A Block Compression Method for Fingerprint Image Storing

Salwa A.Al-alagha*

Received on:25 /6 /2008
Accepted on:2 /7 /2009

Abstract

Storage of fingerprint image databases needs allocation of huge secondary storage devices. To reduce the increasing demand on storage space, efficient data compression techniques are badly needed. In addition to that, the exchange of fingerprint images between governmental agencies could be done fast. The compression algorithm must also preserve the original information in the original image. 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. The reduced file is called the compressed file and is used to reconstruct the image, resulting in the decompressed image. The original image file, before any compression is performed, is called the uncompressed image file. This paper is concerned with compression method of fingerprint image without losing the important features of these images (lossless method), by using (2 × 2), (3 × 3) Block Compression Method. This proposed method done by compressing the fingerprint (FP) image, and decompressing the image for the twice previous methods. And then compare the result which come from experimental results for (size of image, quality of image, thickness of curve), and find the relationship between them.

طريقة ضغط البلوك لخزن صورة بصمة الأصبع

الخلاصة

قاعدة البيانات الضرورية لخزن صورة بصمة الاصبع تحتاج الى مجموعة كبيرة من اجهزة الخزن الثانوية. ولتقليل الزيادة في فضاء الخزن, يتطلب تقنيات كفوءة جدا لضغط البيانات. بالاضافة لذلك, عملية تبادل صور بصمة الاصبع ما بين التعاملات الحكومية تجرى بسرعة هائلة و خوارزمية الضغط يجب ان ترجع المعلومات الاصلية في الصورة الرئيسية. ضغط الصورة هي تطبيق لضغط البيانات في الصور الرقمية. الغرض الرئيسي لهذه العملية هي تقليص تكرار بيانات الصورة لخزن ونقل البيانات بصورة كفوءة. تتضمن هذه العملية تقليل حجم البيانات في فايل الصورة المتضمن معلومات ضرورية. عملية تقليص الفايل تسمى ضغط الفايل, و يستخدم نفس هذا الفايل لاسترجاع الصورة الاصلية بعملية فك ضغط الصورة, فايل الصورة الرئيسي قبل اي عملية ضغط يسمى فايل الصورة غير المضغوط. يتناول هذا البحث

* Electrical and Electronic Department Engineering, University of Technology/Baghdad

https://doi.org/10.30684/etj.27.11.1
2412-0758/University of Technology-Iraq, Baghdad, Iraq

طريقة مقترحة لضغط صورة بصمة الاصبع بدون ضياع الخصائص المهمـــة لهـــذه الـــصورة (طريقة عدم ضياع البيانات), باستخدام طريقة الضغط Block Method (3*3) ,(2*2). فـــي الطريقة المقترحة هذه, يتم عمل ضغط لصورة بصمة الاصبع و فك الـــضغط فـــي الطـــريقتين السابقتين و من ثم مقارنة النتائج المستحصلة من الطريقتين من حيث حجـــم الـــصورة, ونـــوع الصورة, و عرض الخط وايجاد العلاقة بينهما.

Introduction:

Compressing an image is significantly different than compressing raw binary data. Of course, general purpose compression programs can be used to compress images, but the result is less than optimal. This is because images have certain statistical properties which can be exploited by encoders specifically designed for them. Also, some of the finer details in the image can be sacrificed for the sake of saving a little more bandwidth or storage space. This also means that lossy compression techniques can be used in this area. [7]

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. [3] 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. [5]

Block Matching Method:

The LZ77 sliding window method for compression text can be applied to image as well. This section 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. 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. [4]

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. [2]

Quantization is an important technique for data compression in general and for image compression in

particular. The principle used by the block truncation coding (BTC) method and its variants is to quantize pixels in an image while preserving the first two or three statistical moments (mean, variance, and standard deviation).

In the basic BTC method, the image is divided into blocks (normally 4 × 4 or 8 × 8 pixels each).

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. [4]

Run-Length Coding Method:

The idea behind this approach to perform data compression is; if the data item (d) occurs (n) consecutive times in the input stream, replace the (n) occurrences with single pair (nd), (n) consecutive occurrence of a data item is called a run length of (n), and this approach of data compression is called run length coding or (RLC). [6]

Fingerprint Classification:

Henry's classification, the real significance of fingerprint patterns. The well-known Henry system consists of four major steps: [1]

- Arch: as shown in Fig. (1-a).
- Loop: as shown in Fig. (1-b).
- Whorl: as shown in Fig. (1-c).
- Compounds: as shown in Fig. (1-d).

