Volume 8 Issue 2 (2020) 144-147 ISSN 2347 - 3258
International Journal of Advance Research and Innovation

Simultaneous Image Encryption and Compression using Adaptive Bit Plane Quadtree-based BTC

Pranob K Charles, K Jyothi, E Amulya, P Sai Rani
Department of Electronics and Communication Engineering, Andhra Loyola Institute of Engineering and Technology Vijayawada,-520008, Andhra Pradesh, India.

Article Info
Article history:
Received 17 January 2020
Received in revised form
20 May 2020
Accepted 28 May 2020
Available online 15 June 2020

Keywords: JPEG, encryption, compression, decryption, quantization, quadtree, BTC.

Abstract: In the digital World, huge amount of data is shared or stored every day. While sharing the data there may be a chance accessing of the data by an unauthorized persons. So In this study, by modifying standard JPEG compression algorithm, encryption and simultaneous decompression, decryption method is inspected. The proposed system deals with the computation of time and cost by combining compression and encryption by using Adaptive Bit plane Quad tree Block Truncation Coding (ABTC) technique. Here combined with edge detection and adaptive bit plane quantization, the adaptive bit plane quad tree-based BTC algorithm is used. The encryption of the data can be done with key validation technique. The main and ultimate goal is to secure the data from the unauthorized persons, to reduce the redundancy of the image and to store or transmit data in an efficient form.

1. Introduction

In Image processing, use of Image Compression and Encryption leads in transmitting an image securely and in compressed form through unsecured and low bandwidth channel. In image processing, for providing security to an image many encryption techniques are available. But most of the Encryption techniques mask some amount of data to the source image that always increase the size of the image. Encryption makes it difficult to transmit an image through band width constrain channel. To overcome this problem Image Compression can be applied on the encrypted image to reduce its size. But traditional compression algorithms underperform for many of the image encryption techniques. Applying the Hybrid approach increases the computational complexity. This paper discusses about some techniques which are used for performing image Encryption, Image Compression.

Standard Bitmap file(BMP) format also known as Bit Map Image file or device independent bitmap file format or simply a bitmap, is a raster graphics image file format used to store bitmap digital images, independently of the display device. A bitmap graphics is composed of many tiny parts called pixels, which are Red, Green and Blue(RGB) channel intensities.JPEG (Joint Photographic Experts Group) is a wide spread compression algorithm JPEG is a lossy image compression method. It is possible to first encrypt raw image then immediately use the compression. It will reduce the time and cost.

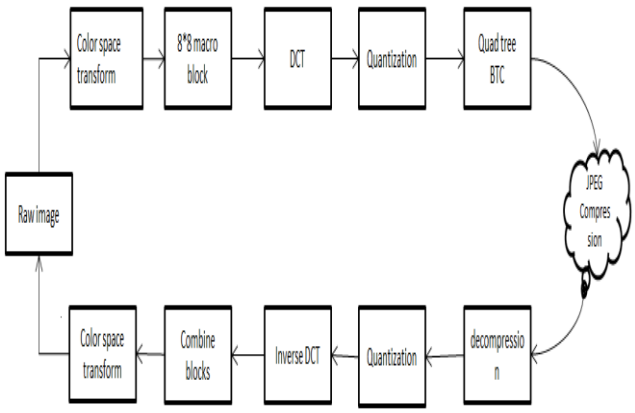

Fig. 1:JPEG Algorithm steps

The problem of large computation time and cost can be resolved by using the proposed method. In the 2nd chapter, JPEG Algorithm can be explained. Respectively in the 3rd chapter, the proposed compression algorithm is explained. Finally, the result can be discussed in 4th chapter.

***Corresponding Author,**
E-mail address: kathilijyothi023@gmail.com;

2. JPEG compression algorithm

2.1. Raw Image

The raw image contains minimally processes data from the image sensor of either a digital camera, a motion picture film scanner or other digital image scanner. It is possible to first encrypt the raw image then immediately use the compression.

2.2. Color Space Transform

Color Space conversion is the transformation of the representation of a image from one color space to another color space. The translated image is similar to the original image. Real time images and videos are stored in RGB color space because it is based on sensitivity of color detection cells in the human visual system.

The *Ycbcr* color spaces is used for the lower resolution capability of the human visual system. The standard RGB image is transformed into *Ycbcr*

Y – brightness, cb – luminance , cr – chrominance.

The image space is sub scaled from 4:4:4 to 4:2:2.

