# A Modified BTC Using Quincunx Subsampling and Pattern Fitting for Very Low bpp

Bibhas Chandra Dhara
Department of Information Technology
Jadavpur University, Salt Lake Campus
Kolkata 700098, India
bibhas@it.jusl.ac.in

Bhabatosh Chanda
Electronics and Communication Sciences Unit
Indian Statistical Institute
Kolkata 700108, India
chanda@isical.ac.in

## Abstract

*In conventional block truncation coding images are divided into blocks of size $4 \times 4$ and then each block is encoded by two gray levels and a bit-pattern. Here instead of $4 \times 4$ blocks, image is divided into $8 \times 8$ blocks. From each block two $4 \times 4$ sub-blocks are generated by quincunx sampling method. In this work a modified scheme of BTC for the $4 \times 4$ block have been proposed where the computed representative gray levels are the bias and the contrast in each block. Secondly, instead of determining bit-pattern for each block, an optimum bit-pattern is selected from a pattern-book.Thirdly, if the contrast is low the block is assumed to be smooth and bit-pattern is not required to reconstruct the block. Again to reduce the bit rate the contrast component and the predictive residual of the bias component are entropy coded to achieve lower bit rate. Further to reduce the bit rate, preserving same quality, indices of best fit patterns are coded using two different index graph. Finally, $8 \times 8$ block is reconstructed using interpolation technique.*

**Key words:** *Image compression, Block Truncation coding, Vector quantization, index graph, Pattern fitting.*

## 1. Introduction

Image compression and image sequence coding has huge applications in video conferencing, video phones, TV transmission etc. Since 1970's the Block Truncation Coding (BTC) [2, 7, 4] has been studied a lot. This is a lossy but an attractive image coding algorithm for its simplicity, low computational cost and relatively high image quality. In this method sharp gray level transition and textured areas are reconstructed well; where as smooth gray level transitions are less well preserved. Like other image compression methods, quality of the reconstructed image is measured in terms of peak-signal-to-noise ratio (PSNR) and degree of compression by bits-per-pixel (bpp). The BTC output data set includes a bit-pattern, which defines the quantization bin of each pixel, and two reconstruction levels. The levels are determined so that the mean and variance of the original image block and reconstructed block are same. To define the bit-pattern in BTC, block mean is used as threshold. In Absolute moment block truncation coding (AMBTC) [7] has a simple algorithm structure as absolute moment is used to define the bit-pattern. In conventional BTC the main draw back is high bit rate (2 bpp). Several modification has been proposed to reduce bit rate, such as, using median filter [1], adaptive coding [10], DCT-BTC [16], BTC with decimation and interpolation [17], etc.

The reconstruction levels generated by BTC are usually expressed by 8+8 bits, they can also be stored by 10 bits using joint quantization [5, 9]. In order to reduce the bit rate Vector Quantization (VQ) [8, 11] technique has also been applied to quantize the vector formed by the two gray levels corresponding to the reconstruction levels generated by BTC for each block [13]. To further reduce bit rate, the generated bit pattern approximated by one of a set of preselected bit-pattern [12, 3], where the index of the approximating bit-pattern is used to code the block.

In this article a modification of BTC is given to reduce bit rate significantly. Here images are partitioned into $8 \times 8$ blocks ($B8$) instead of $4 \times 4$ block ($B4$) used in BTC. From $8 \times 8$ block two $4 \times 4$ sub-blocks $B4_1$ and $B4_2$ are generated according to quincunx subsampling method. The sampling method is shown in Figure 1. Then each $B4$ blocks are coded by proposed method and finally $B8$ is reconstructed by interpolation technique. In coding purpose, to define the partition of block, a set of bit-pattern used. The bit-pattern from pattern book, which fits the block best is used as the bit-pattern of the block in BTC. Thus, the index of the selected pattern is sufficient to reconstruct the block. Based on the best fit pattern two quantities, namely bias and contrast, are computed and that are used in the reconstruction of the block. The paper is organised as follows. Section 2 contains basic concept of BTC and proposed method. How to define the partition of a block is given in section 3. Coding of a block is studied in section 4. Test results are given in section 5. Finally, conclusion is given in section 6.

