Journal of Engineering and Applied Sciences 14 (8): 2546-2549, 2019
ISSN: 1816-949X


# Image Compression Via Block Truncation Coding and Binary System

[1]Dheiaa Shakir Redhaa and [2]Mohammed Abdul Hameed Jassim Al-Kufi
[1]Ministry of Education, Baghdad, Iraq
[2]Department of Islamic Education, University of Kufa, 31001 Al-Najaf, Iraq

**Abstract:** Because of scientific development especially in computers, the speed of delivered information and saving up stores, it's due to press stores in a way where it does not cause losing information after division of pressing. Therefore, this study will tackle the way of pressing photos depending of using the concept of block truncation coding whereas we will segment three dimensional splint into blocks. We will use the mathematical intermediate normative deviation. For every block, we make each one consists of two colorful amounts. After that, we will use the binary system to press columns that whose graphics are gathered. The rate of compression is high and the quality of photos when using algorithm after completing this compression with using size of block (4×4) excellent.

**Key words:** BTC, image processing, compression, binary system, blocks, algorithm

## INTRODUCTION

The rapid development in all fields especially in information technology which requires save or send various data what concerns us is color and mono color images with various scopes and in order to increase the memory of the computers and to facilitate sending images via. internet, it is required to compress the size of the images to the extent that they will not lose much of their information while decoding, that is to restore the image without distortion (Nivedita, 2012).

Image compression means reducing the data of the file while retaining the necessary information, so that, the image remains clear after decompression (Almrabet *et al*., 2002).

Based on that we must differentiate between the data and the information of the image. Data is the values of the gradation level for a point which represent the value of light and brightness of that point. Whereas the information is a meaningful way of interpretation for the data where it is used to communicate the information similarly in using letters to communicate the information by words (Umbaugh, 1998).

Images compression is a two pronged process, first is compression without losing information of the image while decompression and this is the important part of each algorithmic compression. And the second type is the compression with losing information after decompression (Delp and Mitchell, 1979).

**Block truncation coding:** The concept of numbering the mass had been firstly, presented to practice on the grey image with the development of devices with all fields, become to practice on colorful grey images (Franti *et al*., 1994; Vimala *et al*., 2011).

Numbered images consist of three dimensional trilogy and it consists of colorful images. The idea that the numbered mass of division to blocks with sizes according to the researcher's opinion. The mathematical concepts applied for each mass and according to the sizes and the amounts of the block of image. The process of quantify the colorful amounts of block mainly depends on two concepts: the standard deviation (σ) and arithmetical mean $\bar{x}$ for each block that has been calculated through Eq. 1 and 2:

$$\bar{X} = \frac{1}{m}\sum_{i=0}^{n} x_i \quad (1)$$

where, m is the number of pixels in each blocks:

$$\sigma = \sqrt{\frac{1}{m}\sum_{i=0}^{n}\left(x_i - \bar{X}\right)^2} \quad (2)$$

$x_i$ refers to the ith pixel value of the image block and m refers to number of pixels in that block. Because of colorful values have changed from block to another , therefore, each one has standard deviation and arithmetical mean. The second level of infibulation through equation that calculated to Eq. 3 and 4:

$$x^+ = \bar{X} + \sigma\sqrt{\frac{n^+}{n^-}} \quad (3)$$

**Corresponding Author:** Dheiaa Shakir Redhaa, Ministry of Education, Baghdad, Iraq

$$x^- = \bar{X} - \sigma\sqrt{\frac{n^+}{n^-}} \quad (4)$$

Where:

$n^+$ = The number of pixel values greater than or equal to the mean $\bar{x}$

$n^-$ = The number of pixel whose gray levels are less than the mean $\bar{x}$

$$y(i,j) = \begin{cases} x^+ \text{ if } x(i,j) \geq \bar{x} \\ x^- \text{ if } x(i,j) < \bar{x} \end{cases}$$

After we convert each block in the matrix image to two values: $x^+$ and $x^-$. we now bring the second level of quantization and the pixel $x^+$ changed by putting number (1) and if it is $x^-$, we put (0) at this level all blocks will be converted the image matrix into (1) (0). In this case, we will take the important step of the algorithm for the process of compression is the conversion from the binary system to the decimal system and the operation and change the size of the block to reach the maximum pressure cases as shown below: the given image is divided into non overlapping rectangular regions. For the sake of simplicity the blocks were let to be square regions of size 2×2 or 4×4 or 8×8 or 16×16. Calculate mean $\bar{x}$ and standard deviation σ for each blocks The pixels in each block convert to any two value by Eq. 1 and 2 by the matrix:

$$x(i,j) = \begin{cases} x^+ & y(i,j) \geq \bar{x} \\ x^- & y(i,j) < \bar{x} \end{cases}$$

where, y (i, j) is the pixels of a blocks. For each block, we replace $x^+$ with (1) and $x^-$ with (0). In this way, all blocks will be represented with the above mentioned numbers:

$$y(i,j) = \begin{cases} 1 & \text{if } x(i,j) = x^+ \\ 0 & \text{if } (x(i,j) = x^- \end{cases}$$

where i, j size of blocks: matrix in step (5) turned into a size (n/2, 2n). The resulting matrix in step (6) convert to decimal system. After the conversion to decimal system the value of $x^+$ and $x^-$ for each Block reservation in the last two rows Kapde and Patil,2012; Maharani *et al*., 2013.

