International Journal of Emerging Trends in Engineering and Basic Sciences (IJEEBS)
ISSN (Online) 2349-6967
Volume 2, Issue 3(May-June 2015), PP 70-77

Modified TiBS Algorithm for Image Compression

Pravin B. Pokle[1], Vaishali Dhumal[2],Jayantkumar Dorave[3]

[123]*(Department of Electronics Engineering, Priyadarshini J.L.College of Engineering/ RTM N University, India)*

Abstract :- In recent times the integration of video, audio and data in telecommunication devices has revolutionized communication world. It has proven to be useful to almost every industry: the corporate world, entertainment industry, multimedia, education and even at home. The major problems associated with these applications are the high data rates, high bandwidth and large memory required for storage and computing resources. Even with faster internet speed, throughput rates and advanced network infrastructure, there are major bottlenecks to transfer such high volume data through the network due to bandwidth limitations. There is a need to develop compression techniques in order to make the best use of available bandwidth. Thus storage and compression of these high resolution images plays a vital role in such applications to conserve the energy and processor's computational resources. This paper presents a lightweight modified TiBS algorithm for image compression and storage. The proposed modified compression method operates on a 3x3 block and is based on pixel removal technique. The results shows that proposed method provides a maximum compression of 33% which is more than that achievable by standard TiBS algorithm.

Keywords :- Image compression, DWT, DCT, TiBS (tiny block-size encoding), Huffman, RLE**.**

I. INTRODUCTION

The increase in the digital imaging techniques in the last few decades has made it possible to capture high resolution images using hand held devices such as digital cameras, mobile phones, ipads etc. Further with the available wireless technologies it is often required to transfer these images wirelessly from a remote location to a destination location as in case of a wireless sensor network. As these digital images are usually represented by a very large number of bits, communication of a single image from source to sink node involves the transmission of several data packets. This results in large energy consumption at the source node than nodes collecting and forwarding scalar data. As the radio transceivers are one of the most power consuming components on sensor nodes, it is obvious that compression of image data would definitely lead to significant energy savings [1]. Image compression, is the science of reducing the amount of data required to represent an image. The use of available compression techniques like JPEG, JPEG 2000, PNG, SPIHT, etc can serve the purpose to compression the image. But more an image is compressed; more is the need for an ARQ (Automatic Repeat-reQuest) or FEC (Forward Error Correction) based protocol to maintain a certain level of image quality. This error control method counter balances the energy saved by compression algorithm. In [2, 6] comparison of energy consumption of JPEG, JPEG2000 and SPIHT is carried out. These algorithms lead to greater energy consumption than the transmission of the uncompressed image. Transform-based image coding algorithms have been the object of intense research during the last twenty years. Eventually they have been selected as the main mechanism of data compression in the definition of digital image and video coding standards. For example JPEG, MPEG-1, MPEG-2, H.261 and H.263 all rely on an algorithm based on the Discrete Cosine Transform (DCT).

Image compression schemes come under two categories: lossless and lossy compression. Losslesscompression uses coding techniques to compress the data while retaining all information content. However, the compression achieved using this method for file size reduction is not sufficient for many applications [3, 9]. On the other hand lossy image compression, as its name implies, results after compression contains the loss of some information content while the file size reduction can be much more significant than obtained with lossless compression. Run-length encoding (RLE) and JPEG compression algorithms are examples of lossless

compression. The Lempel–Ziv (LZ) compression methods are among the most popular algorithms for lossless storage. Wavelet and higher-level JPEG are the examples of lossy compression techniques. JPEG 2000 is a progressive lossless-to-lossy compression algorithm. JPEG handles only still images; also another standard called MPEG is used for motion pictures [8, 10].

JPEG is one of the most used image compression that has revolutionized lossy data compression by providing a method to decrease the size of natural images with minimal loss of perceivable data. The codec uses a combination of DCT, quantization matrices, and entropy encoding and Huffman encoding to discard less relevant data and compress the image. However, the JPEG standard causes harsh blocking artifacts, due to the uniform discretization of the image into 8x8 blocks, when images are compressed to low bit rates [4, 11].

The newer standard, JPEG2000 [12], significantly reduces the distortion present after compression. Here first a global wavelet transform to the image is applied, and then EBCOT (Embedded Block Coding with Optimal Truncation Points) encoding is performed, followed by further entropy encoding [5]. The use of EBCOT and entropy coding after the wavelet transformation allows a highly compressed version of the image with very little distortion. However, the algorithm does not provide a technique to take advantage of local patterns within the image, such as a low frequency area in the background of an image, due to the initial global wavelet transform, which prevents this approach from having the highest amount of compression possible.