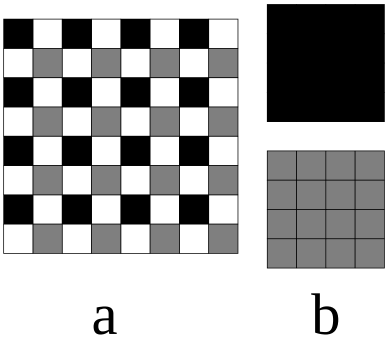



Figure 1: Two $4 \times 4$ sub-block from $8 \times 8$ block by quincunx sampling method. (a) a block of size $8 \times 8$, (b) two sub-block of size $4 \times 4$

## 2. Conventional BTC

In original BTC method [2] images are partitioned into $n \times n$ ($n$=4) sub-blocks. Then two level quantization is performed for the pixels of the block so that block mean and variance of the reconstructed block are identical to those of the original block. The output of the BTC method is two gray levels and the quantization information (ie., partition of the block into two subsets which can be represented by binary pattern). Suppose an image block $C$ contain k (=$n^2$) number of pixels. Let $f(\mathbf{x}_i)$, $\mathbf{x}_i \in C$ are the original intensity of the pixels where $C$ represents the set of coordinates of pixels in the block, i.e., $C = \{\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_k\}$. The first two block moments $m_1$ and $m_2$ are given by

$$m_1 = \frac{1}{k}\sum_{i=1}^{k} f(\mathbf{x}_i) \quad \text{and} \quad m_2 = \frac{1}{k}\sum_{i=1}^{k} f^2(\mathbf{x}_i)$$

where sample variance $\sigma^2$ of image block is $\sigma^2 = m_2 - m_1^2$.

Suppose, based on the pixel intensities, the block $C$ is partitioned into two sets of pixels $C_0$ and $C_1$ such that $C = C_0 \cup C_1$ and $C_0 \cap C_1 = \emptyset$ where $C_0 = \{\mathbf{x}'_1, \mathbf{x}'_2, \ldots, \mathbf{x}'_{k'}\}$ and $C_1 = \{\mathbf{x}''_1, \mathbf{x}''_2, \ldots, \mathbf{x}''_{k-k'}\}$. This partition is represented by assigning one of two labels, say 0 and 1, to the pixels. Without loss of generality, let the pixels of set the $C_0$ are marked by 0 and that of $C_1$ by 1. Thus the partition can be represented as a bit-pattern as stated above. In original BTC the partition is defined as

$$C_0 = \{\mathbf{x}_i | f(\mathbf{x}_i) \leq m_1\}$$
$$C_1 = \{\mathbf{x}_i | f(\mathbf{x}_i) > m_1\}.$$

During reconstruction, the pixels marked by 0 will be given the value $a$ and that marked by 1 will be given the value $b$. The values $a$ and $b$ satisfy

$$km_1 = k'a + (k - k')b$$
$$km_2 = k'a^2 + (k - k')b^2$$

Solving for $a$ and $b$ we get

$$a = m_1 - \sigma\sqrt{\frac{k - k'}{k'}} \tag{1}$$

$$b = m_1 + \sigma\sqrt{\frac{k'}{k - k'}} \tag{2}$$

The intensity $\hat{f}(\mathbf{x}_i)$ of the pixels of the corresponding block of the reconstructed image is given by

$$\hat{f}(\mathbf{x}_i) = \begin{cases} b & \text{if } \mathbf{x}_i \in C_1 \\ a & \text{if } \mathbf{x}_i \in C_0 \end{cases}$$