2.3. 8*8 Macro Block Separation

First an image need to be separated for 8*8 pixel blocks. That means each block has 8*8 pixels and it has 64 pixels in one block. It is also possible to separate the image into 16*16 pixels .Each algorithm steps are applied to macro blocks.

2.4. Forward DCT Transformation

Discrete Cosine Transformation is applied to each macro block. The DCT works by separating image into parts of different frequencies. This method eliminates the high frequencies in the image because human eye is not sensitive to the very high frequency changes of the image. The DCT creates a new matrix that has the values in left upper corner of the matrix and other less dominant components are located at lower right side. It contain one DC component with lowest possible value and zero AC components i.e., rest of the matrices is zero.

2.5. Quantization

DCT values near to zero is converted to the zero and other elements also will be shrink towards zero. Then each value of the resultant matrix is divided by another matrix called Standard JPEG Quantization table, which includes quantization of DCT symbols into absolute nearest integer.

3. Existing System

In this existing model, the images cannot be simultaneously encrypted and compressed .Here different algorithms has to be used for encryption and compression. This will not immediately prevent the intermission data listing since unauthorized persons will be able to make sense of the encrypted data. In existing system, time consuming is more and it is of highly expensive and more complex.

At present, special domain image encryption and compression algorithms have problems such as poor encryption and image compression, long time consuming of encryption and compression, and no guarantee of image compression quality. Hence,

simultaneous image encryption and compressed algorithms are proposed.

4. Background

4.1.AMBTC

The goal of AMBTC is to preserve the mean and the first absolute central moment of image blocks. It first partitions the image into 16 × 16 non-overlapping blocks. For each block, the block mean is first calculated by averaging the k pixel values in the current block. Then, using the mean value for threshold two quantization levels are calculated by averaging the pixels whose values are greater or smaller than block mean (formula 1). The two quantization levels are then sent to the decoder using 8+8 bit.

$$a = \frac{1}{k-q}\sum_{x_i<\bar{x}} x_i \,, b = \frac{1}{q}\sum_{x_i\geq\bar{x}} x_i \tag{1}$$

Here, q denotes the number of pixels greater or equal to threshold. For each pixel in the block, there is a corresponding bit in the bit plane that records which quantization level is used to encode this pixel. Thus, a trio consisting of two quantization levels and a bit plane, 32 bits per block, is sent to the decoder. AMBTC needs very low computation ability and memory space, while the reconstructed image is acceptable for most applications. However, the bit rate is fixed at 2 bits per pixel when encoding, which prevents it from being an excellent image code.

4.2.Quardtree BTC

The quadtree segmentation technique is a hierarchical decomposition technique that partitions image into variable sized blocks based on the quad-tree structure (Fig.2). It first determines the maximum and minimum block size and processes block by block. For each block, if the current block satisfies the threshold criterions, that means the current block is inactive, so it can be replaced by its mean value. Otherwise, it will be subdivided into four sub-blocks. Then each subblock is processed recursively until threshold criterions is satisfied or the minimal block size is reached. The most important part in quadtree-based BTC is how to encode minimal active blocks. One can use all kinds of BTC variants, even some new schemes specially designed to deal with active block. All of these methods include encoding one indicator bit, quantization data and bit plane information. In decoding phase, the indicator bit is first decoded, deciding whether the current block is active or not. Using this information, the decoder knows how many bits should be read from the bit stream and their exact meaning. This procedure is done block by block until the whole image has been reconstructed.