As most of digital images are intended for human observers, some loss of information in the digital image can often be unnoticed by the human eye, hence much of the research work nowadays is focused on lossy compression that minimizes visual distortion and possibly obtains visually lossless results. In general in any compression algorithms, there are three basic steps:

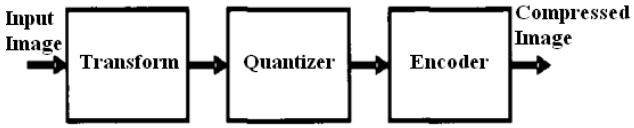


Figure 1: Typical image compression system

In this paper a modified lightweight image compression algorithm called TiBs (tiny block-size image compression), that employs a 3x3 block size instead of a 2x2 block is proposed. The proposed modified TiBS algorithm provides a good trade-off between energy consumption and image quality. Finally a comparison is made between DCT, DWT, Huffman, RLE, TiBS (2x2 block size) and proposed modified TiBS (3x3 block size).

The paper is organized as follows: Section 2 describes the technical principles of the TiBS algorithm. In Section 3 we present the proposed modified TiBS algorithm. Section 4 gives the results of the proposed algorithm and finally, Section 5 cover conclusions and provides some future directions.

II. TIBS ALGORITHM

TiBS is a lossy compression algorithm with very low complexity which provides communication of images in an energy-efficient manner, even for high packet loss rates. Since DCT or DWT is computationally intensive and the Encoder in TiBs does not use DCT or DWT, its working is somewhat different than the conventional structure as shown in figure 1. Figure 2 depict block diagram of TiBS algorithm. This algorithm operates on blocks of 2x2 pixels. Each block is encoded independently, based on three stages: uniform scalar quantization, self-adaptive pixel removal, and variable-length coding. The quantization stage is optional; depending on the user requirements. No quantization stage results in better image quality instead of energy savings. The most original part of TiBS is the self-adaptive pixel removal technique.

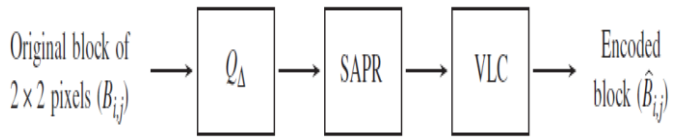


Figure 2: TiBS compression scheme.

Consider a digital image (P) to be a C×R array of pixels and let us assume that, in color plane, the pixel intensities are represented using m bits. Thus the pixel intensities are integers in the range 0 to M, where $M=2^m-1$. TiBS algorithm divides the color plane into non overlapping 2×2 blocks of pixels. Let $B_{i,j}$ denote the block at the i^{th} row and j^{th} column (with $0\leq i<H/2$ and $0\leq j<W/2$). Before encoding process $B_{i,j}$ contains four pixel intensities, denoted $X_{0|i,j}$, $X_{1|i,j}$, $X_{2|i,j}$ and $X_{3|i,j}$ as shown in Figure 3. Let $\hat{B}_{i,j}$ and $\tilde{B}_{i,j}$ denote the encoded and decompressed version of $B_{i,j}$ respectively.

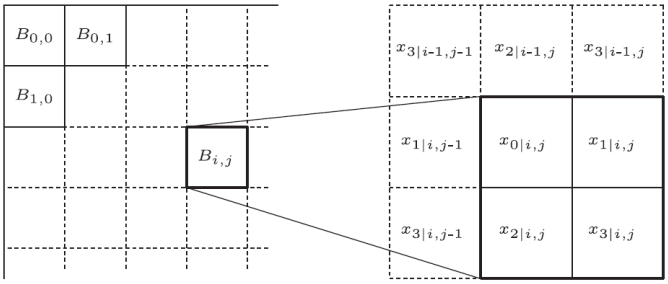


Figure 3: Representation of 2x2 block Bi,j [5].

a) Self-adaptive pixel removal

This step where certain number of pixels from the image are removed in such a way that spatial correlation among the unremoved pixels is retained, so as to estimate the missing pixels at the decoder end. In the standard TiBS algorithm 1-pixel is removed out of the four pixel intensities so that the encoded 2×2 block can be represented with 3×m bits. At the decoder end this missing pixel can be estimated from its surrounding pixels based on spatial error concealment method. One drawback of this method is that the quality of the resulting image cannot be controlled as the magnitude of block distortion can vary from block to block significantly. However this drawback can be dealt with to a good extent by a self-adaptive pixel removal (SAPR) method as proposed in [7]. This SAPR method removes that pixel which induces the smallest block distortion. In this method first $\hat{X}_0|_{i,j}$ and $D_o|i,j$ are computed assuming that $X_o|i,j$ is the removed pixel and so on for the remaining three pixels. Then to find the best pixel to be removed fin k such that $min_k\ (D_k|i,j)$. Finally this pixel is removed and its place is inserted into the LSBs of the three retained pixels, using the transform τ.