The output of conventional BTC (CBTC) on 'LENA' ($512 \times 512$) shown in Figure 4 (b) with PSNR = 32.89 and bpp = 2. In this proposed method during reconstruction, gray levels $A-d$ and $A+d$ are used in place of $a$ and $b$. The values $A$ and $d$ represents the bias (low frequency component) and contrast (high frequency component) respectively within the block. Simplifying according to above we get,

$$A = m_1 + \frac{\sigma(2k' - k)}{2\sqrt{k'(k - k')}} \tag{3}$$

$$d = \frac{\sigma k}{2\sqrt{k'(k - k')}} \tag{4}$$

Now the reason why we used $(A, d)$ instead of $(a, b)$, that is explained later. In this proposed method, for each $B8$ two $B4$ sub-block $B4_1$ and $B4_2$ generated through quincunx subsampling are considered for coding purpose. When each $B4$ is coded by CBTC and $B8$ is reconstructed by interpolation, output of this method (QS-CBTC) is shown in Figure 4(c) with PSNR = 30.36 and bpp = 1.

## 3. Partitioning of Blocks

In conventional BTC method, output for each block is a bit-pattern and two gray levels. In current method the gray values are defined by $A$ and $d$. How the partition of a block is defined?

To define the partition of block, a pattern-book containing $N$ two-level patterns of size $n \times n$ are used. To generate the pattern-book a collection of large number of images are considered. Each block of size $n \times n$ of those images is transformed so that average of the block becomes zero and variance of each block becomes same. Such transformation is needed to ensure that all the block having similar gray level or intensity pattern must belong to same cluster irrespective of their mean and variance. Then a clustering algorithm (e.g. K-means algorithm) is applied on these transformed block data-vectors to form $N$ clusters. Bit-pattern is generated by applying zero as threshold to each gray level pattern corresponding to cluster center. Once the pattern-book is formed we fit the candidate image block to each of

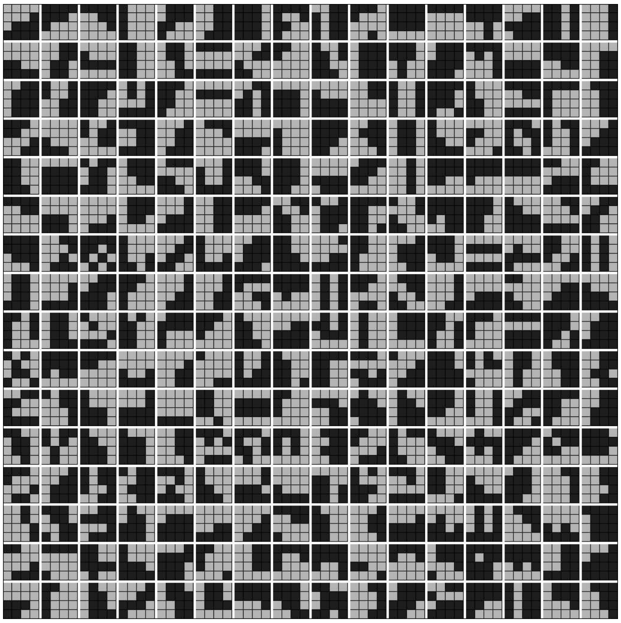


Figure 2: Set of bit-pattern used to partition $B4$ in proposed method.

these patterns $C_j, j = 0, 1, \ldots, N-1$, say, in mean-square-error sense. For example, if we try to fit the image block to the pattern $C_j$, mean-square-error in fitting is computed as

$$e_{j0} = \frac{1}{k'} \sum_{\mathbf{x}_i \in C_{j0}} (f(\mathbf{x}_i) - \mu_0)^2$$

$$e_{j1} = \frac{1}{k-k'} \sum_{\mathbf{x}_i \in C_{j1}} (f(\mathbf{x}_i) - \mu_1)^2$$

where $\mu_0 = \frac{1}{k'} \sum_{\mathbf{x}_i \in C_{j0}} f(\mathbf{x}_i) \quad and$
$\mu_1 = \frac{1}{k-k'} \sum_{\mathbf{x}_i \in C_{j1}} f(\mathbf{x}_i)$

