Application of Matrix in Image Compression

Vitra Chandra - 13514043
Program Studi Teknik Informatika
Sekolah Teknik Elektro dan Informatika
Institut Teknologi Bandung, Jl. Ganesha 10 Bandung 40132, Indonesia
13514043@std.stei.itb.ac.id

***Abstract*—With today's rapid demand of large data storage, image compressing is important today to decrease data consumption. This paper will discuss about how matrix is used in image compression, particularly JPEG image compression, and the analysis of JPEG formats.**

***Keywords*—image compression, linear algebra, matrix, linear transformation, jpeg technique**

I. Introduction

In 1989, Joint Photographic Experts Group, known as JPEG, discuss and standard image compression method to minimize data usage in image storing because most computers that day weren't capable of handling image files, which are quite large. Hence, they form a universal standard to ease data handling since different these data needed to be interchangeable.

In 1991, the chairman of JPEG, Gregory Wallace, published a paper outlining their compression standard. This compression standard was then adopted in 1994, and became so widespread that it is even used today.

The reason for its success is simple. This compression standard by JPEG, allows large data to be compressed down to a much smaller size, while maintaining its quality.

II. Matrix

Linear algebra is a branch of mathematics that studies linear mapping, and vectors. The scope of study of linear algebra are matrix theory, linear transformation, and vector spaces.

A. Matrix Theory

Matrix is a rectangular array of numbers arranged in rows and columns.

$$A = \begin{pmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \cdots & a_{mn} \end{pmatrix} \qquad (2.1)$$

Equation (2.1) is a notation for matrix A. Each a_{ij} in matrix A are called elements or entries located at row *i* and column *j*. The size of a matrix are the number of rows times the number of columns. Therefore, matrix A in (2.1) is a matrix with a size of $m \times n$. Matrix with a single row or single column is called a row vector or column vector respectively.

There are a few basic operation in matrices, such as addition, scalar multiplication, matrix multiplication, and transpose.

In addition, take C = A + B, whereas A is an $m \times n$ matrix and B is an $m \times n$ matrix. Therefore, C will be an $m\times n$ matrix with each $C_{ij} = A_{ij} + B_{ij}$. Note that in matrix addition, the size of A and B must be the same.

$$C = A + B = \begin{pmatrix} a & b \\ c & d \end{pmatrix} + \begin{pmatrix} e & f \\ g & h \end{pmatrix} = \begin{pmatrix} a+e & b+f \\ c+g & d+h \end{pmatrix} \qquad (2.2)$$

In scalar multiplication, take matrix A and a float number *k*, then X = *k*A, where X is a matrix with the same size as matrix A and $X_{ij} = k \times A_{ij}$.

$$X = k \times A = k\begin{pmatrix} p & q \\ r & s \end{pmatrix} = \begin{pmatrix} kp & kq \\ kr & ks \end{pmatrix} \qquad (2.3)$$

In matrix multiplication, take matrix $A_{m\times n}$ and matrix $B_{p\times q}$, then $C = A \times B$ may be done if $n = p$ with C the size of $m \times q$, and $C = B \times A$ may be done if $q = m$ with C the size of $p \times n$.

$$C = A_{2x3} \times B_{3x2} = \begin{pmatrix} a & b & c \\ d & e & f \end{pmatrix}\begin{pmatrix} p & s \\ q & t \\ r & u \end{pmatrix}$$

$$C = \begin{pmatrix} ap+bq+cr & as+bt+cu \\ dp+eq+fr & ds+et+fu \end{pmatrix} \qquad (2.4)$$

Finally, a transposed matrix is a matrix in which each row are swapped into a column and vice versa. For example, let A be a matrix of size $m \times n$ from (2.1), then $B = A^T$ will be of size $n \times m$.

$$B = \begin{pmatrix} a_{11} & a_{21} & \cdots & a_{m1} \\ a_{12} & a_{22} & \cdots & a_{m2} \\ \vdots & \vdots & \ddots & \vdots \\ a_{1n} & a_{2n} & \cdots & a_{nm} \end{pmatrix} \qquad (2.5)$$

B. Linear Transformation

Let V and W be vector spaces, then T: V → W is called a linear transformation. For each vectors a, b ∈ V and $k \in$ R, the following conditions are satisfied:

i. $T(a+b) = T(a) + T(b)$
ii. $T(ka) = kT(a)$

In linear algebra, linear transformation can be represented by a matrix. For example, the linear transformation of rotation is written as:

$$T = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.5)$$

$$T = \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} \quad (2.6)$$

Equation (2.5) is the linear transformation of rotation by θ angle clockwise and (2.6) is of rotation θ angle counter-clockwise.