b) Uniform scalar quantization

In SAPR method, for each block of 2×2 pixels only one pixel out of four pixels is discarded and hence this achieves a compression ratio of 4:3. Higher compression ratios can be obtained when a scalar quantizer is applied to a block first. Quantization here acts as a rounding off of the input values to decrease the number of distinct output values to a smaller set m′ such that m′< m. As in [7], a simple uniform quantizer, QΔ, such that $Q\Delta(x) = x/2$, where 2Δ is the step size and x is the original pixel intensity, can increase the compression ratio substantially. Basic use of this step is to reduce the number of blocks in an image.

III. PROPOSED MODIFIED TIBS ALGORITHM

In the proposed modified TiBS algorithm, we consider a 3x3 block as shown in figure 4 below. In our proposed modified TiBS method we remove three pixels; the maximum, the minimum and the medium one, out of each 3x3 block. Thus by applying this method each 3x3 block requires 3x2x8 bits as compared to 3x3x8 bits for the uncoded image. By using this method the compression ratio of more than 25% (up to 33%) can be achieved as compared to 25% in case of standard TiBS algorithm discussed above. The encoding and decoding process of proposed method is elaborated with the help of an example below.

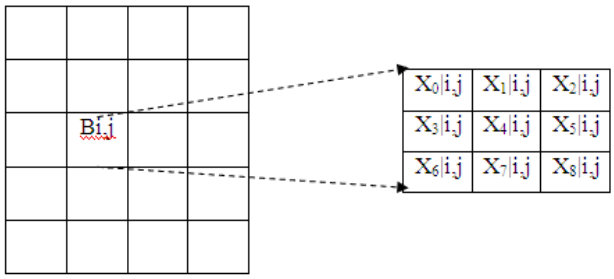


Figure 4: Representation of 3x3 block Bi,j.

A. Encoding Process

Let us consider that the selected 3x3 block consist of following pixel values:

$$\begin{pmatrix} 64^{0} & 128^{1} & 220^{2} \\ 32^{3} & 157^{4} & 64^{5} \\ 78^{6} & 129^{7} & 255^{8} \end{pmatrix}$$

The figures to the top right indicate the index of the pixel values in a 3x3 block. Now out of these the minimum value is 32, the maximum value is 255 and medium value is 128. The medium pixel is found by arranging the pixels in ascending order and selecting the medium value, in this case it will the value at the 5th place of the ascending order. These pixels are removed so this leaves us the below matrix:

$$\begin{pmatrix} 64 & X & 220 \\ X & 157 & 64 \\ 78 & 129 & X \end{pmatrix}$$

The indexes of these removed pixels are 3, 8 and 1 respectively. So this leaves us with the 6 unremoved values as:

$$[64 \quad 220 \quad 157 \quad 64 \quad 78 \quad 129]$$

The index of the minimum value pixel removed is 3, which give 0011 in binary. This binary is replaced with the binary equivalent of the first value i.e.; 64 of the unremoved pixel as shown below;

$$(64)_{10} = (\underline{00}100000)_{8}$$

$$(\underline{00}100011)_{8} = (67)_{10}$$

Similarly the same process is repeated for the maximum pixel i.e., the index of the maximum value pixel removed is 8, which give 1000 in binary. This binary is replaced with the binary equivalent of the second value i.e.; 220

$$(220)_{10} = (\underline{11}011100)_8$$

$$(\underline{11}011000)_8 = (216)_{10}$$

And following the same process for the medium pixel we get:

$$(157)_{10} = (\underline{10}011101)_8$$

$$(\underline{10}010001)_8 = (145)_{10}$$

So this gives the final encoded array of six unremoved pixel which are the transmitted and thus convert our 3x3 block into a 3x2 block of pixels.