The Proposed Block Method:

Compressing fingerprint images will reduce size of data files, while returning necessary information (without loss of any important features).

Compression of fingerprint will divided into two methods:

2×2 Block Compression Method with Index:

Our suggest method depends on the specificity of the shape and the principle of block compression, this method includes many steps, which are:

1. Algorithm of 2*2 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 4 pixels (2*2) and each pixel has 1 byte.

Step2: Prepare the index, which contains 2^4 blocks. This means 16 blocks to store as database each block with its symbol (symbol from A to P), as shown below in Fig. (2).

Step3: Start from left top of the image.

Step4: Repeat

Take block (2*2) pixels from the image.

Compare it with the blocks in the index.

If operation of matching is achieved with one of them,

Then return the character of block instead of block itself

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.

Step5: Apply the RLC (Run Length coding) algorithm for (temp.txt) file.

Save the new output in the (result.rlc) file.

Step6: End.

When applying 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 occurence of character and the repeated character itself in the file, which represents a part of file (temp.txt).

2. Represent these three values:

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

The experimental result, Fig. (3) represent fingerprint image before the compression operation, size of this image, size of RLC file (Coded), and compressed file. Fig. (4) Represent the content of Coded file (tempt.txt), and compressed file (result.rlc) after the compression operation.

3×3 Block Compression Method with Index:

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

1. 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.

Step2: Start from left top of the image.

Step3: 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 Tab.(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.

Step5: 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.

Step6: 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 occurence of character and the repeated character itself in the file, which represents a part of file (temp.txt).

2. Also represent three values:

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

The experimental result, Fig. (6) represent fingerprint image before the compression operation, size of this image, size of RLC file (Coded), and compressed file. Fig. (7) Represent the content of Coded file (tempt.txt), and compressed file (result.rlc) after the compression operation.

Decompression Methods:

The decompression operation will be done to return the necessary information of fingerprint image without losing the important feature.

Decompression operation method is determined by the compression operation method, so the decompression of fingerprint, can be divided into three methods:

Decompression of 2*2 Block Index:

This method is one of the suggested methods to decompress the FP image; the decompression operation will be done by returning the block itself instead of character depending on the index of block method in Fig. (5), this suggested method includes many steps, which are:

Algorithm of 2*2 Block Index Decompression:

Input: Text file (result. rlc).
Output: Fingerprint image (monochrom image).
Step1: From the text file of RLC result return back the occurrence of a data
item:
1- For each single pair (nd), [d: data item, n: number of occurs], return occurs for this (d) item as its numbers.
2- Save this occurs in another text file of data (temp2.txt).
Step2: 1- From the text file (temp2.txt), for each data item occurs (character)
return the block itself (2*2 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.
Step3: End.

The experimental result, Fig. (5) represent reconstructed fingerprint image after the decompression operation, the content of compressed file (result.rlc), and decompressed file (tempt2.txt) after the compression operation.

Decompression of 3*3 Block Index:

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

Algorithm of 3*3 Block Index Decompression:

Input: Text file (result.rlc).
Output: Fingerprint image (monochrom image).
Step1: From the text file of RLC result return back the occurrence of a data
item:
1- For each single pair (nd), [d: data item, n: number of occurs], return occurs for this (d) item as its numbers.
2- Save this occurs in another text file of data (temp2.txt).
Step2: 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.

Step3: End.

The experimental result, Fig. (8) represent reconstructed fingerprint image after the decompression operation, the content of compressed file (result.rlc), and decompressed file (tempt2.txt) after the compression operation.

