresponding value of high and low quantization level and ‖ is concatenate.
- For embedding 1 secret bit we change the order of compression code as BM‖L‖H, where BM, H and L has the same values as above.

Thus for hiding 000 and 111, the corresponding order of the compression code would be as follow.

$BM_r \| H_r \| L_r \| BM_g \| H_g \| L_g \| BM_b \| H_b \| L_b$

$BM_r \| L_r \| H_r \| BM_g \| L_g \| H_g \| BM_b \| L_b \| H_b$

Where BM_r, BM_g and BM_b are the bit maps of red green and blue color plane respectively and H_r, L_r , H_g, L_b and H_b, L_b are the high and low value of quantization level respectively and ‖ represents the concatenation operation.

3.4 Data Extracting Rule

In this stage the data are extracted and the original image is recovered back as follow.

- First we check the Low and High value of each color plane.
- If Low value > High value, we decode 1 as secret bit otherwise 0.
- If Low value = High value then it means nothing is hidden in that block.

4. IMAGE QUALITY MEASUREMENT

Image quality measures play important roles in various image processing application. Once Image compression system has been designed and implemented, it is important to be able to evaluate its performance. Peak Signal to Noise Ratio (PSNR) is a commonly used measurement to evaluate the visual quality of compressed image. PSNR is defined as follows:

$$PSNR = 10 \times \log 10 \frac{255^2}{MSE} \quad (5)$$

$$MSE = \frac{MSE_R + MSE_G + MSE_B}{3} \quad (6)$$

$$MSE_R = \frac{1}{w \times h} \sum_{i=0}^{h-1} \sum_{j=0}^{w-1} (I_R(i,j) - (I'_R(i,j))^2 \quad (7)$$

Where MSE represents the mean value of mean square errors (i.e. MSE_r, MSE_g, and MSE_b) of red, green and blue plane. $I_R(i,j)$ and $I_R'(i,j)$ represent the red value of the original image and the decoded image located at the pixel (i,j), respectively, and h and w are height and width of the block respectively. A large value of PSNR means that the reconstructed image after compression is similar to the original image. Contrarily, a small PSNR value means that the reconstructed image looks different from the original image.

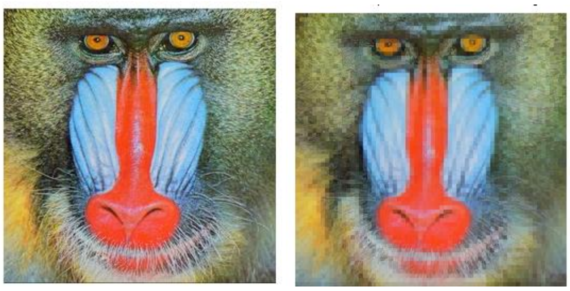
Table 1: The psnr values obtained with the BTC for test images of size 256x256 & 512x 512 for different block sizes

Image	psnr for size 256 x 256			psnr for size 512 x 512		
Blksize	2x2	4x4	8x8	2x2	4x4	8x8
Lena	31.6	27.2	24.0	33.8	29.2	26.0
Baboon	27.3	23.9	21.7	33.9	27.2	24.2
Tiffany	32.6	29.3	26.9	39.9	32.5	29.4
Zelda	34.2	29.2	25.9	39.8	33.3	29.0
Peppers	30.0	25.6	22.2	36.3	29.4	25.4
Airplane	28.6	24.6	22.3	34.9	28.1	24.5
Sailboat	27.8	23.5	20.9	33.6	27.1	23.3
Goldhill	28.2	24.6	22.5	34.9	28.1	24.9

