CHAPTER 3

Review of Block Truncation Coding based Image Compression Techniques

3.1. Introduction

The aim of the image compression is to transform the image to a compressed form so that the information content is preserved as far as possible when decompressing the encoded image. The performance of any image compression method is measured using a set of features. The features include the bit-rate, which is the number of bits required to store the intensity value (gray level) of one pixel. Bit-rate is the prime factor to be considered when measuring the performance of an image compression technique. Another feature is the quality of the reconstructed image. A technique in which the compressed image is reconstructed with out any loss of data is called the lossless technique. If there is any loss, it is called a lossy technique. The third feature is the time taken to compress and decompress the image. In addition to these features, two other features: ease of implementation and the computational complexity, are also considered to measure the efficiency of the image compression algorithms. Block Truncation Coding (BTC) is one such algorithm that satisfies all the above features, except that it is a lossy compression technique. But the loss in image data is acceptable by the human vision system. In this chapter, the BTC method and its variants are explained.

3.2. Block Truncation Coding

Block truncation coding is a simple and fast lossy compression technique for digitized gray scale images originally introduced by [Delp and Mitchell, 1979]. When compared to other image compression techniques, BTC requires less computational effort and has good capability of combating channel errors [Soo-Chang Pei and Ching-Min, 1997]. The key idea of BTC is to perform moment preserving quantization for blocks of pixels so that the quality of image will remain acceptable and at the same time the demand for the storage space will decrease. Even if the compression gain of the BTC algorithm is inferior to the standard JPEG algorithm [Pennebaker et' al., 1993], BTC has gained popularity due to its special usefulness. Several improvements of the basic method have been recently proposed in the literature. In BTC, the input image of size m x m pixels is divided into blocks of size 4 x 4 pixels. Each block is processed independent of each other. The mean value $\bar{x}$ and the standard deviation σ are calculated for each block using the equations:

$$\bar{x} = \frac{1}{k} \sum_{i=1}^{k} x_i \tag{3.1}$$

$$\overline{x^2} = \frac{1}{k} \sum_{i=1}^{k} x_i^2 \tag{3.2}$$

$$\sigma = \sqrt{\overline{x^2} - \left(\bar{x}\right)^2} \tag{3.3}$$

where k is the number of pixels in a block.

A two-level quantization is done by transforming each pixel value into either 1 or 0. If the pixel value x_i is greater than or equal to the mean $\bar{x}$, then the pixel value x_i is transformed to 1 otherwise the 0. This collection of 1's and 0's is called a bit-plane (B) of that block. The two statistical moments $\bar{x}$ and σ are preserved and are transmitted or stored along with the bit-plane. Hence, a compressed block is a set of {$\bar{x}$, σ and B}. For an uncompressed image, each block of pixels requires 16 x 8 = 128 bits (in case of a gray scale image, where the bpp is 8) per block. But for an image compressed using BTC, a block requires only 32 bits (8 bits for $\bar{x}$, 8 bits for σ and 16 bits for the bit-plane) leading to a bit-rate of only 2 bpp.

In general, the bit-rate is computed as:

$$(b + b + k)/k = 1 + 2m/k \text{ bits per pixel (bpp)}$$

Where, b = $Log_2(L)$, L is the maximum gray level intensity and k is the number of pixels in a block. The value of L is 256 for a gray scale image with 256 gray levels. While decoding, two quantizing levels q_1 and q_2 are calculated using the equations:

$$q_1 = \bar{x} - \sigma . \sqrt{\frac{q}{(16 - q)}} \tag{3.4}$$

$$q_2 = \bar{x} + \sigma . \sqrt{\frac{(16 - q)}{q}} \tag{3.5}$$

Where, q denotes the number of 1's in the bit plane. Each 1 in the bit-plane is replaced with q_2 and 0 is replaced with q_1. This gives the reconstructed image. This method is fast, easy to implement and has low computational complexity.

BTC Algorithm

Step1: Divide the image into small non-overlapping blocks of size 4 x 4 pixels.

Step2: Compute the statistical moments $\bar{x}$ and σ of the block using the equations 1, 2 and 3.

Step3: Generate the bit-plane with 0's and 1's.

Step4: Transmit or store the bit-plane, $\bar{x}$ and σ.

Step5: Repeat the steps 2 through 4 for all the blocks of the input image.

