Journal of Engineering and Applied Sciences 3 (9): 664-668, 2008
ISSN: 1816-949X

Storing Fingerprint Images in (3×3) Block Compression Lossless Method

Salwa Adel
Department of Electrical and Electronic Engineering,
University of Technology, Engineering, Baghdad, Iraq

Abstract: Image compression is the application of data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. It involves reducing the size of image data files, while retaining necessary information. This study is concerned, with compression method of fingerprint image without losing the important features of these images (lossless method), by using, (3×3) Block Compression Method. This proposed method done by compressing the Fingerprint (FP) image and decompressing the image for the (3×3) blocks of image, coded and decoded them. And then compare the result, which come from experimental results for (size of image, quality of image) and find the relationship between them.

Key words: Image data, compression, imformation, block comperssion method, fingerprint

INTRODUCTION

Fingerprint has been in use as personal identification for along time. It's unique for individual and does not change throughout one's life; this makes it an ideal signature of a person. Fingerprint can be defined as collection of curves; each has a specified thickness, which depends on the ink of print, the age of the person and the type of scanner. Henry system, which is the most widely used and which is known as Henry classification, the real significance of fingerprint patterns, there is a variety of patterns, for the purposes of the Henry system they are divided into 4 main groups, namely: Arch-Loop-Whorl-Compounds (Cherrill, 1954).

There are numerous applications of image processing, such as satellite imaging, medical imaging and video where the image size or image stream size is too large and requires a large amount of storage space or high bandwidth for communication in its original form. Image compression techniques can be used effectively in such applications. Lossless (reversible) image compression techniques preserve the information so that exact reconstruction of the image is possible from the compressed data (Sahni *et al*., 1997).

Image compression is the application of Data compression on digital images. In effect, the objective is to reduce redundancy of the image data in order to be able to store or transmit data in an efficient form. Image compression can be lossy or lossless. Lossless compression is sometimes preferred for artificial images such as technical drawings, icons or comics. This is because lossy compression methods, especially when used at low bit rates, introduce compression artifacts. Lossless compression methods may also be preferred for high value content, such as medical imagery or image scans made for archival purposes. Lossy methods are especially suitable for natural images such as photos in applications where minor (sometimes imperceptible) loss of fidelity is acceptable to achieve a substantial reduction in bit rate (Xiong, 2003).

Lossless compression involves with compressing data which, when decompressed, will be an exact replica of the original data. This is the case, when binary data such as executables, documents etc. are compressed. They need to be exactly reproduced when decompressed. On the other hand, images (and music too) need not be reproduced exactly. An approximation of the original image is enough for most purposes, as long as the error between the original and the compressed image is tolerable (Sayood, 2002).

BLOCK MATCHING METHOD

The LZ77 sliding window method for compression text can be applied to image as well. This study describes such an application for the lossless compression of images. A search buffer and look-ahead buffer can be made to slide in raster order along an image. The principle of image compression says that the neighbors of pixels P tend to have the same values as P or very similar values. Thus, if P is the leftmost pixel in the look-ahead buffer, there is a good chance that some of its neighbors on the left (i.e., in the search buffer) will have the same value as P . There is also, a chance that a string of neighbor pixels

in the search buffer will match P and the string of pixels that follow it in the look-ahead buffer. Experience also, suggests that in any nonrandom image there is a good chance of having identical strings of pixels in several different locations in the image.

Since, near-neighbors of a pixel are located also above and below it and not just on the left and right, it makes sense to use wide search and look-ahead buffers and to compare blocks of, say 4×4 pixels instead of individual pixels. This is the reason for the name block matching. When this method is used, the number of rows and columns of the image should be divisible by 4. If the image doesn't satisfy this, up to three artificial rows and/or columns should be added. Two reasonable edge conventions are, add columns of zero pixels on the left and rows of zero pixels on the top of the image. Duplicate the rightmost column and the bottom row as many times as needed (Salomon, 2000).

BLOCK TRUNCATION CODING METHOD

The basic BTC algorithm is a lossy fixed length compression method that uses a Q level quantizer to quantize a local region of the image. The quantizer levels are chosen such that a number of the moments of a local region in the image are preserved in the quantized output. In its simplest form, the objective of BTC is to preserve the sample mean and sample standard deviation of a grayscale image. Additional constraints can be added to preserve higher order moments. For this reason BTC is a block adaptive moment preserving quantizer (Delp *et al*., 2000).

