Comparison Of Lossless Image Compression Methods By Using Lossy Image Compression Method

Ibrahim Emiroglu, Armagan Elibol, Cuneyt Ertekin, Yildiray Koyuncu

Mathematical Engineering Department
Yildiz Technical University, Turkey
{emir,aelibol}@yildiz.edu.tr
{ertekinc,yildiray_koyuncu}@yahoo.com

Abstract

The rapid growth of digital imaging applications, including desktop publishing, multimedia, teleconferencing and high definition television (HDTV) has increased the need for effective and standardized image compression techniques [1]. In this study, we aimed to compare three well-known lossless compression methods (Huffman, Run Length and Arithmetic encoder) by using lossy image compression method. We have used Discrete Cosine Transformation (DCT)[1,2] based technique to lossy compression. We have observed and compared the results on grey-scale fingerprint images. During the study, fingerprint pictures, which are one pixel one byte and 512*512 sized, are used [5]. Lossy compression techniques aim's is to reduce the amount of data needed to describe the image without sacrificing too much image quality. The image reconstructed by lossy compresion is not numerically identical with the original image [6,7,8,9].

1. Introduction

The rapid growth of digital imaging applications, including desktop publishing, multimedia, teleconferencing and high definition television (HDTV) has increased the need for effective and standardized image compression techniques [1]. The fingerprint image which is shown in figure1, has 512*512 pixels and needs 512*512 bytes place for 1 pixel 1 byte image, this means 262144 bytes. If we think that the images are coloured, we need 786432 bytes. These numbers make it essential to do image compression.

Fig.1

Image data compression aims at taking advantage of this redundancy to reduce the number of bits required to represent an image [2]. In this study, we have used transform coding that is one of the lossy image data compression methods, by using three different types of entropy encoders. Lossy compression techniques aim's is to reduce the amount of data needed to describe the image without sacrificing too much image quality. The image reconstructed by lossy compression is not numerically identical with the original image [6]. The most used lossy compression technique is transform coding [6]. Transform coding steps are shown in figure 2. In fig. 2, all the steps are same apart from entropy encoders and decoders. Because, we want to observe their compression performances

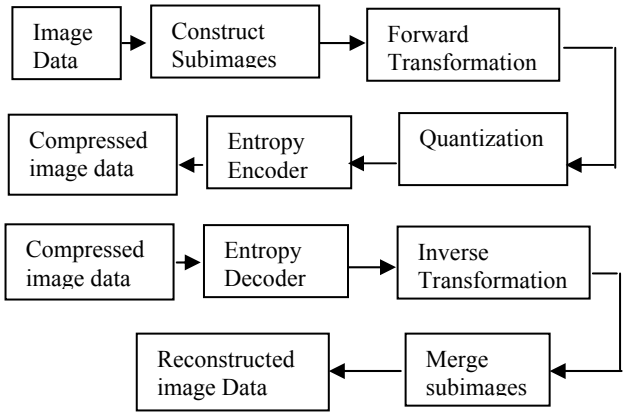


Fig.2.

2. Lossy Compression Steps

2.1. Transformation

The aim of the forward transformation step is to decrease correlation the pixels of image or to convert original image data to as possible as the smallest transform coefficient. Most transform coding systems use Discrete Cosine Transform for transformation steps in Fig 2. [1, 2] In this study we were used Discrete Cosine Transform (DCT). Two-dimensional DCT is following as below For nxn matrix s

$$C(u,v) = \alpha(u)\alpha(v)\sum_{y=0}^{n-1}\sum_{x=0}^{n-1} f(x,y)\cos\left[\frac{(2x+1)u\pi}{2n}\right]\cos\left[\frac{(2y+1)v\pi}{2n}\right]$$

for u, v=0,1,…,n-1

where $\alpha(u) = \begin{cases} \frac{1}{\sqrt{n}} & for\ u = 0 \\ \sqrt{\frac{2}{n}} & for\ u = 1,2,...n-1 \end{cases}$

and inverse DCT

$$f(x,y)=\sum_{u=0}^{n-1}\sum_{v=0}^{n-1}\alpha(u)\alpha(v)C(u,v)\cos\left[\frac{(2x+1)u\pi}{2n}\right]\cos\left[\frac{(2y+1)v\pi}{2n}\right]$$

for x, y=0,1,2,……,n-1

Two-dimensional DCT can be computed by one-dimensional DCT. First, one dimension DCT of image rows is taken and then one dimensional DCT is implemented to the columns over the new values. This gives the same result with two-dimensional DCT, and also these processes are same for the invers DCT.