Comparison (Calculation) Operation:

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:

1- FP image (monochrom image) size and text file (result.rlc) size, as shown in Tab.(2) and Tab.(3):

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:

1- New decompressed image (reconstructed image), (2*2 block index image). This calculation done for 7 image (f1, f2, f3, f4, f5, f6, and f7).bmp. As shown in Tab. (4):

ERMS =

$$\sqrt{\frac{1}{N*M}\sum_{r=0}^{n-1}\sum_{c=0}^{m-1}\begin{bmatrix} new\ .decompress\quad ed\ . \\ image\ (r,c) \\ -\ original\quad .image\ (r,c) \end{bmatrix}}$$

2- New decompressed image (reconstructed image), (3*3 block index image). This calculation done for 7 image (f1, f2, f3, f4, f5, f6, and f7).bmp. As shown in Tab. (5):

ERMS =

$$\sqrt{\frac{1}{N*M}\sum_{r=0}^{n-1}\sum_{c=0}^{m-1}\begin{bmatrix} new\ .decompress\quad ed\ . \\ image\ (r,c) \\ -\ original\quad .image\ (r,c) \end{bmatrix}}$$

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:

1- New decompressed image (reconstructed image), (2*2 block index image). This calculation done for 7 image (f1, f2, f3, f4, f5, f6, and f7).bmp. As shown in Tab. (4):

SNRRMS=

$$\sqrt{\frac{\sum_{r=0}^{N-1}\sum_{c=0}^{M-1}[decompressimage(r,c)]^2}{\frac{1}{M*N}\sum_{r=0}^{N-1}\sum_{c=0}^{M-1}[decompressimage(r,c)-originalimage(r,c)]^2}}$$

2- New decompressed image (reconstructed image), (3*3 block index image). This calculation done for 7 image (f1, f2, f3, f4, f5, f6, and f7).bmp. As shown in Tab. (5):

SNRRMS =

$$\sqrt{\frac{\sum_{r=0}^{N-1}\sum_{c=0}^{M-1}[decompressimage(r,c)]^2}{\frac{1}{M*N}\sum_{r=0}^{N-1}\sum_{c=0}^{M-1}[decompressimage(r,c)-originalimage(r,c)]^2}}$$

Conclusions and Comparisons:

Fingerprint compression method's is very important and necessary to reduce the size of image data files, and returning necessary

information without losing important features. After implementing new methods and reviewing the practical images; we can introduce the following conclusions and comparisons:

1. These compression methods (2*2), (3*3) are lossless methods, reduce the size of images without losing the important features of these images.
2. From Tab. (2), Tab. (3) in *Image Size* fields, and *Compressed File Size* fields; we can see reducing the size of these new methods (2*2), (3*3) Block Methods. But the *Compressed File Size* of (2*2) method less than the *Compressed File Size* of (3*3) method because of the (-) sign which separate between the symbols, but make the file larger.
3. Compression Ratio is height for these new methods, but still the quality good because of these methods are lossless methods, as shown in table(2) , table(3), figure(5), and figure(8).
4. From practical results we can see the quality of images for these new (2*2), (3*3) Block Methods after compression and decompression operation still good, as shown in figures (3), (5), and figures (6), (8).
5. From Tab. (4), Tab. (5), 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.
6. Depending on the thickness of the curve of the fingerprint image; we can extract that images (f1, f2, f3), compressed by (2*2) block method better than another method because these have thick curves. But images (f4, f5, f6, f7), compressed by (3*3) block method, as shown in table (2), and (3), depending on compression ratio value.

References:

[1] Cherrill F. R, "THE FINGER PRINT SYSTEM AT SCOTLAND YARD", DCJS, Central Library/Her Majesty's Stationery Office/University of England, London, p. 13, www.criminaljustice.state.ny.us/ojis/history/appndx_4.htm, 1954.

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

[3] Sahni S., Vemuri B.C., Chen F., Kapoor C., Leonard C., Fitzsimmons J., "State of The Art Lossless Image Compression Algorithms", CiteSeerx, p.1, DOI=10.1.1.27.2645 - 28k, citeseer.ist.psu.edu/428215.html, 30 Oct 1997.