III. Image Compression

Image compression is the process of minimizing down the size of an image with minimum damage to the quality of the image. The minimized image allows for easier access, storage, and transport.

Image compression technique may be lossy or lossless. Lossless image compression compresses an image without introducing errors, thus retaining the image information. Lossless compression is generally used in compresing text files and program files because a single error may prove fatal in a program, On the other hand, lossy image compression compacts an image while losing information during the compression. Though it may seem true, lossless compression is not always suitable for every image compression. Lossy compression results in better compression due to its nature of “losing” useless information. This compression method is generally used in JPEG compression because the discarded information are mostly imperceptible to human eyes, thus retaining the quality visually.

A. Matrix as an Image

An image can be represented by using matrices. For example, a Felix the cat image as follows.

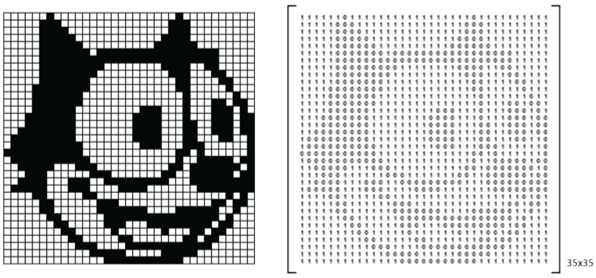

Fig 3.1 The image of Felix the cat and the matrix corresponding

Figure 3.1 shows an example of an image represented by a matrix. Each element in the matrix corresponds to each pixel in the image, a 0 indicating black and 1 indicating white. This type of image, that only uses two colors are called *boolean images* or *binary images.*

A grayscale image, may also be represented with a matrix, with each element corresponding with the image shows the intensity of the pixel. The data in each pixel usually uses an integer to represent the intensity, with 0 as black and 255 as white, allowing one to use 256 different shades of gray.

On the other hand, colored images, also known as true color, can be represented with three or more matrices, depending on its coloring system. A few coloring system are known to computers today, with RGB and CMYK the most generally used.

An RGB image are represented with three matrices. Each matrix represent one shades of color, with red, green, and blue respectively. Similar to a grayscale image matrix, each RGB image matrix element are represented with an integer number from 0 to 255. To construct the image, the three matrix will then overlap each other to represent a color.

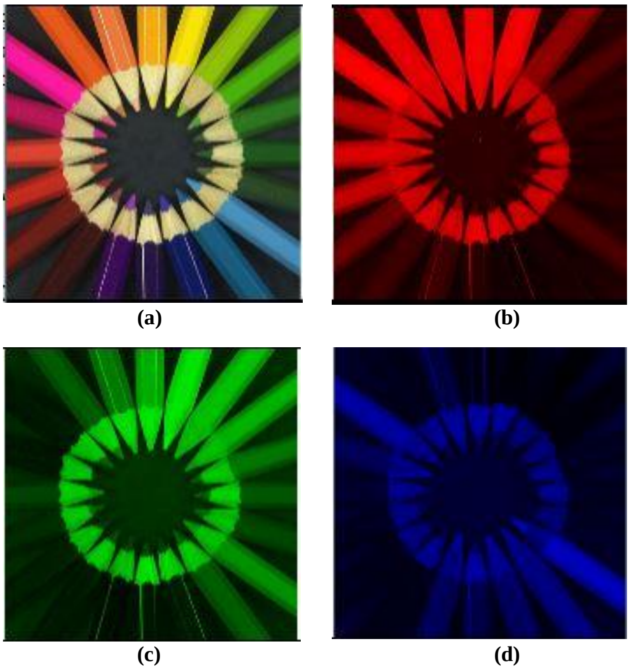

Fig 3.2 (a) Original picture, (b) red, (c) green, (d) blue components

Therefore, in RGB system, a single pixel can be represented with $256^3 = 16777216$ colors.

B. Image Processing

After an image are represented with matrices, it is possible to operate the image into several transformation. For example, consider the following figure.

(a) (b) (c)

Fig 3.3 (a) Original image, (b) transposed image, (c) reflected image

In figure 2.3, a binary image of Felix the cat (a) can be transposed into (b). While the image (c) is the reflected image of (a). Let C be the matrix of image (c) and A be the matrix of image (a), thus $C_{ij} = A_{i,35-j+1}$, allowing the image to be reflected by the vertical axis.