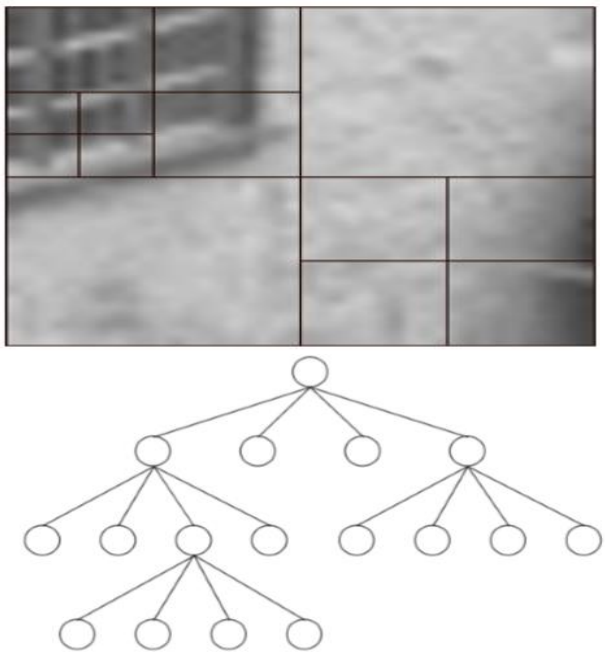

Fig.2: Original image and its quadtree structure

5. Proposed Method

First, the original image is partitioned into non-overlapping blocks. Then, Sobel operator is applied on each block to detect the direction of edge in each block, and the current block is decided whether it should be further subdivided horizontally or vertically. For blocks with different row and column, divide it into two sub blocks of the same size. For the minimal block, we first encode it by AMBTC and attempt to decode it using two quantization levels in encoding phase. If the MSE generated from this procedure is greater than the post-threshold, a 2-bit-plane quantization scheme is used.

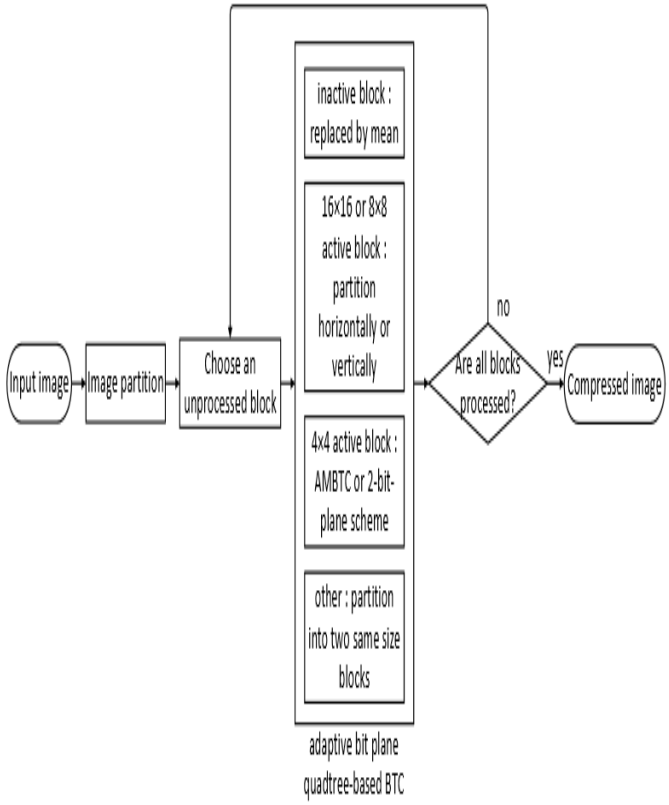

Fig.3: Framework of proposed method

5.1 Edge detection

Here Sobel operator is used to detect possible edge in the block. The correlation between current block and two Sobel operators.

$$w(x,y) * f(x,y) = \sum_{s=-1}^{1}\sum_{t=-1}^{1} w(s,t) f(x+s, y+t) \tag{2}$$

Here, $w(x,y)$denotes 3 × 3 Sobel operator and denotes the pixel value in the image. To simplify the computation, valid part of pixel in the original image without padding is used. Thus, the size of correlation result is smaller than that of the original block. Then, the summation of absolute of result are calculated. If one is greater than the other one, it has higher probability of having an edge in the corresponding direction. So utilizing the edge information to divide the current block, making its sub blocks more likely to be an inactive one, which could be replaced by their mean value. Thus, there may be less active blocks, therefore reducing the bit rate efficiently. Note that any block segmentation using the quadtree can be achieved using our method, though more levels in the tree structure may be needed. However, the reverse is not true.

5.2. Two bit plane quantization

For 2 bit plane, there are 4 quantization levels for a block, denoted by quan(i) (i=1,2,3,4). We use uniform quantization levels to save the bit rate. In order to prevent the maximum or minimum pixel value dominating the result, we use average of the four largest pixels for the highest quantization level quan(4), and average of the four smallest pixels for the lowest quantization level quan(1). Then, other quantization level can be calculated by formula 3

$$quan(i) = \frac{4-i}{3} quan(1) + \frac{i-1}{3} quan(4) (i = 1,2,3,4) \tag{3}$$

Thus, we only need to encode the lowest quantization level and step length obtained in formula 4.

$$l = \frac{quan(4) - quan(1)}{3} \quad (4)$$

We use 8 bits to represent quan(1) and 7 bits for step length, since its value is no more than 127. As for bit plane, each pixel is quantized to nearest quantization level using formula 5 and represented by 00,01,10,11, respectively.

$$index(i) = \min(abs(x_t - quan(i)))(t = 1,2,...16, i = 1,2,3,4) \quad (5)$$

Here, x denotes each pixel in the block. Adding two additional indicator bits (one for active block another one for 2 bit plane coding decision), the active 4 × 4 block is encoded by totally 49 bits. Indeed, the bit rate of this block is rather high. However, just like what we analyzed in section 2.1, there are only a few blocks occupy most of the MSE loss. So we can adjust the post-threshold to control how many blocks are allowed to be encoded by our scheme.