Hence, total error in fit for the the $j$-th pattern is given by $e_j = e_{j0} + e_{j1}$. Finally, index $m$ of best fit pattern is obtained for $m = arg\min_j\{e_j\}$. To represent the partition of blocks, proposed BTC sends the index $m$ instead of the entire bit-pattern. That means only $\log_2 N$ bits is transmitted instead of $k$ bits. If former one is much less than the latter (which is usually taken), a significant reduction in bpp can be achieved. The 256 bit-patterns which are used in our experiment are shown in Figure 2. Result of BTC obtained from fitting patterns of Figure 2 to two $B4$ blocks and then reconstructing $B8$ block by interpolation (QS-PF) is shown in Figure 4(d) with PSNR = 29.55, bpp = 0.56.

## 4. Encoding the Block

For each $B4$, two quantization levels and one bit-pattern (the index of the best fit bit-pattern) ie., a triplet $(A, d, index)$ has to be encode.

| | | |
|---|---|---|
| $(NA)_2$ | $(NA)_3$ | $(NA)_4$ |
| $(NA)_1$ | $A_c$ | |

Figure 3: Blocks used in Prediction of $A'$ in current block.

In CBTC and also in AMBTC the quantization data to be stored in the compressed file is either the pair $(a, b)$ or $(a, b-a)$. They are usually expressed by 8+8 bits, but they can also be stored by 10 bits using joint quantization [5, 9]. In current method $A - d$ and $A + d$ are quantization levels used in reconstruction. Here we store $A$ and $d$. It is evident from equation (1), (2), (3) and (4) that $A = \frac{b+a}{2}$ and $d = \frac{b-a}{2}$. In view of the Wavelet theory [14], $A$ may be considered as the response of the scaling function or the low-resolution ($\frac{1}{4}$ times) representation of the block and $d$ is the response of wavelets whose natures are represented by the patterns shown in Figure 2. So the standard deviation of $d$ is smaller than that of $a$ and $b$, and this leads to higher compression for $d$ by entropy coding. Secondly, small value of $d$ means the gray levels are more or less uniform and block intensity can be represented by $A$ only. From this point of view we apply a small threshold, $d_{th}$, to obtain

$$d' = \begin{cases} 0 & \text{if } d \leq d_{th} \\ d - d_{th} & \text{otherwise} \end{cases} \tag{5}$$

$d_{th}$ is selected in such a way that the error introduced due to this approximation does not reduce the PSNR of that particular block below a fixed quantity, say, $Z$ dB (see equation (7)).

Thus, in general, we need to transmit $A$, $d'$ and index of the corresponding pattern in the said order. However, it may be noted that if $d'$ is zero, we need not transmit the pattern index. This leads to further reduction in bit rate. As $A$ is bias of the block, it has strong correlation with that of its neighboring blocks because of spatial homogeneity. So for further reduction in bpp, $A$ is coded by predictive coding. Figure 3 shows the blocks that are used for prediction of $A_c$. Suppose the values are $(NA)_1, (NA)_2, (NA)_3, (NA)_4$ and $A_c$, are known. Assuming a linear model, the prediction error is:

$$\triangle A = A_c - \sum_{i=1}^{4} w_i (NA)_i \tag{6}$$

weights $w_i$ are estimated considering a large number of images. It is evident that the variance of $\triangle A$ is much less than that of $A$. Hence, through entropy coding $\triangle A$ can be represented by less number of bits than that is needed to represent $A$ directly. This is a lossless coding.

Here, images are divided into $B8$. From $B8$ two $B4$ subblocks $B4_1$ and $B4_2$ are considered for coding purpose, so we need to transmit $(A_1, d_1, index_1)$ and $(A_2, d_2, index_2)$. With above approximation we transmit $d_1'$ and $d_2'$ respectively in place of $d_1$and $d_2$. It is highly expected that $A_1$ and $A_2$ are nearer, and $|A_1 - A_2|$ will be small value.

In coding purpose for the block $B4_1$ we use the predictive error $\triangle A_1$, and for $B4_2$ the value of $A_1 - A_2$ is used. Hence, in coding purpose for each $B8$ $(\triangle A_1, A_1 - A_2, d_1^{'}, d_2^{'}, index_1, index_2)$ is used, where first four components are coded by entropy coding.