Another example of image processing is using the transition effect on presentations and slide shows. For example, consider two image of the same size, with the matrix A and Z respectively, and a real number t in an interval [0, 1], and the matrix M(t) as the matrix of the moment of transition. Thus, by using scalar multiplication and addition of matrices, the matrix

$$M(t) = (1 - t)A + tZ. \tag{3.1}$$

At $t = 0$, the matrix $M(0) = A$, and at $t = 1$, the matrix $M(1) = Z$. In the interval of t = 0 to 1, the matrix M(t) are between A and Z.

C. JPEG Compression Method

There are a few methods in image compression. For example, Singular Value Decomposition (SVD), Wavelet, Fractal, and JPEG. JPEG compression method exploits the inability of the human eye to discern minimal changes in an image. JPEG process discards imperceptible changes in color and brightness to human eye. This method makes JPEG compression ideal for photograph compression.

In JPEG compression, there are four main steps: the Discrete Cosine Transform (DCT), quantization, reordering, and Huffman coding. DCT changes the value of each pixel in a matrix into a matrix containing the change of color in an image. Then, the quantization discards the imperceptible details or changes. After reordering the quantization matrix, the matrix is encoded through Huffman coding to compress the file further.

Discrete Cosine Transform

The main process of the JPEG compression technique is the DCT. The DCT changes the image matrix into one more suitable for compression, thus easing out the calculation and processes. A DCT matrix changes the image matrix from spatial domain into a frequency domain represented by a linear combination of sinusoidal functions of frequencies in two dimensions. Thus, the matrix only records the changes in color intensity in the image, not the value of the intensity itself, with higher number indicating large changes in intensity. Below is a two-dimensional DCT in JPEG compression:

$$C_{uv} = a_u a_v \sum_{m=0}^{M=1} \sum_{n=0}^{N=1} A_{mn} \cos\frac{\pi(2m+1)u}{2M} \cos\frac{\pi(2n+1)v}{2N} \tag{3.2}$$

where $0 \le u \le M - 1$ and $0 \le v \le N - 1$ and

$$a_u = \begin{cases} \frac{1}{\sqrt{M}}, & if\ u = 0 \\ \sqrt{\frac{2}{M}}, & if\ 1 \le u \le M - 1 \end{cases} \tag{3.3}$$

$$a_v = \begin{cases} \frac{1}{\sqrt{M}}, & if\ v = 0 \\ \sqrt{\frac{2}{M}}, & if\ 1 \le v \le M - 1 \end{cases} \tag{3.4}$$

These equations produce DCT matrix C where the actual DCT of the image A is:

$$DCT = CAC^T \tag{3.5}$$

Performing the calculation above will require quite a bit of computing power, thus JPEG technique divides the image into 8 × 8 blocks, in which the blocks then are modified using DCT. This requires much less computing power. DCT also tends to concentrate the intensity changes in the upper-left corner of the matrix, which will be vital in quantization.

Quantization

Quantization is the real process of the compression. By applying quantization matrix to the DCT, the JPEG technique discards the excess information in the bottom-right corner of the DCT matrix. Then, each element in the remaining transformed matrix is divided by the corresponding element in the quantization matrix.

$$R = round\left(\frac{D_{ij}}{Q_{ij}}\right) \tag{3.3}$$

A commonly used quantization matrix are as follows:

$$Q = \begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 55 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix} \tag{3.4}$$

The quantized matrix (the DCT matrix divided by the quantization matrix using (3.3)) R will be considered a sparse matrix due to its abundance of zero elements in the matrix.

Reordering

After quantizing the DCT matrix into a quantized matrix R, the matrix R can now be reordered into a 64 × 1 vector V. Due to the sparseness of matrix R, the vector V will have a string of zeros that may be eliminated. The reordering sequence are shown in the figure below.

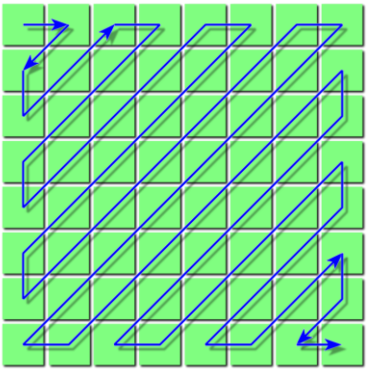

Fig 3.4 Zigzag reordering sequence

Huffman Coding

The last process in JPEG compression method is Huffman coding, after the DCT matrix has been quantized and reordered. Huffman coding allows redundancies in the symbol to be compressed because it uses the frequencies in the symbol and not the symbol itself. Moreover, Huffman coding is uniquely decodable that the code generated can only represent the original input when decoded. Huffman coding reduces the file size and then store the image.