2.2. Quantization

A quantization is the process of reducing the number of bits needed to store the transformed coefficients by reducing the precision of those values.
Quantization is the place where there is loss in lossy compression; because loss becomes when transform coefficients are divided by quantization coefficients. If quantization step is omitted, the compression will be lossless compression instead of lossy.
In this study, we have used 8*8 quantization matrix.

$$q(u,v)=\begin{bmatrix}1&1&2&4&8&16&32&64\\1&1&2&4&8&16&32&64\\2&2&2&4&8&16&32&64\\4&4&4&4&8&16&32&64\\8&8&8&8&8&16&32&64\\16&16&16&16&16&16&32&64\\32&32&32&32&32&32&32&64\\64&64&64&64&64&64&64&64\end{bmatrix}$$

Fig.5 Quantization Base Matrix

Quantized coefficients can be calculated by employing the formula given below:

$$\overline{C}(u,v)=round\left(\frac{C(u,v)}{q(u,v)}\right)$$

Where C (u,v) input forward transformed image data

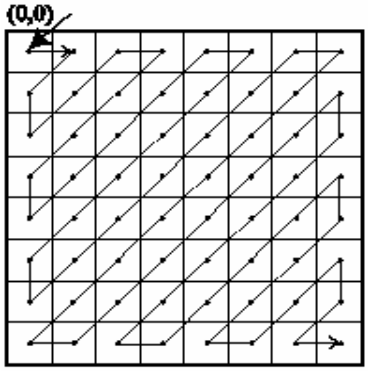


Fig 6 Zigzag ordering

Basically, this matrix we have chosen is same for all images and by the data given quality ratio is equals 100 the DCT coefficients are not changed, then all the quality will be chosen 50. It will be the basic matrix we have chosen.
The reason for computing quantization matrix independent from the image is to make image data have less space by not involving quantization matrix [9].
The quantized coefficients are made one-dimensional sequence according to zigzag order from two-dimensional matrix. The reason for doing this is to make the ones at the lower triangular matrix in the quantized coefficients matrix.
The coefficients that are put in order by the order of zigzag make very long zeros coming after another and this will make the compression techniques in entropy encoder, which will be used after quantization more efficient [9].

2.3. Entropy Encoder-Decoder

Entropy encoder compresses quantized coefficients. Lossless compression methods such as Huffman Coding, Run-Length Coding etc…can be used for this process.
We have used Huffman coding, Run-Length encoding and Arithmetic encoding as entropy encoder. All subimage data, which are put in order during zigzag, are written on the file and these three encoders compress all image data.

2.3.1. Huffman Coding

Huffman's algorithm assumes that a single tree from a group of trees is constructed. Firstly, all the trees have a single nodewith a character and the character's balance. Trees are combined by distinguishing two trees, and made a new tree from the two trees. This reduces the number of trees by one at each step since two trees are merged into one tree.[10, 11]

- The algorithm begins with a forest of trees. All trees are one node, with balance of the tree equal to the balance of the character in the node. Charcters that happen most frequently have the highest weights. Characters that occur least frequently have the smallest balances.
- The next step includes repeating the first step until there is only one tree: Firstly, choose two trees with the smallest balances.
- Create a new tree whose root has a balance equal to the sum of the balances two trees and whose left subtree is first tree and whose right subtree is second tree.
- In this algorithm the last step contains the single tree left after the previous step is an optimal encoding tree.

2.3.2. Arithmetic Coding

Arithmetic coding is a method of encoding data using a variable number of bits. The number of bits used to encode each symbol varies according to the probability assigned to that symbol. Low probability symbols use many bits; high probability symbols use fewer bits.
So far, this makes Arithmetic Coding sound very similar to Huffman coding. However, there is an important difference. An arithmetic encoder doesn't have to use an integral number of bits to encode a symbol. If the optimal number of bits for a symbol is 2.4, a Huffman coder will probably use 2 bits per symbol, whereas the arithmetic encoder uses very close to 2.4. This means an arithmetic coder can usually encode a message using fewer bits. [11]

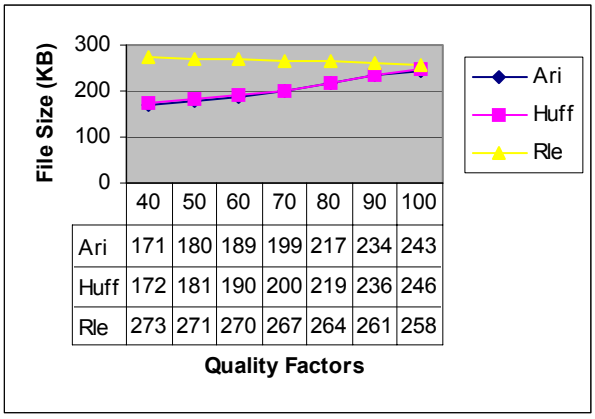
2.3.3. *Run-Length Encoding (RLE)*

Run-length coding is probably the simplest coding scheme that takes adventage of the context. The basic idea is to identify strings of contiguous messages of equal value and replace them with a single occurrence along with a count, although there are many variants.