It may be noted that here the blocks producing $A_i$'s are further away from each other compared to those of CBTC. So to improve correlation among $A_1$'s and that between $A_1$ and $A_2$ we apply a low-pass filter (neighborhood averaging) before subsampling. This also reduce deviation in $d_i$values. Then result of this modification (QS-PF-$d^{'}$) using with $d_{th}$ = 4 is shown in Figure 4(e) for which PSNR = 29.42 and bpp = 0.41 considering entropy coding for $\triangle A_1$, $|A_1 - A_2|$ and $d^{'}$-values with normal coding for indices. Only the coding scheme for indexing part is left. To encode the indices in normal way $log_2 N$ (=8) bits have to used for each index. We denote $ij^{th}$ $B8$ by $B8^{ij}$and its sub-blocks are denoted accordingly. Let $X_k^{ij}$, for k= 1,2 be the random variable for the best fit pattern index of the block $B4_k^{ij}$. Using the joint frequency distribution of $X_1^{ij} X_2^{ij}$ and $X_1^{ij} X_1^{ij+1}$ we define two index graph $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$. $V_1$ is same as $V_2$ ($|V_1| = |V_2|$ = 257) that represents the indices of bit-patterns in pattern book and one auxiliary index for those blocks for which $d'$ = 0. When $d'$=0 this auxiliary index not being actually transmitted. Degree of each vertices in both graph is taken $\leq$ 31.

Following is the coding scheme for indices. First find the best fit bit-pattern by linear search method on original pattern book.

Now consider three blocks $B4_1^{ij}$, $B4_2^{ij}$ and $B4_1^{ij+1}$ and corresponding indices are $x_1^{ij}$, $x_2^{ij}$ and $x_1^{ij+1}$ respectively. Assume that $x_1^{ij}$ already being coded. Now for the coding of $x_2^{ij}$ index graph $G_1$ is used and following is the logic being used for coding purpose.

1. if($d_2^{ij'} > 0$)
   (a) if($x_1^{ij}$ is adjacent of $x_2^{ij}$)
      Let $x_2^{ij}$ is $k^{th}$ ($\leq$ 31) adjacent of $x_1^{ij}$ then corresponding encoding string is binary representation of $k$, which takes 5 bits.
   (b) else (ie., $x_1^{ij}$ is non-adjacent of $x_2^{ij}$)
      Encoding string is 11111 followed by binary representation of $x_2^{ij}$, so 13 bits required.
2. else (ie., $d_2^{ij'}$ = 0 )
   (a) No need to transmit any code for index.

Similarly, for $x_1^{ij+1}$ index graph $G_2$ is used and following coding scheme being used for coding purpose.

1. if($d_1^{ij+1'} > 0$)
   (a) if($x_1^{ij}$ is adjacent of $x_1^{ij+1}$)
      Let $x_1^{ij+1}$ is $k^{th}$ ($\leq$ 31) adjacent of $x_1^{ij}$ then corresponding encoding string is binary representation of $k$, which takes 5 bits.
   (b) else (ie., $x_1^{ij}$ is non-adjacent of $x_1^{ij+1}$)
      Encoding string is 11111 followed by binary representation of $x_1^{ij+1}$, so 13 bits required.
2. else (ie., $d_1^{ij+1'}$ = 0)
   (a) No need to transmit any code for index.

This index graph concept preserve the quality like pattern fitting with $d^{'}$ (QS-PF-$d'$) in same time constraint, but this gives better result in bit rate. The result at final stage (QS-PF-$d'$ with index graph) of compression (QS-PF-$d^{'}$-IG) is shown Figure 4(f) with PSNR 29.42 and 0.38.