**For decompression:**

- After pressing the image matrix is divided into blocks size (n+2/2+1)×1
- Transformation elements of the blocks in the first step (except the last two rows) to binary system
- We return with the block elements in the binary system to the size 2×2 or 4×4 or 8×8 or 16×16
- The number 1 replaces with the value $x^+$ and the number 0 replaces with the value $x^-$

**Example:** For example to block 4×4:

$$A = \begin{bmatrix} 172 & 169 & 155 & 137 \\ 169 & 160 & 144 & 122 \\ 163 & 150 & 132 & 108 \\ 151 & 136 & 114 & 93 \end{bmatrix} \Rightarrow \begin{matrix} \text{Mean } (\bar{x}) \text{ of A} = 142, \\ \text{standard deviation}(\sigma) = 23 \\ n^+ = 9 \text{ and } n^- = 7 \end{matrix}$$

$$y = \begin{bmatrix} 168 & 168 & 168 & 115 \\ 168 & 168 & 168 & 115 \\ 168 & 168 & 115 & 115 \\ 168 & 115 & 115 & 115 \end{bmatrix} \Rightarrow X = \begin{bmatrix} 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 \\ 1 & 1 & 0 & 0 \\ 1 & 0 & 0 & 0 \end{bmatrix}$$

$$B = \begin{bmatrix} 1\,1\,1 & 1\,0\,1 & 1\,0\,0 \\ 1\,1\,1 & 0\,1\,0 & 0\,0\,0 \end{bmatrix}_{2\times 8} \Rightarrow B = \begin{bmatrix} 13 \\ 168 \\ 115 \end{bmatrix}$$

We derive the legal values for each block using $x^+$, $x^-$ and the result by using the arithmetic mean and the standard deviation in the quantization process the values $x^+$, $x^-$ are very close to the original values of the images when applying this algorithm and thus, the error ratio relative to the original image is slim (Table 1 and 2 and Fig. 1).

Table 1: Apply the algorithm on the Lena and child data to more than one block

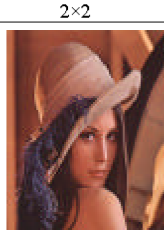
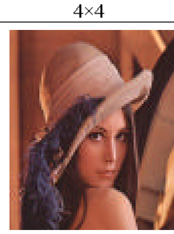
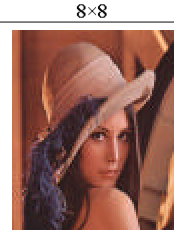
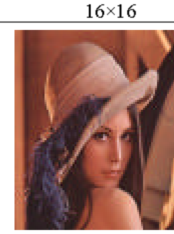
| | Block size | | | |
|---|---|---|---|---|
| Image | 2×2 | 4×4 | 8×8 | 16×16 |
| Lena |  |  |  |  |

Table 1: Continue

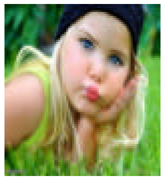
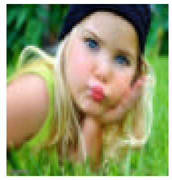
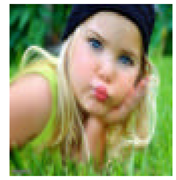
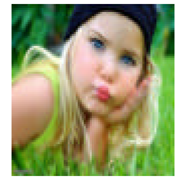
| Image | Block size 2×2 | 4×4 | 8×8 | 16×16 |
|---|---|---|---|---|
| Child |  |  |  |  |

Image: Topic of recursive polygon the compression ratio and time of compression (PSNR) and time of decompression

Table 2: Comparisons with readings of other algorithms

| Algorithm | Notice | Compression ratio | PSNR |
|---|---|---|---|
| DWT | | >>35 | 34.66 |
| K-means | | <25 | 27.36 |
| 3D spiral JPEG | | ≤60 | 31.73 |
| Wavelet | | >>32 | 32.47 |
| JPEG | | ≤50 | 29.64 |
| VQ | | <32 | N/A |
| Fractal | | ≥16 | N/A |
| Proposed algorithm | Image not already compressed | >>150 | 34.16 |
| | Compressed image with JPEG (further compression) | >25 | 34.24 |