For example, the message sequence acccbbaaabb could be transformed to (a, 1), (c, 3), (b, 2), (a, 3), (b, 2). Once transformed, a probability coder (e.g., Huffman coder) can be used to code both the message values and the counts. An example of a real-world use of run-length coding is for Facsimile (fax) machines. Fax machines transmit black-and-white images. Each pixel is called a pel and the horizontal resolution.The vertical resolution varies depending on the mode.Since there are only two message values black and white, only the run-lengths need to be transmitted. For example, the sequence bbbbwwbbbbb would be transmitted as 1,4,2,5.

The facsmile standard uses static Huffman codes to encode the run-lengths, and uses separate codes for the black and white pixels. [10]

50	271	-5.86%
60	270	-5.47%
70	267	-4.30%
80	264	-3.13%
90	261	-1.95%
100	258	-0.78%



3. Conclusions

When quantization matrix is being existed and the quality is 40-100, DCT is used and examined for the image on Fig1. The results we have found are below

Table 1: Compression Ratios for Huffman Encoder

Original File Size: 256KB		
Quality	Compressed File size (KB)	Compression Ratio
40	172	33.07%
50	181	29.57%
60	190	26.07%
70	200	22.18%
80	219	14.79%
90	236	8.17%
100	246	4.28%

Table 2: Compression Ratios for Arithmetic Coding

Original File Size: 256KB		
Quality	Compressed File size (KB)	Compression Ratio
40	171	33.46%
50	180	29.96%
60	189	26.46%
70	199	22.57%
80	217	15.56%
90	234	8.95%
100	243	5.45%

Table 3: Compression Ratios for Run-Length coding

Original File Size: 256KB		
Quality	Compressed File size (KB)	Compression Ratio
40	273	-6.64%

We use ten different fingerprint images for three encoders. A maximum compression ratio for all entropy encoders we used is 34.375% when the quality equals 40. Huffman and Arithmetic encoder produced similar results.

From the tables, compression ratios of run length encoder (RLE) are negative. It means that RLE does not compress our quantized image data. It can be concluded that selected quantization base matrix is not suitable and it does not reduce the frequency of transformed image data. Quantized coefficients vary in a wide range so RLE does not find to compress long runlengths. RLE usually produced good results for binary images.

4. Acknowledgements

Mr. Armagan ELIBOL is grateful to TUBITAK (The Scientific and Technical Research Council of TURKEY) for their financial support during this research.

5. References

[1] A.B. Watson, Image Compression Using The Discrete Cosine Transform, Mathematica Journal, Vol.4, p.81-88, 1994

[2] R.C.Gonzales and R.E.Woods, Digital Image Processing, Addison-Wesley Publishing Company, September 1993

[3] J.N.Bradley, C.M.Brislawn and T.Hopper, The FBI Wavelet/Scalar Quantization Fingerprint Image Compression Standart, Tech.Rep.LA-UR-94-1409, Nat'l.Media Lab Conf., May 1994

[4] Ibrahim Emiroglu, Fingerprint Image Enhancement & Recognition, PhD Thesis, University of Hertfordshire, Herts, October 1997

[5] C.I.Watson and C.L.Wilson, NIST Special Database 4, Fingerprint Database, National Institute of Standards and Technology Advanced Systems Division, Image Recognition Group, March 1992.

[6] S. Saha, Image Compression-from DCT to Wavelets:A Review,www.acm.org/crossroads/xrds63/sahaimgcoding.html.

[7] http://www.cs.sfu.ca/undergrad/CourseMaterials/CMPT365/material/notes/Chap4/Chap4.2/Chap4.2.html
[8] H.Tao,Data Compression Lecturer Notes, University of California
[9] V.Britanak, Transform and Data Compression, Chapter 4,CRC Press LCC, 2001
[10] W. Kou, Digital Image Compression: Algorithms and Standards (Kluwer International Series in Engineering and Computer Science, 333), Kluwer Academic Publishers, 1995
[11] http://datacompression.info/