Quantization is an important technique for data compression in general and for image compression in particular. Any quantization method should be based on a principle that determines what data items to quantize and by how much. The principle used by the Block Truncation Coding (BTC) method and its variants is to quantize pixels in an image, while preserving the first 2 or 3 statistical moments.

In the basic BTC method, the image is divided into blocks (normally 4×4 or 8×8 pixels each). Assuming that a block contains n pixels with intensities p1 through pn, the first 2 moments are the mean and variance, defined as:

$$\bar{p} = \frac{1}{n}\sum_{i=1}^{n} pi \tag{1}$$

and

$$\overline{p^2} = \frac{1}{n}\sum_{i=1}^{n} pi^2 \tag{2}$$

respectively. The standard deviation of the block is

$$\sigma = \sqrt{\overline{p^2} - \bar{p}} \tag{3}$$

The principle of the quantization is to values, a threshold pthr, a high value p^+ and a low vaue p^-. Each pixel is replaced by either p^+ or p^-, such that the first 2 moments of the new pixels (i.e., their mean and variance) will be identical to the original moments of the pixel block. The rule of quantization is that a pixel pi is quantized to p^+ if it is greater than the threshold and is quantized to p^- if it is less than the threshold (if pi equals the threshold, it can be quantized to either value). Thus,

$$pi \leftarrow \begin{cases} P^+, \text{if } P_i \geq P_{thr}, \\ P^-, \text{if } P_i < P_{thr}, \end{cases} \tag{4}$$

Intuitively, it is clear that the mean $\bar{p}$ is a good choice for the threshold. The high and low values can be calculated by writing equations that preserve the first two moments and solving them. We denote by n^+ the number of pixels in the current block that are greater than or equal to the threshold. Similarly, n^- stands for the number of pixels less than the threshold. The sum $n^+ + n^-$ equals, of course, the number of pixels n in the block. Once the mean $\bar{p}$ has been computed, both n^+ and n^- are easy to calculate. Preserving the first 2 moments is expressed by the Eq. 2.

$$\begin{aligned} & n^- \bar{p}\ n^- p^- - n^+ p^+, n\overline{p^2} \\ & = n^- (p^-)^2 - n^+ (p^+)^2 \end{aligned} \tag{5}$$

These are easy to solve even though they are nonlinear and the solutions are

$$p^- \bar{p} - \sigma\sqrt{\frac{n^+}{n^-}}.\ p^+ \bar{p} + \sigma\sqrt{\frac{n^-}{n^+}} \tag{6}$$

These solutions are generaly real numbers, but they have to be rounded to the nearest integer, which implise that the mean and variance of the quantized block may be somewhat different from the original ones. Notice that the solutions are located on the 2 sides of the mean $\bar{p}$ at distances that are proportional to the standard deviation σ of the pixel block.

The basic BTC method is simple and fast. Its main drawback is the way it loses image information, which is based on pixel intensities in each block and not on any properties of the human visual system. Because of this, images compressed under BTC tend to have a blocky character when decompressed and reconstructed. This led many researchers to develop enhanced and extended versions of the basic BTC (Salomon, 2000).

THE PROPOSED ALGORITHM

The proposed system is designed for obtaining the small area to store FP by using this new method to compress FP.

Compression methods: This suggests method depends on the shape and principle of block compression, this method includes many steps, which are:

Algorithm of 3×3 Block Method with Index:
Input: Fingerprint image (monochrom image).
Output: Text file (result.rlc).
Step1: Divide the image into blocks and the size of each block is 9 pixels (3×3) and each pixel has 1 byte.
Step 2: Start from left top of the image.
Step 3: Repeat
Take block (3×3) pixels from the image.
Change it from pixel form to the bits form (9 pixel-change to -9 bits)
Compare it with the list in the index (tb-table) as show in Table 1:
If this block not found in the index,
Then add it to the index with its new symbol.
And return the symbol of block instead of block itself.
If operation of matching is achieved with one of them,
Then return the symbol of block instead of block itself.
Put (-) sign between symbol and another to separate between them.
Save the output in the (temp.txt) file.
Else
Record the coordinates of the block.
Move to right bottom.
Until reaching the end of image.
Step 5: Apply the RLC (Run Length coding) algorithm for (temp.txt) file.
Put (-) sign between symbol and another to distinguish between them.
Save the new output in the (result.rlc) file.
Step 6: End.