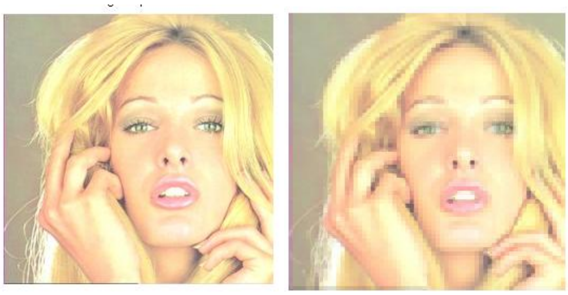
Experiments are carried out with the standard images Lena, Baboon, Tiffany, Zelda, Peppers, Airplane, Sailboat and Goldhill of size 256x256 and 512x512 pixels. The PSNR value is taken as a measure of reconstructed image quality. It is an attractive measure for the loss of image quality due to its simplicity and mathematical convenience. Peak signal to noise ratio is a qualitative measure based on the mean square error of the reconstructed image. If the reconstructed image is close to the original image, then MSE is small and PSNR takes a large value. PSNR is dimensionless and is expressed in decibels.

(a)

(b)



(c)



(d)

(e)

(f)

(g)

(h)

(i)

(j)

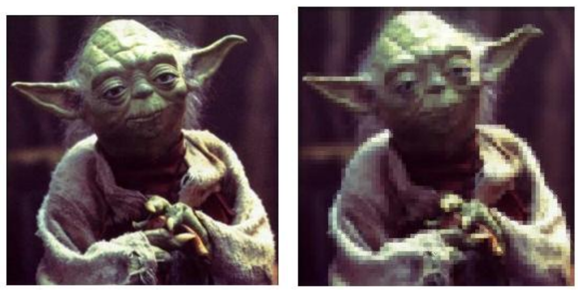


Fig. 2. Test images lena, baboon, tiffany, Zelda, peppers, airplane, sailboat, goldhill, boy,and yodal.(left original & right BTC compressed)

Table 2: Results of data hiding for different test images of pixel size 256x256 pixels for different block size.

Image	Capacity Bits 2x2 blk	Capacity Bits 4x4 blk	Capacity Bits 8x8 blk
Lena	41080	12288	3072
Baboon	46456	12288	3072
Sailboat	41592	12288	3064
Boy	38080	12112	3056
Yodal	40504	12272	3072
Airplane	38688	11840	2992

Table 3: Comparison with other methods

Image	Method	PSNR dB	Hiding Capacity
Lena	Keissaerian	31.28	26.76 Kbytes
	Sun & Yang	33.23	29.67 Kbytes
	Present Method	31.37 (BTC) 34.52(Filtered)	41.08 Kbytes
Peppers	Keissaerian	31.34	28.16 Kbytes
	Sun & Yang	33.20	32.66 Kbytes
	Present Method	30.16 (BTC) 34.69(Filtered)	38.68 Kbytes

5. RESULTS AND DISCUSSION

We have evaluated the performance of the proposed coding scheme through a computer simulation on different test images as shown in fig. These images are 8 bits per pixel and 256x256 pixels in size and 512x512 pixels in size. In order to enhance the quality of Image, decoded image is filtered through unsharp filter. This step further improves the PSNR as shown in table 2. The code is also tested with encryption and decryption algorithm. In order to make the hidden data more secure first the secret code is encrypted with RSA algorithm and the cipher text obtained is hidden as the secret code under the cover image. At the decoder first cipher text is obtained and then secret code is obtained after decryption. The simulation platform is Microsoft windows 7, core 2 duo and the proposed scheme is implemented using Mat lab. Two performance matrics are used to measure the performance of the proposed compression and hiding schemes: Hiding capacity (CAP) and the Image quality (PSNR).

6. CONCLUSION

In this paper color image compression using Block Truncation Coding followed by data hiding has been investigated. This technique is applied to different test images of 512x512 and 256x256 sizes respectively for block size 2x2, 4x4 & 8x8 with 8 bit per pixel for each color plane (i.e. 24 bits per pixel). The reconstructed image has a bit rate of 6 bits per pixel. Objective measures were used to evaluate the image quality such as PSNR and bit rate. It is observed as the block size increases, the image quality decreases. Maximum PSNR value is obtained for block size 2x2. Table 1 shows the compression ability of Color BTC. Table 2 shows the data hiding results for different test images. For 2x2 block hiding capacity is more. Table 3 shows the comparision of proposed method with other methods. From table 3, we observe that the proposed method outperforms in visual quality as well in the hiding capacity.