Example:

Original image block — Bit-plane — Reconstructed Image block

$$\begin{pmatrix} 156 & 159 & 158 & 155 \\ 160 & 154 & 157 & 158 \\ 156 & 159 & 158 & 155 \\ 160 & 154 & 157 & 158 \end{pmatrix} \xrightarrow{\text{encode}} \begin{pmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \end{pmatrix} \xrightarrow{\text{decode}} \begin{pmatrix} 155 & 159 & 159 & 155 \\ 159 & 155 & 155 & 159 \\ 155 & 159 & 159 & 155 \\ 159 & 155 & 155 & 159 \end{pmatrix}$$

$\bar{x} = 157.13$, $\sigma = 1.90$ — $p = 8$ — $q1 = 155.23$, $q2 = 159.02$

Advantages of BTC

- Fast
- Requires little extra memory
- Easy to implement
- Low computational complexity
- Less prone to transmission errors.

Drawbacks of BTC

- High bit-rate when compared to techniques such as DCT, VQ, etc.
- Blocky appearance in the reconstructed image in some cases
- Ragged edges

The BTC method described above can be extended to color images by applying the BTC technique to each of the color planes. [M.D.Lema and O.R.Mitchel, 1984] reported 6 bits per pixel with good quality color images based on empirical results.

Variants of BTC

There are several variants of BTC. Some of the methods are presented in [Delp and Mitchell, 1979; Efrati N., et. al., 1991; Goeddel T., et. al., 1981; Halverson D., et. Al., 1984; Lema M.D., et. al., 1984; Qiu G., et. al., 1991]. The aim of these developments is the reduction of bit-rate. The greatest deficiency of BTC is the bit-rate when compared to other image compression techniques such as DCT [Ahmed N., Natarajan T. and Rao K.R., 1974], Vector Quantization [Gersho A. and Gray R.M., 1992; Nasrabadi N.M. and King R.A., 1988], etc.

Heuristic Approach

A heuristic approach was proposed by Goeddel and Bass, 1981. In this method, the threshold value is calculated as

$$x_t = \frac{(x_{min} + x_{max})}{2} \tag{3.6}$$

where x_{min} and x_{max} stand for the minimum and minimum pixels values of a block.

This approach gives better MSE value.

Absolute Moment Block Truncation Coding

Absolute Moment Block Truncation Coding (AMBTC) by [Lema M.D. and Mitchel O.R., 1984] preserves $\bar{x}$ and the sample first absolute central moment

$$\bar{\alpha} = \frac{1}{m} \cdot \sum_{1}^{16} \left| x_i - \bar{x} \right| \tag{3.7}$$

Quantization levels are calculated from

$$a = \bar{x} - \frac{\bar{\alpha}}{2} \cdot \frac{16}{16 - q} \tag{3.8}$$

$$b = \bar{x} + \frac{\bar{\alpha}}{2} \cdot \frac{m}{q} \tag{3.9}$$

Where, $\bar{x}$ is the mean value of the pixels in that block.

The values a and b can be proved as the lower mean $\overline{x_l}$ and the higher mean $\overline{x_h}$, where

$$\overline{x_l} = \frac{1}{16 - q} \cdot \sum_{x_i < \bar{x}} x_i \tag{3.10}$$

$$\overline{x_h} = \frac{1}{q} \cdot \sum_{x_i >= \bar{x}} x_i \tag{3.11}$$

While encoding, the bit plane B, the $\overline{x_l}$ and $\overline{x_h}$ are stored in the compressed file. While decoding, all the 1's in the bit-plane are replaced with the $\overline{x_h}$ and the 0's are replaced with $\overline{x_l}$. AMBTC reduces the MSE when compared to that of BTC. Also the coding and decoding processes are very fast because the square root and multiplication operations are omitted. The bit rate remains same as that of BTC.

Adaptive Block Truncation Coding

In Adaptive Block Truncation Coding (ABTC) [Lucas Hui, 1990], a four level quantizer is designed. The input image blocks are categorized into three types namely: low-activity blocks, medium-activity blocks and high-activity blocks. In low-activity blocks, all the pixels will have more or less same gray level. Hence a one-level quantizer is designed with the mean value. This requires only 0.5 bpp. For medium-activity blocks, two-level quantizer is designed as standard BTC and requires 2 bpp. The 2-level quantizer has been given in Figure 3.1. For high-activity blocks, a four-level quantizer is designed with quantizers being x_{min}, $(2x_{min}+x_{max})/3$, $(x_{min}+2x_{max})/3$, x_{max}. Hence the bit-rate for high-activity blocks is 4 bpp. The ABTC gives better MSE when compared to standard BTC and AMBTC.