When applaying this method to output of block algorithm file this will reduce the number of bytes or character in the file by storing the number of occurrences of character and the repeated character itself in the file, which represents a part of file (temp.txt).
Also, represent three values:

- Image file size (size of image before compression operation)
- Coded file size (size of temp.txt file)
- Compressed file size (size of result.rlc file)

The experimental result showed below, Fig. (1) represent fingerprint image before the compression operation, size of this image, size of RLC file (Coded) and Compressed file. Figure 2 represent the content of Coded file (tempt.txt) and Compressed file (result. rlc) after the compression operation.

Table 1: Index table for (3×3) block method

Pixs	ii	Code	Pixs	ii	Code
000000000	0	A1	111101110	494	A20
000000001	1	B1	111101111	495	B20
000000010	2	C1	111110000	496	C20
000000011	3	D1	111110001	497	D20
000000100	4	E1	111110010	498	E20
000000101	5	F1	111110011	499	F20
000000110	6	G1	111110100	500	G20
000000111	7	H1	111110101	501	H20
000001000	8	I1	111110110	502	I20
000001001	9	J1	111110111	503	J20
000001010	10	K1	111111000	504	K20
000001011	11	I1	111111001	505	L20
000001100	12	M1	111111010	506	M20
000001101	13	N1	111111011	507	N20
000011101	14	O1	111111100	508	O20
000001111	15	P1	111111101	509	P20
000010000	16	Q1	111111110	510	Q20
000010001	17	R1	111111111	511	R20
000010010	18	S1			

Fig. 1: Fingerprint image with (3×3) block method

Decompression methods: This method is also one of the suggested methods to decompress the FP image, this operation will be done by the following algorithm:

Input: Text file (result.rlc).
Output: Fingerprint image (monochrom image).
Step 1: From the text file of RLC result return back the occurrence of a data item:
For each single pair (nd), (d: data item, n: number of occurs), return occurs for this (d) item as its numbers.
Save this occurs in another text file of data (temp2.txt).
Step 2: 1- From the text file (temp2.txt), for each data item occurs (character) return the block itself (3×3 blocks) instead of character (symbol) depending on the index of block, which saved in this system in table.
2- Save these blocks in a new image file.
Step 3: End.

Fig. 2: Compression operation of (3×3) block method

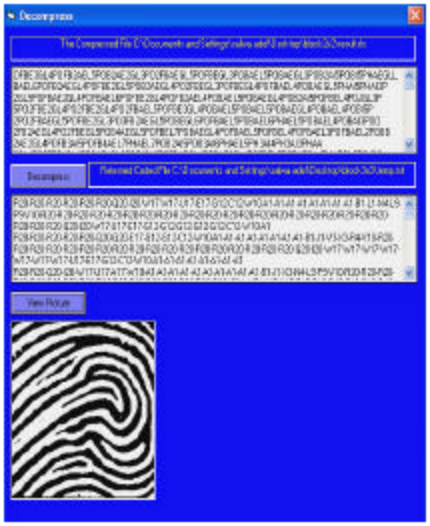

Fig. 3: Decompression operation of (3×3) block method

The experimental result showed in Fig. (3) represent reconstructed fingerprint image after the decompression operation, the content of Compressed file (result.rlc) and Decompressed file (tempt2. txt) after the compression operation.