Completion of the JPEG Compression

The JPEG compression is now complete. However, the above stated that the image is divided into 8 × 8 blocks. In reality, the image is divided into 8 × 8 blocks, in which, the blocks will be divided again into 8 × 8 blocks until each pixel can be represented. The DCT is applied to every block into a DCT matrix, then quantized and reordered, with into vectors separated by a non-zero coefficient to determine each block. After which, the Huffman coding is applied.

Accessing the Image

When accessing the image, the image is reconstructed through Huffman decoding, reverse ordering from vector to matrix, dequantization, and reverse DCT. Although the reconstructed image may differ from the original (due to loss of information in the compression), the image quality tend to be the same as the original. The full flow of JPEG compression can be seen in the figure below.

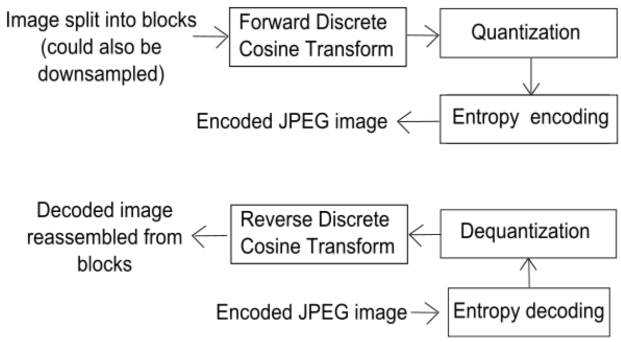

Fig 3.5 Flow of JPEG compression and accessing the image

IV. Analysis and Discussion

Block size

The reason why DCT divided the matrix into 8 × 8 blocks is because when dividing the matrix into large sizes, the frequencies of the finer color differences will be insignificant, thus making a compression resulting in a worse quality. However, this does not mean that 4 × 4 or 2 × 2 will be better because the compression performed for smaller block size will take longer to perform. Hence, the 8 × 8 block size is the chosen way to divide the image.

Although 8 × 8 size is the best common method of dividing, it is not always the best method. A different block size may be more suited for different images.

Usage

JPEG compression is more commonly used for images that is viewed online. A single image file can be arduous to load if the image size in a web page is too huge. In addition, the limited resolution in monitors and human eyes allows the imperfections to not be noticeable.

However, JPEG compression is not suitable for image with crisp, sharp lines, as it will ruin the sharpness. Using lossless compression is better for this kind of image.

Advantages

- JPEG image is extremely portable
- JPEG image is compatible with almost every image processing application and hardware devices.
- JPEG image has been used since long time ago
- JPEG image size can be reduced significantly and is very suitable for transferring images.
- JPEG image can be compressed down to 5% of its original size.

Disadvantages

- JPEG compression is lossy.
- JPEG image quality is being reduced each time it is compressed.
- JPEG compression is not suitable for images with crisp sharp lines.
- JPEG image cannot handle animated graphic images.
- JPEG image does not support layered images for graphic designers.

V. Conclusion

JPEG compression is one of the first acknowledge image compression method. This method is ideal for storing images that does not heavily rely on its precision and unimportant information, and not recommended for use in medical sector and/or technical drawings. This type of compression is considered lossy compression, thus suitable for photographs.

VI. ACKNOWLEDGMENT

The author would like to express his gratitude to God Almighty for his guidance, to his parents for their love and education, and to the Mr. Rinaldi Munir for his teachings of Discrete Mathematics to complete this paper.

REFERENCES

[1] H, Anton. *Elementary Linear Algebra, 10th edition.* John Wiley and Sons. 2010.
[2] S, Franco. T, Prince. I, Salva. C, Windolf. *Mathematics behind Image Compression.* Journal of Student Research. 2014.
[3] S, Ian. *JPEG Image Compression*. 14 Dec. 2009.
[4] S, Venna. J, Krishna. *Image Compression and Linear Algebra.* 15 Nov. 2013.
[5] A, Nectoux. *Matrices and Digital Images.* Retrieved on 14 December 2015. From Klein Project Blog: http://blog.kleinproject.org/?p=588.
[6] M, Marcus. *JPEG Image Compression.* 1 Jun. 2014.

PERNYATAAN

Dengan ini saya menyatakan bahwa makalah yang saya tulis ini adalah tulisan saya sendiri, bukan saduran, atau terjemahan dari makalah orang lain, dan bukan plagiasi.

Bandung, 15 December 2015

Vitra Chandra
13514043