[67 216 145 64 78 129]

B. Decoding Process

Now at the decoder side on receiving the 6 pixel values we reconstruct the 3x3 block as follows with following received pixel values:

[67 216 145 64 78 129]

From the first received value we check the four LSB's, for 67 its 0011, this gives us the position of this value in the decoded 3x3 block. Next we check for the four LSB's of the second value, in this case for 216 it is 1000 and finally for the third value, in this case for 145 it is 0001. Now the next step is to arrange the received pixel values in ascending order as:

[64 67 78 129 145 216]

Now the minimum pixel value of the arranged array above is placed 0011=3rd location, the maximum value is place at 1000=8th location and the medium value is place at 0001=1st location in the decoded array as:

$$\begin{pmatrix} \_ & 78 & \_ \\ 64 & \_ & \_ \\ \_ & \_ & 216 \end{pmatrix}$$

Note the medium of the above arranged array is considered as the pixel with value 78. Now to complete the decoding process the remaining positions are filled out serially from the received pixel values. Note here that the order here is not changed it's the same as the pixels were originally received. So we get the finally decoded pixel value as:

$$\begin{pmatrix} 67 & 78 & 216 \\ 64 & 145 & 64 \\ 78 & 129 & 216 \end{pmatrix}$$

So to sum up we have reconstructed the 3x3 pixel array with transmission of only 3x2 values at the decoder side, this is represented graphically as below.

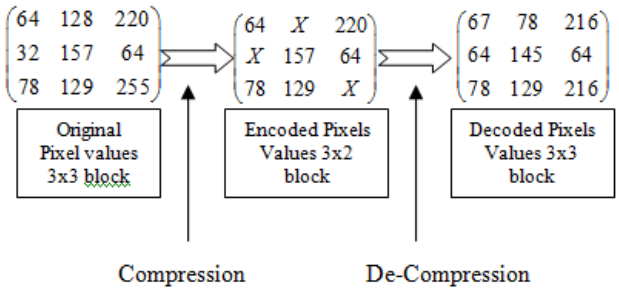


Figure 5: Modified TiBS encoding-decoding Process

Now comparing this with the original pixel values before encoding, from this it is clear that the proposed technique is lossy compression technique as three out of nine pixels are removed, but the result shows that the loss of image information is not that substantial considering the human vision.

IV. RESULTS OF MODIFIED TIBS ALGORITHM

Fig 6: Input image

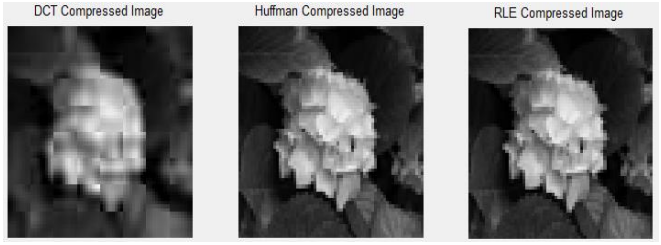


Fig 7: Compressed Image.

Fig 7(a): Compressed Image using standard TiBS.

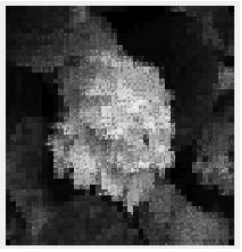
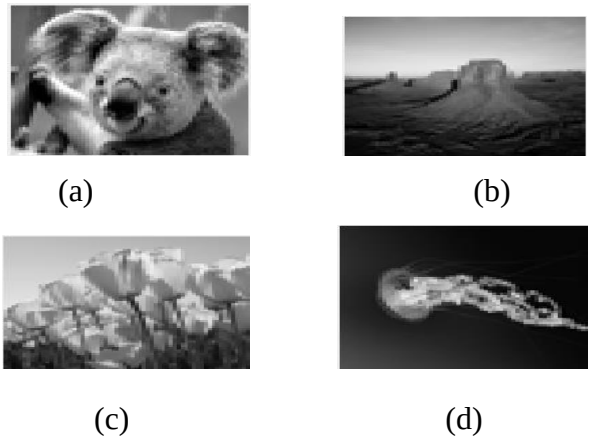


Fig 7(b): Compressed Image using proposed modified TiBS.



(a) (b)

(c) (d)

Fig 8: Test Images.