In decoding phase, two indicator bits are first decoded to decide how to reconstruct the current block. Then 15 bits are read as first quantization level and step length, and four quantization levels are calculated using formula 6

$$quan(i) = quan(1) + (i-1) * l(i = 1,2,3,4) \quad (6)$$

Finally, the bit plane is decoded and the block is reconstructed by corresponding quantization level.

5.3.Proposed Compression Algorithm

Combined with edge detection and adaptive bit plane quantization, the proposed adaptive bit plane quadtree-based BTC algorithm is as follow:

Step 1: If the image is color image, decompose it into red, green and blue channel and process each channel as grey image separately. Otherwise, proceed to Step 2.

Step 2: Partition the image into 16 × 16 non-overlapping blocks.

Step 3: For each unprocessed 16 × 16 image block, if its variance is smaller than pre-threshold, encode this 16 × 16 block using its block mean and proceed to Step 6. Otherwise, proceed to Step 4.

Step 4: According to current block size and its variance, choose how to process it.

4.1 If the variance of current block is smaller than pre-threshold, encode this block using its block mean and proceed to Step 6. Otherwise, the current block is active.

4.2 If the current block size is 4 × 4, proceed to Step 5.

4.3 If the current block size is 16 × 16 or 8 × 8, use the edge detection technique, and divide the current block into two sub blocks. For each sub block, return to Step 4.

4.4 Otherwise, the current block has different sizes of row and column. Just divide it into two sub blocks with the same size and return to Step 4.

Step 5: The 4×4 active block is first encoded by AMBTC introduced. Then use two quantization levels and bit plane information to decode immediately. If the MSE generated from reconstructed block is smaller than post threshold, encode it by AMBTC. Otherwise, encode it by 2-bit-plane scheme.

Step 6: If there are still any 16 × 16 blocks to be processed, return to Step 3.

6. Experimental Result

In this section, extensive experiments are carried out to validate the performance of our compression scheme. Any of the images available in matlab can be taken.

By trying different combinations with pre- and post-threshold, the rate-distortion performance of Lena is shown in Table 1. From the results, some key points are to be noticed: (1) Bit rate mainly depends on pre-threshold. The larger prethreshold is, the lower bit rate will be due to more blocks are replaced by mean values. But the influence of changing the post-threshold is insignificant, even in large post-threshold region; (2) At the same pre-threshold, PSNR doesn't change uniformly following the change of post-threshold. When post-threshold is slightly higher than pre-threshold, PSNR decreases sharply as the post-threshold increases. However, it changes more smoothly when the post-threshold is relatively high; (3) Computational complexity mainly depends on post-threshold. Because the 2 bit plane quantization is more time-consuming than AMBTC, when post-threshold is high, there are only several blocks encoded by 2 bit plane. Compressed image shown in figure 5 to 7.