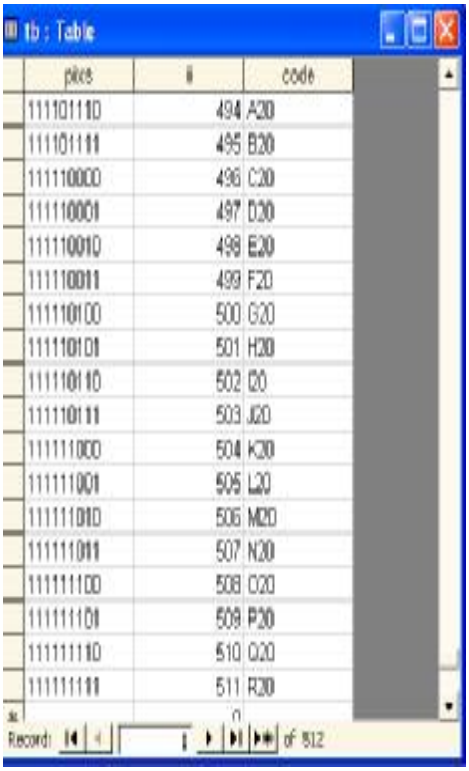
[4] Salomon D., "Data Compression The Complete Reference", Springer-Verlag New York, second edition, p. 345-350, ISBN 0-387-95045-1, www.ecs.csun.edu/~dxs/DC2advertis/DComp2Ad.html, 2000.

[5] Sayood K., "Lossless Compression Handbook", Academic Press, p. 207, ISBN-10**:** 0126208611, ISBN-13: 978-0126208610, www.amazon.com/Lossless-Compression-Communications-Networking-Multimedia/dp/0126208611,August 15 2002.

[6] Umbaugh S. E., "Computer Vision and Image Processing", Prentice Hall PRT, ISBN: 0 13 264599 8, http://www.imageprocessingplace.com/root_ files_V3/publications.htm, 1998.

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

Table (1) Index Table for (3*3) Block Method



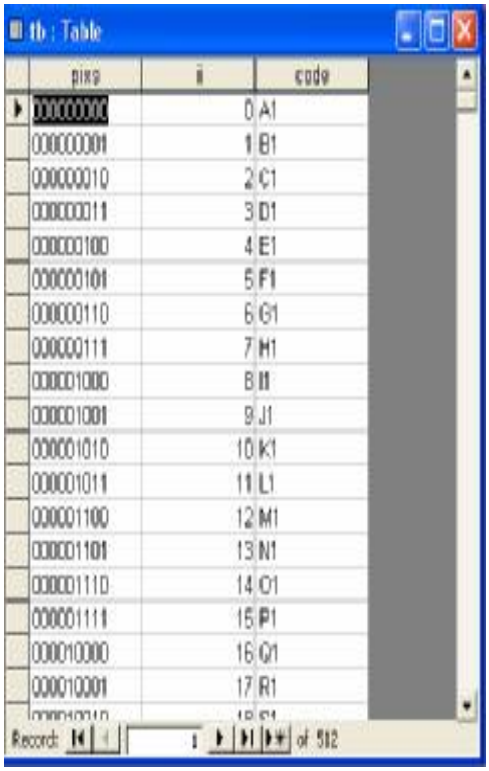


Table (2) Compression Ratio Result of 2*2 Block Method

Image	Image size	Coded file size	Compressed file size	Compression Ratio (Cr)
F1	85254	7242	5250	16.2388
F2	119254	10100	3656	32.6187
F3	118454	10000	4561	25.9710
F4	97254	8262	488	199.2909
F5	93654	8000	5826	16.0751
F6	71798	6080	4055	17.7060
F7	70582	6004	4145	17.0282

Table (3) 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 (4) ERMS and SNR of 2*2 Block Method

Image	Image size	Compression Ratio (Cr)	ERMS	SNR
F1	85254	16.2388	1.4732	88.2557
F2	119254	32.6187	2.1662	84.2762
F3	118454	25.9710	2.0608	84.0819
F4	97254	199.2909	8.6809	80.7445
F5	93654	16.0751	1.1853	91.1333
F6	71798	17.7060	1.6201	86.0441
F7	70582	17.0282	1.5433	86.9576