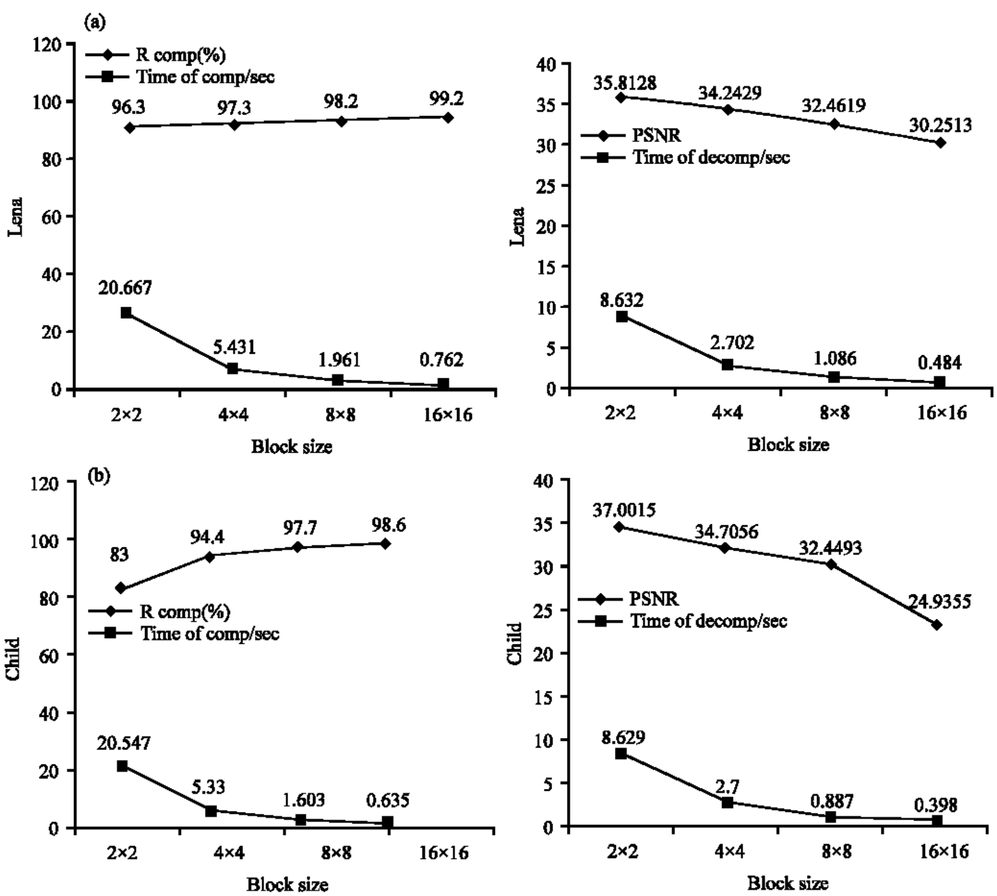



Fig. 2a, b): Replicated polygons to represent the compression ratio and the time of encryption. As well as BSNR and decoding time for both image of Lena and child

## CONCLUSION

- The algorithm gives a high pressure
- The more the size of block bigger, the more the compression is higher but the quality of image is not good enough
- A higher PSNR .we got it with the size of block 4×4 with time of good pressure

## REFERENCES

Almrabet, M.M., A.R. Zerek, A. Chaoui and A.A. Akash, 2009. Image compression using block truncation coding. Intl. J. Sci. Tech. Autom. Control Comput. Eng., 3: 1046-1053.

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

Franti, P., O. Nevalainen and T. Kaukoranta, 1994. Compression of digital images by block truncation coding: A survey. Comput. J., 37: 308-332.

Kapde, S.K. and S.V. Patil, 2012. Image compression using BTC and AMBTC. Intl. J. Adv. Res. Comput. Sci. Electron. Eng., 1: 18-22.

Maharani, M., B.K. Dewi, F.A. Yulianto and B. Purnama, 2013. Digital image compression using graph coloring quantization based on wavelet-SVD. J. Phys. Conf. Ser., 423: 1-8.

Nivedita, S.J., 2012. Performance analysis of svd and spiht algorithm for image compression application. Intl. J. Adv. Res. Comput. Sci. Software Eng., 2: 1-6.

Umbaugh, S.E., 1998. Computer Vision and Image Processing. Prentice Hall, Englewood Cliffs, NJ., Pages: 209.

Vimala S., P. Uma and B. Abidha, 2011. Improved adaptive block truncation coding for image compression. Int. J. Comput. Applic., 195: 1-5.