EXPERIMENTS RESULTS

These calculations have three parts:

Compression Ratio (CR): The compression ratio is the degree of data reduction obtained as a result of the compression process, this calculation done between:

Table 2: Compression ratio result of (3×3) block method

Image	Image size	Coded file size	Compressed file size	Compression Ratio (CR)
F1	85254	12190	11859	7.1889
F2	119254	15938	7466	15.9729
F3	118454	15711	7053	16.7948
F4	97254	14793	451	215.6407
F5	93654	11531	5342	17.5316
F6	71798	8714	3230	22.2284
F7	70582	8666	3829	18.4335

Table 3: ERMS and SNR of (3×3) block method

Image	Image size	Compression Ratio (CR)	ERMS	SNR
F1	85254	7.1889	0.2262	97.4341
F2	119254	15.9729	1.0472	91.0301
F3	118454	16.7948	1.5503	89.2842
F4	97254	215.6407	10.7981	79.5673
F5	93654	17.5316	1.6023	87.3224
F6	71798	22.2284	1.9014	85.0841
F7	70582	18.4335	1.6980	86.5827

FP image (monochrom image) size and text file (result.rlc) size, as shown in Table 2. This calculation done for 7 images (f1, f2, f3, f4, f5, f6 and f7) bmp.

Cr = FP image size/result.rlc size

Root Mean Square Error (ERMS): The smaller the value of the ERMS metrics, the better the compressed image to represent the original image, this calculation done between:

New decompressed image (reconstructed image), (3×3 block index image). This calculation done for 7 image (f1-f7) bmp. As shown in Table 3:

$$ERMS = \sqrt{\frac{1}{N \times M} \sum_{r=0}^{n-1} \sum_{c=0}^{m-1} \left[\begin{matrix} new.decompressed image(r,c) \\ -original image(r,c) \end{matrix} \right]}$$

Signal to Noise Ratio (SNR): The larger number implies a better image, with signal to noise metrics (SNR), this metrics considers the decompressed image to be Signal and the error to be Noise, this calculation done between:

$$SNR = \sqrt{\frac{\sum_{r=0}^{n-1} \sum_{c=0}^{M-1} \left[decompress image(r,c) \right]^2}{\frac{1}{M \times N} \sum_{r=0}^{n-1} \sum_{c=0}^{M-1} \left[\begin{matrix} decompress.image(r,c) \\ -original image(r,c) \end{matrix} \right]^2}}$$

CONCLUSION

After implementing new methods and reviewing the practical images; we can introduce the following conclusions and comparisons:

These compression method (3×3) are lossless method, reduce the size of images without losing the important features of these images.

From Table 2 and 3 in Image Size fields and Compressed File Size fields; we can see reducing the size of these new method, (3×3) Block Methods.

Compression Ratio is height for this new method, but still the quality good because of this method are lossless method.

From practical results we can see the quality of images for this new (3×3) Block Method after compression and decompression operation still good, as shown in Fig. 1-3.

From Table 3, in ERMS fields and SNR fields; the values of ERMS are small and the values of SNR are so large that mean the better compressed method representing the original image.

REFERENCES

Cherrill, F.R., 1954. The finger print system at scotland yard. DCJS, central Library/Her Majesty's Stationery Office/University of England, London, pp: 13. www.criminaljustice.state.ny.us/ojis/history/appndx_4.htm.

Delp, E.J., M. Saenz and P. Salama, 2000. Block Truncation Coding (BTC), The Handbook of Image and Video Processing. In: Bovik, A.C. (Ed.). Academic Press, ftp://skynet.ecn.purdue.edu/pub/dist/delp/btc-chapter/btc-delp.pdf.

Sahni, S., B.C. Vemuri, F. Chen, C. Kapoor, C. Leonard and J. Fitzsimmons, 1997. State of the art lossless image compression algorithms. Cite Seerx, pp: 1. DOI: 10.1.1.27.2645-28 k, citeseer.ist.psu.edu/428215.html.

Salomon, D., 2000. Data Compression The Complete Reference. 2nd Edn. Springer-Verlag New York. pp: 345-350. ISBN: 0-387-95045-1, www.ecs.csun.edu/~dxs/DC2 advertis/DComp2Ad.html.

Sayood, K., 2002. Lossless Compression Handbook. Academic Press, pp: 207. ISBN: 10: 0126208611, ISBN-13: 978-0126208610, www.amazon.com/Lossless-Compression-Communications-Networking-Multimedia/dp/0126208611.

Xiong, X.Z., S.W. Cheng and J. Hua, 2003. Lossy-To Lossless Compression of Medical Volumetric Data Using Three Dimensional Integer Wavelet Transforms, IEEE Trans. Med. Imaging, 22 (3).