## 5. Experimental Results

To study the nature of $d^{'}$, $A_1 - A_2$ and $\triangle A_1$ several 8-bit images of size $512 \times 512$ are used. Here number of two-level patterns used to define the partition is 256 as shown in Figure 2. The original images used in this experiment are shown in Figure 5. The performance is evaluated by bit-per-pixel (bpp) and peak signal-to-noise ratio (PSNR) defined as

$$PSNR = 10 log_{10} \frac{255^2}{MSE} dB \qquad (7)$$

where $MSE$ (mean square error) for a reconstructed image $\hat{f}(\mathbf{x}_i)$ is defined as

$$MSE = \frac{1}{M} \sum_i [f(\mathbf{x}_i) - \hat{f}(\mathbf{x}_i)]^2$$

where $M$ is the total number of pixels in the image.

The results for comparisons are given in Table 2 and Table 3 respectively. In Table 1 the output of different steps of the proposed method is shown. The current method is a combination of BTC and patter fitting method. In all steps of Table 1, for encoding purpose entropy coding is used for $d_1^{'}$, $d_2^{'}$, $\triangle A$ and $A_1 - A_2$ to reduce bit rate. In step QS-PF and QS-PF-$d^{'}$ indices are encoded using 8 bits. In the last step indices are encoded using index graphs.

Results of the proposed method are compared with that of three modified BTC methods namely PF64 [3], VQ-CIBTC [6] and Adapt.D/I [15]. The figures are shown in Table 2 and Table 3 respectively. PF64 is a modified BTC using pattern fitting method and where images are partitioned into $4 \times 4$ block and size of pattern book is 64. In VQ-CIBTC method we have set *Thr1* to 10, *Thr2* to 50 as suggested in the original paper [6] and used 2 bits for identifying the class of a block. In Adapt.D/I method *Thr1* and *Thr2* have been set to 6 and 5 respectively to achieve the

PSNR close to what was reported in the original paper [15] and bit-rate has been measured with predictive entropy coding. The reconstructed images using proposed method are shown in Figure 6.

Table 1: Results of different steps of Proposed method.

| Steps | Average PSNR | Average bpp |
|---|---|---|
| CBTC | 33.89 | 2.00 |
| QS-CBTC | 30.98 | 1.00 |
| QS-PF | 30.37 | 0.56 |
| QS-PF-$d'$ | 30.06 | 0.43 |
| QS-PF-$d'$-IG | 30.06 | 0.40 |

## 6. Conclusion

In this paper we have presented a modified version of BTC to achieve high compression ratio sacrificing image quality a little. Such compression methods are useful for transmitting very large amount of data in real-time such as in video. The method considers $8 \times 8$ image block (unlike CBTC which considers $4 \times 4$ block) and generates two $4 \times 4$ by quincunx subsampling. Each $4 \times 4$ block is processed step-by-step to achieve as low bpp as possible while keeping PSNR as high as possible. Result of each step in terms of PSNR and bpp is presented in Table 1 so as to track the improvement in terms of our objective. Final results are compared with some other modified version of BTC and are found superior regarding the trade-off between PSNR and bpp.

Table 2: Performance comparison in terms of PSNR.

| Methods | Images | | | | Avg. |
|---|---|---|---|---|---|
| | Couple | Lena | Peppers | Zelda | |
| Proposed Method | 29.36 | 29.42 | 29.02 | 32.54 | 30.06 |
| PF64 | 31.65 | 31.59 | 31.69 | 34.89 | 32.45 |
| VQ-CIBTC | 30.61 | 30.92 | 30.76 | 33.57 | 31.46 |
| Adapt.D/I | 32.46 | 31.87 | 32.69 | 34.14 | 32.79 |

Table 3: Performance comparison in terms of bpp.

| Methods | Images | | | | Avg. |
|---|---|---|---|---|---|
| | Couple | Lena | Peppers | Zelda | |
| Proposed Method | 0.42 | 0.38 | 0.39 | 0.39 | 0.40 |
| PF64 | 0.75 | 0.64 | 0.68 | 0.60 | 0.67 |
| VQ-CIBTC | 0.82 | 0.78 | 0.79 | 0.76 | 0.79 |
| Adapt.D/I | 1.30 | 1.20 | 1.30 | 1.10 | 1.23 |

## References


[1] G. Arce and N. C. J. Gallagher. Btc image coding using median fi lter poots. *IEEE Transaction on Communication*, 31:784–793, 1983.