2-Level quantizer of BTC

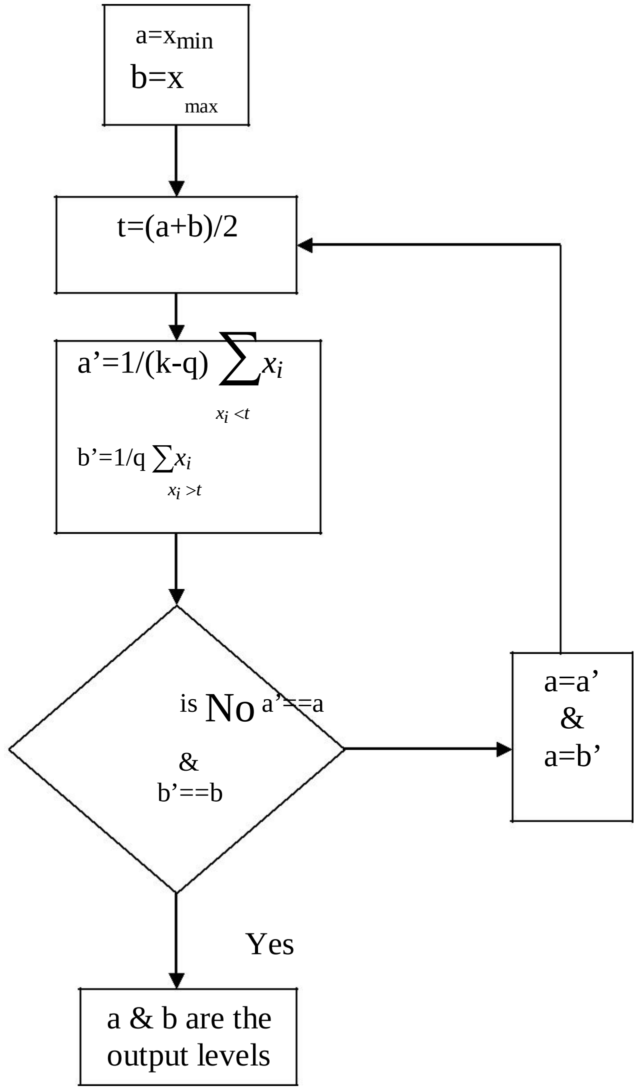

Figure 3.1.: 2-Level quantizer of BTC

AMBTC based on MSE criterion

[Madhu Shandilya and Rajesh Shandilya, 2003] proposed an idea of multilevel quantizer having n levels. The threshold values for the multilevel quantizer can be calculated by

$$th_r = m_n + ((m_x - m_n)r / n) \tag{3.12}$$

where m_n and m_x are the minimum and maximum intensity values of the block respectively. r takes the values ranging from 1 to n. th_r represents the r^{th} value of threshold and n is the number of quantization levels. To compute the first threshold value, r takes the value 1, and so on. As the quantization level increases, the bpp increases with better MSE. Somasundaram and Kaspar Raj, 2006 proposed a multi-stage still gray image compressor based on BTC. They employed a combination of four techniques to reduce the bit rate. They are quad-tree segmentation, bit plane omission, bit plane coding using 32 visual patterns and interpolative bit plane coding and achieved a bit rate of o.46 bpp with an average PSNR value of 30.25. There has been an algorithm by [Halverson, D.R., 1984] for preserving moments which results in less mean square error (MSE). The coding algorithm described by [Udpikar.D. and Raina J.P., 1985] preserves the higher mean and the lower mean of the small blocks. So the statistical overhead to be coded per block is a pair of mean values. The truncated block of the BTC output is the bitplane representing the block consisting of a one-bit output of the quantizer for every pixel in the block. This algorithm involves less computational complexity. Some of the important efforts to improve the coding efficiency are reported in [Healy D.J. and Mitchell O.R., 1981and Gonzalo A. and Gallagher, 1983].