Table (5) 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

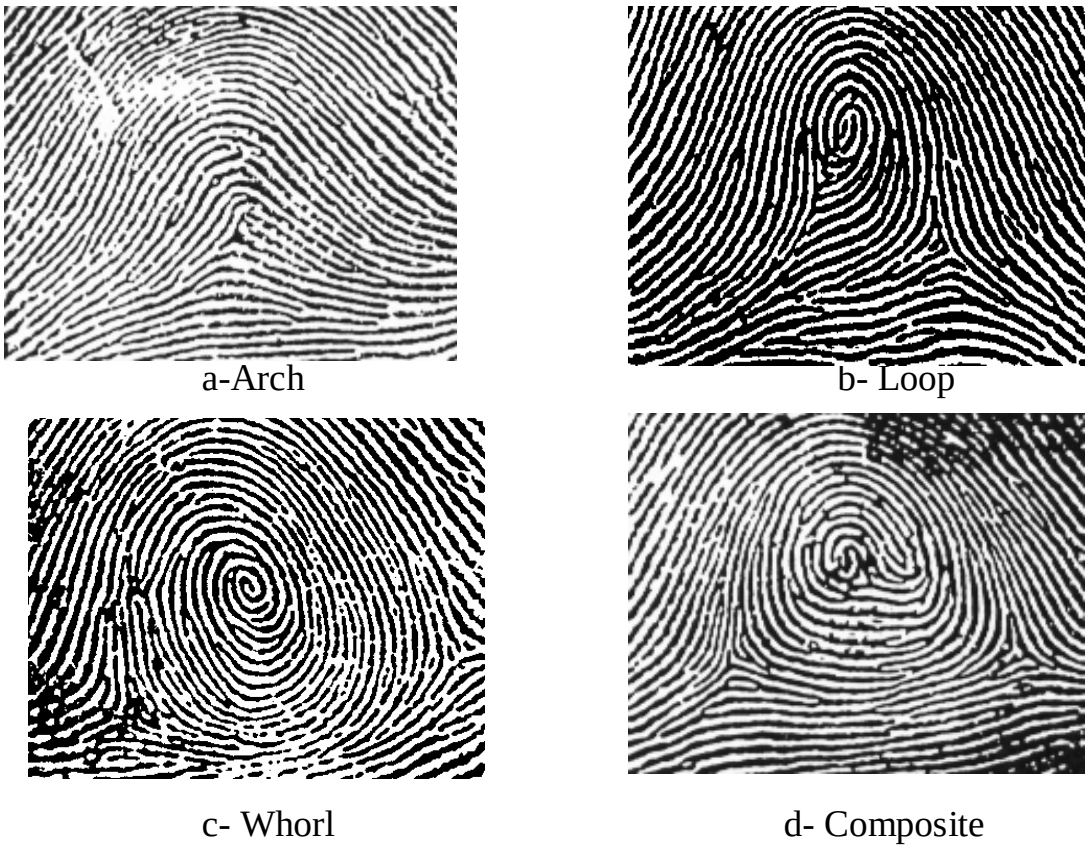


Figure (1) Fingerprint Classification

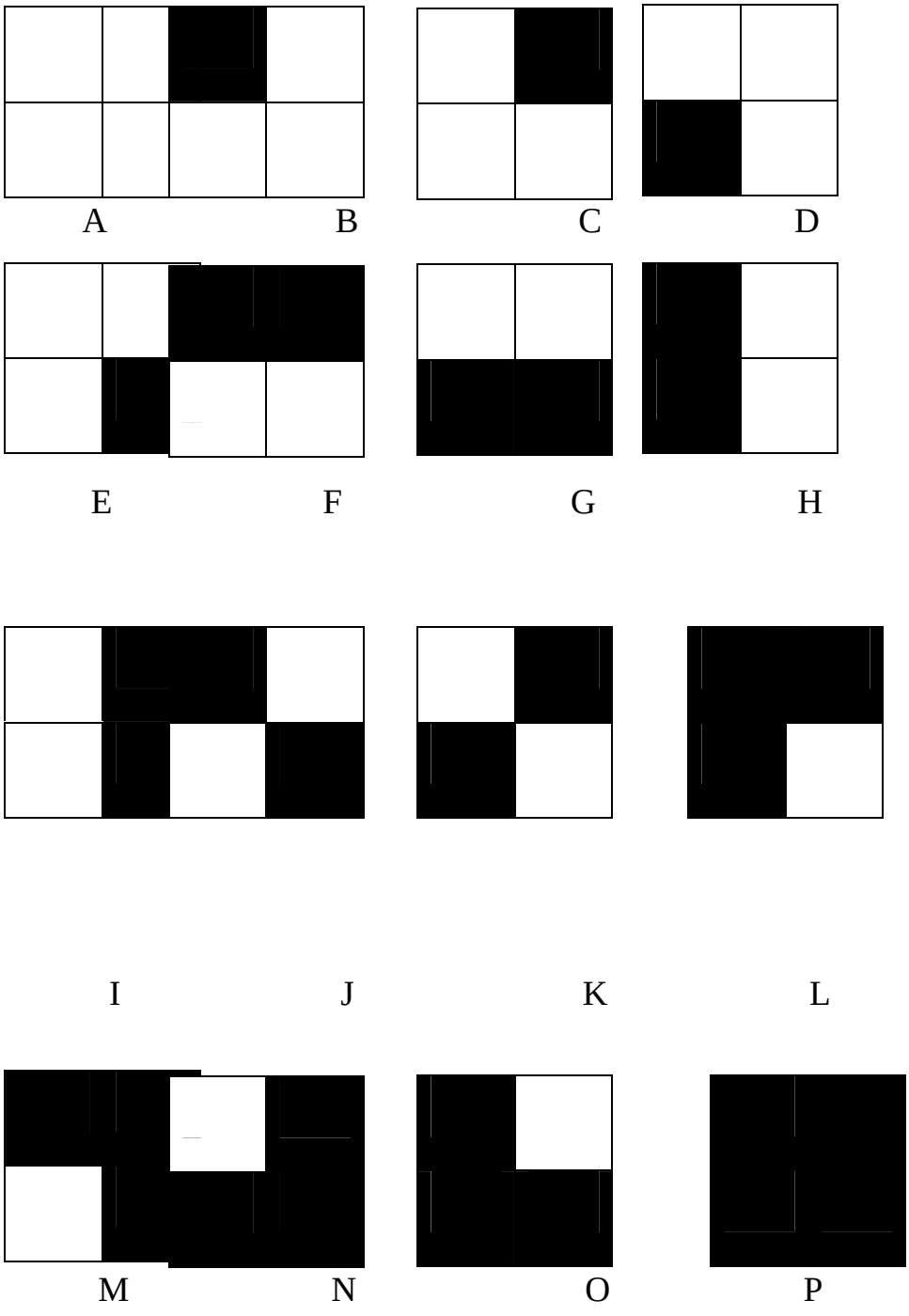


Figure (2) The index of block Method

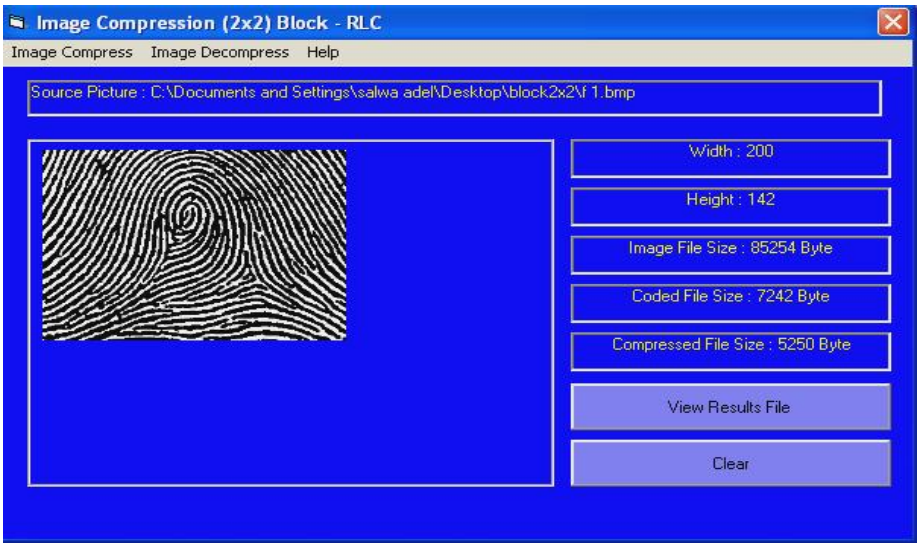


Figure (3) Fingerprint Image with (2*2) Block Method

Figure (4) Compression Operation of (2*2) Block Method

Figure (5) Decompression Operation of (2*2) Block Method

Figure (6) Fingerprint Image with (3*3) Block Method

Figure (7) Compression Operation of (3*3) Block Method

Figure (8) Decompression Operation of (3*3) Block Method