	Parameters	Huffman	DCT	RLE	TiBS	Modified TiBS
Image 1 Hydrangeas Image (Figure 5)	MMSE (dB)	225.57	37.27	225.57	46.01	53.25
	PSNR (dB)	23.53	15.71	23.53	83.53	83.97
	Compression Ratio (%)	7.181	74.90	51.42	25.0	32.0
	Time Elapsed (s)	0.472	0.138	0.119	0.29	1.30
Image 2 Kuala Image (Figure 8. a)	MMSE (dB)	248.37	55.76	248.37	48.78	56.37
	PSNR (dB)	23.95	17.46	23.95	84.14	84.19
	Compression Ratio (%)	2.97	74.90	48.56	25	29.08
	Time Elapsed (s)	0.359	0.012	0.105	0.28	1.19
Image 3 Grand Canyon Image (Figure 8. b)	MMSE (dB)	242.02	22.11	242.02	41.96	47.03
	PSNR (dB)	23.83	13.44	23.84	83.91	83.93
	Compression Ratio (%)	3.26	74.90	48.93	25	23.32
	Time Elapsed (s)	0.358	0.010	0.104	0.28	1.19
Image 4 Tulip Image (Figure 8. c)	MMSE (dB)	254.69	50.25	254.69	47.67	56.30
	PSNR (dB)	24.06	17.01	24.06	84.23	8 4.23
	Compression Ratio (%)	4.81	74.90	47.52	25	24.42
	Time Elapsed (s)	0.32	0.010	0.103	0.27	1.17
Image 5 Jelly Fish Image (Figure 8. d)	MMSE (dB)	197.43	20.79	197.43	46.01	50.83
	PSNR (dB)	22.95	13.18	22.95	80.0	81.98
	Compression Ratio (%)	20.04	74.90	55.71	25	30.67
	Time Elapsed (s)	0.268	0.011	0.102	0.28	1.26

V. CONCLUSION

In this paper, a modified TiBS algorithm is proposed which uses a 3x3 block and removes 3 pixels from each of these blocks. The results obtained for different images show that the proposed modified TiBS algorithm has better compression ratio and PSNR values as compared to the standard TiBS algorithm. In future certain modifications can be made in the proposed method for further improvement of the PSNR and compression ratio.

REFERENCES

[1] I.F.Akyildiz, W.Su, Y.Sankarasubramaniam and E.Cayirci, "Wireless Sensor Networks: A Survey", Computer Networks 38(4)(2002) 393–422.

[2] Ferrigno L, Marano S, Paciello V, Pietrosanto A. Pietrosanto. Balancing computational and transmission power consumption in wireless image sensor networks. International Conference on Virtual Environments, Human-Computer Interfaces, and Measures Systems. Italy. 2005: 61–66

[3] Shantanu D. Rane and Guillermo Sapiro, *Member, IEEE*,"Evaluation of JPEG-LS, the New Lossless and Controlled-Lossy Still Image Compression Standard, for Compression of High-Resolution Elevation Data", IEEE Transactions on Geoscience and Remote sensing, VOL. 39, NO. 10, Oct. 2001

[4] Med Lassaad KADDACHI, Adel SOUDANI, Ibtihel NOUlRA, Vincent LECUlRE and Kholdoun TORKI "Efficient hardware solution for low power and adaptive image-compression in WSN", 978-1-4244-8157-6110, ICECS 2010, IEEE.

[5] Zhang, Y. Z., Xu, C., Wang, W. T., & Chen, L. B. Performance Analysis and architecture design for EBCOT encoder in JPEG2000. IEEE Transactions on Circuits and Systems for Video Technology, 17(10), 1336–1347.-2007.

[6] C. Duran-Faundez, V. Lecuire, "Error resilient image communication with chaotic pixel interleaving for wireless camera sensors", Workshop on Real-World Wireless Sensor Networks (REALWSN 2008), ACM, Glasgow, Scotland (2008), pp. 21–25.

[7] Cristian Duran-Faundez, Vincent Lecuire, Francis Lepage, "Tiny block-size coding for energy-efficient image compression and communication in wireless camera sensor networks", Signal Processing: Image Communication 26 (2011) 466–481. 2011 Elsevier

[8] Z. Wang, A. C. Bovik, H. R. Sheikh and E. P. Simoncelli, "Image quality assessment: From error visibility to structural similarity," IEEE Transactions on Image Processing, Vol. 13, no. 4, 2004.

[9] B. Goossens, A. Pizurica, and W. Philips, "Removal of correlated noise by modeling the signal of interest in the wavelet domain," *IEEE Trans.Image Process.*, vol. 18, no. 6, pp. 1153–1165, Jun. 2009.

[1]0 R. C. Gonzalez and R. E. Woods, "*Digital Image Processing*", 2nd Ed., Prentice Hall, 2004.

[11] Wallace, G. K. "The JPEG Still Picture Compression Standard", Comm. ACM, vol. 34, no. 4, April 1991, pp. 30- 44.

[12] D.S.Cruz, T.Ebrahimi, " A Study of jpeg 2000 Still Image Coding versus other standards" <http:/www.jpeg.org/public/>, ISO/IECJTC1/SC29/WG1 N1814, July 2000.