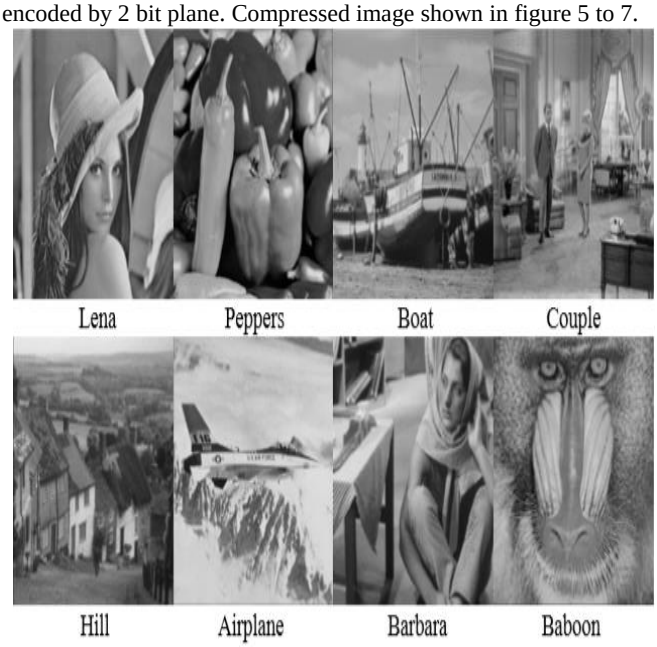

Fig. 4: Standard test images

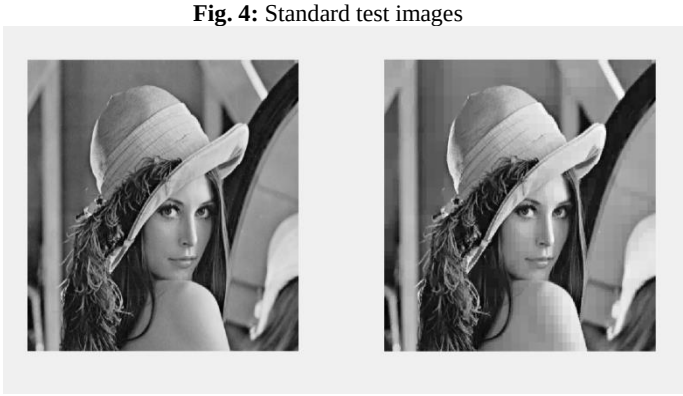

Fig.5: original image vs output image.

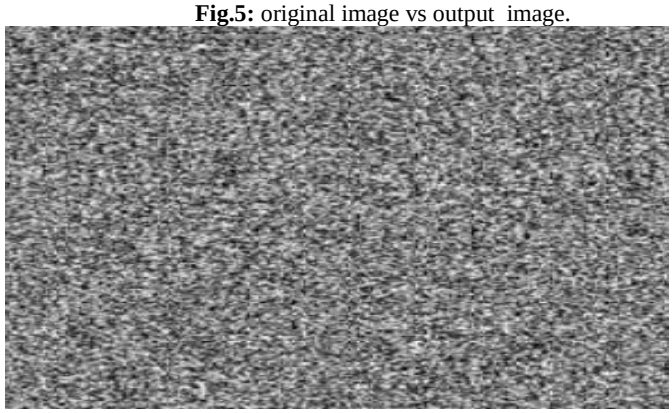

Fig.6: Encrypted image

Fig.7: Compressed image

7. Conclusions

Many of the current image compression and encryption techniques have been presented and analyzed. The best way of fast and secure transmission is by using compression and encryption of multimedia data like images. The compression technique observed is lossy or lossless. Always lossless compression is preferred. But to achieve secrecy some image quality degradation is accepted. The proposed method deals with the quadtree-based block truncation coding algorithm combined with adaptive bit plane quantization. By adjusting the pre- and post-threshold, we balance the PSNR, bit rate and computation complexity, and get the rate-distortion performance of the proposed method. The use of adaptive bit plane, targeted at high MSE loss block, can compensate the degradation of image quality and occurrence of blocking effect. As a result, the proposed method combines those two techniques, which can both improve the image quality and decrease bit rate at the same time.

REFERENCES

[1] S Lian, D Kanellopoulos, G Ruffo. Recent Advances in Multimedia Information System Security, International Journal of Computing and Informatics, 33(1), 2009, 3-24

[2] GK Wallace. The JPEG still picture compression standard, IEEE Transactions on Consumer Electronics, 38(1), 1992, 14-22.

[3] Xilinx Appl. Note, XAPP616 (v1.0) April 22, 2003

[4] CP Wu, CCJ Kuo. Design of integrated multimedia compression and ecryption systems, IEEE Trans. Multimedia, 7(5), 2005, 828-839.

[5] CP Wu, CCJ Kuo. Fast encryption methods for audio-visualdata confidentiality, inSPIE Int. Symp. Information Technologies, 4209(11), 2000, 284.

[6] J Nechvatal. Report on the Development of the Advanced Encryption Standard, National Institute of Standards and Technology, U.S. Dept. Commerce, Tech. Rep., 10, 2009

[7] EJ Delp, OR Mitchell. Image coding using block truncation coding, IEEE Trans. Comm. 27, 1979, 1335–1342.

[8] P Fränti, O Nevalainen, T Kaukoranta. Compression of Digital Images by Block Truncation Coding: A Survey, Computer Journal 37(4), 1994, 308-332.

[9] M Lema, OR Mitchell. Absolute Moment Block Truncation Coding and Its Application to Color Images, Comm. IEEE Trans. on 32(10), 1984, 1148-1157.