7. REFERENCES

[1] C. K. Chan and L. M. Cheng, `Hiding Data in Images by Simple LSB Substitution,'Pattern Recognition, Vol. 37, No. 3,Mar. 2004, pp. 469-474.

[2] C. C. Chang, T. S. Chen, and L. Z. Chung, `A Steganographic Method based uponJPEG and Quantization Table Modi_cation,' Information Sciences, Vol. 141,No.1-2,Mar.2002,pp 123-138

[3] C. C. Chang,C. Y. Lin, and Y. H. Fan, `Lossless Data Hiding for color images Based on Block Truncation Coding,' Pattern Recognition vol. 41, no. 7, pp. 2347-2357, July 2008.

[4] Y. C. Chou and H. H. Chang,`A Data Hiding Scheme for Color Image Using BTC Compression Technique,' IEEE Int. Conf. on Cognitive Informatics 2010.

[5] W. Bender, D. Gruhl, N. Morimoto, and a. Lu., `Techniques for data hiding,' IBM Syst. J. 35 (34), no. 3, pp. 313-326, 1996.

[6] R.G.V. Schyndel,A.Z.Tirkel,C.F.Osborne, `A digital watermark' Proceedings of theIEEE International Conference on Image Processing (ICICP-94),Austin Texas, 1986, pp.86-90.

[7] C. C. Chang, J. S. Hasiao and C.S. Chan, `Finding optimal LSB substitution in image hiding by dynamic programming strategy,' Pattern Recognition 36(7), 2003,pp 1583-1595.

[8]] Y. H. Yu, C. C. Chang,and Y. C. Hu, `Hiding secret data in images via predictive coding,' Pattern Recognition 38(5) pp. 691-705, 2005.

[9] C. K. Chan, L.M.Cheng, `Hiding data in images by simple LSB substitution ,' Pattern Recognition34(3) 2001, 671-683.

[10] R.Z.Wang,C.F.Lin,J.C.Lin,`Image hiding by optimal LSB substitution and genetic algorithm,' Pattern Recognition(34)3 2001,671-683.

[11] J. Fridrich,M. Goljan and R.Du, `Lossless data embedding-new paradigm in digital watermarking,' Emerging Appl. Multimedia Data Hiding 2002(2) pp. 185-196, 2002.

[12] M. U. Celik, G. Sharma, A. M. Tekalp, and E. Saber, `Reversible Data Hiding' Proceedings of the IEEE International Conference on Image Processing Rochester, NY, 2002, pp.157-160.

[13] C. De. Vleeschouwer, J. F. Delaigle, B. Macq., `Circular interpretation of bijective transformation in lossless watermarking for media asset management,' IEEE Trans. Multimedia 5 (1), 2003,pp 97-105.

[14]] J. Tian, `Reversible data embedding using a di_erence expansion,' IEEE Trans. Circuits Syst. Video Technol. 13(8) pp. 890-896, 2003.

[15] A. M. Allatar, `Reversible watermark using the di_erence expansion of a generalized integer transform ,' IEEE Trans. Image process. 13(8) 2004, 1147-1156.

[16] C.C.Chang, T.S.Chen, L.Z.Chung, `A steganographic method based upon JPEG and quantization table modi_cation,' Inf. Sci. 141 (1) pp. 123-138, 2002.

[17] I.J. Cox, J. Kilian, F.T.Leighton, T.Shamoon, `Secure spread spectrum watermarking for multimedia' IEEE Trans. Image Process.6(12), 1997, pp. 1673-1687.

[18] R.M.Gray, `Vector quantization,' IEEE Trans. Commun.27(9), 1979,pp 1335-1341.

[19] E.J.Delp,O.R. Mitchell,`Image Compression using block truncation coding,' IEEE Trans. Acoust. Speech Signal Process. 1(2), 1984,pp 4-29.

[20] C.C.Chang,Y.C.Chou, and T.D. Kieu, `High capacity data hiding for grayscale Images,' Fundamenta Informaticae, Vol. 86, No. 4, Jul.2008, pp. 435-446.