[2] E. J. Delp and O. R. Mitchell. Image compression using block truncation coding. *IEEE Transactions on Communications*, 27:1335 – 1342, 1979.

[3] B. C. Dhara and B. Chanda. Block truncation coding using pattern fi tting. *Pattern Recognition*, 37:2131 – 2139, 2004.

[4] P. Franti and O. Nevalainen. Block turncation coding with entropy coding. *IEEE Transaction on Communication*, 43:1677–1685, 1995.

[5] D. J. Healy and O. R. Mitchell. Digital video bandwidth compression using btc. *IEEE Transactions on Communications*, 29:1809 – 1817, 1981.

[6] C. H. Kuo, C. F. Chen, and W. Hsia. A comression algorithm based on classifi ed interpolative block truncation coding and vector quantization. *Journal of Information Science and Engineering*, 15:1–9, 1999.

[7] M. Lema and O. Mitchell. Absolute moment block truncation coding and applications to color images. *IEEE Transactions on Communications*, 32:1148 – 1157, 1984.

[8] Y. Linde, A. Buzo, and R. M. Gray. An algorithm for vector quantization. *IEEE Transactions on Communications*, 28:84 – 95, 1980.

[9] O. R. Mitchell and E. J. Delp. Multilevel graphics representation using block truncation coding. *Proceedings of the IEEE*, 68:868–873, 1980.

[10] P. Naciopoulos, R. Ward, and D. Morse. Adaptive compression coding. *IEEE Transactions on Communications*, 39:1245 – 1254, 1991.

[11] M. N. Nasrabadi and R. B. King. Image coding using vector quantization: a review. *IEEE Transactions on Communications*, 36:957 – 971, 1988.

[12] S. I. Olsen. Block truncation and planar image coding. *Pattern Recognition Letters*, 21:1141–1148, 2000.

[13] V. R. Udpikar and J. P. Raina. Btc image coding using vector quantization. *IEEE Transactions on Communications*, 35:352 – 355, 1987.

[14] C. Valens. A really friendly guide to wavelets. *Available at: http://perso.wanadoo.fr/*, 1999.

[15] Y. K. Wang and G. F. Tu. Block truncation coding with adaptive decimation and interpolation. In *Proc. SPIE International Conference on Visual Communication and Image Prrocessing 2000 (VCIP 2000)*, volume 4067, pages 430–437, Perth, Australia, 2000.

[16] Y. Wu and D. C. Coll. Btc-vq-dct hybrid coding of digital images. *IEEE Transactions on Communications*, 39:1283 – 1287, 1991.

[17] B. Zeng and Y. Neuvo. Interpolative btc image coding with vector quantization. *IEEE Transactions on Communications*, 41:1436 – 1438, 1993.


(a) (b)


(c) (d)


(e) (f)

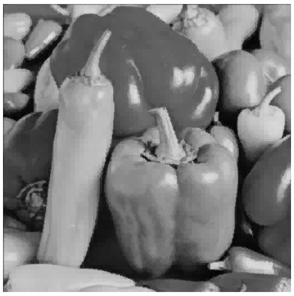
Figure 4: (a) Original Image, (b) - (f) Output of different steps. (b) CBTC (PSNR = 32.89, bpp = 2), (c) QS-CBTC (PSNR = 30.36, bpp = 1), (d) QS-PF (PSNR = 29.55, bpp = 0.56), (e) QS-PF with $d'$ (PSNR = 29.42, bpp = 0.41), (f) QS-PF-$d'$ using index graph (PSNR = 29.42, bpp = 0.38).


(a) Couple (b) Lena


(c) Peppers (d) Zelda

Figure 5: Original Images used in the experiment.


(a) Couple (b) Lena




(c) Peppers (d) Zelda

Figure 6: The Reconstructed images using Proposed Method.