ISSN 2278-3083
Volume 8, No.6, November –December 2019
International Journal of Science and Applied Information Technology
Available Online at http://www.warse.org/ijsait/static/pdf/file/ijsait06862019.pdf
https://doi.org/10.30534/ijsait/2019/06862019

Comparative Study between Different Image Compression Algorithms

Jumana Al-Shweiki[1], Nawal Hamdan[2], Khalid Alkaabneh[3]
[1]Computer Science Department, Amman Arab University, Amman, Jordan

ABSTRACT

This term paper presents a comparison between compression algorithms. The two types of image compression, which are considered in this paper, are the lossless and lossy image compression algorithms. First we compare between two lossless image compression methods: Run Length Encoding and Huffman, then we compare between two lossy image compression methods: Discrete Cosine Transform and Wavelets.

Key words: Image Compression, Huffman, RLE, Wavelets, DCT.

1. INTRODUCTION

INTRODUCTION

The need for image compression presented itself with the rise of amount of data available to be shared, compression is useful in decreasing the storage size needed to store file by removing the redundancy of pixels, and it decreases the cost of transmitting files over the network, hence comes its great value and the need to develop compression methods is of upmost importance. The two common compression methods are categorized into two groups: lossless and lossy compression. The first method (Lossless image compression) entails that there is no loss in data, so the compression ratio will be small; it is used in applications where data loss is intolerable such as text compression, medical images. This type includes Run Length Encoding, Huffman Coding, Arithmetic Coding, and LZW.

The second method of image compression is Lossy, this type entails that the resultant image is not necessarily identical to the original image, it is used in application that stand for insignificant data loss, such as video and audio streaming, in this type of compression, data degradation and loss of quality is expected while it produces high compression rates and smaller compressed data size than the lossless methods. The lossy method Types include: Discrete Cosine Transform, Wavelet Transform, and JPEG.

Theoretical

[1] 1. COMPRESSION TECHNIQUES

Run Length Encoding

Run Length encoding is a general-purpose compression algorithm that is suitable for any kind of data. It is a simple data compression method and easy to implement; the word (RUN) means the repeated data elements, the repeated elements can be stored as a number of iterations beside this single data value.

For example:
If we have this statement: {RRRRRRRRJJJJJRRRRRJJJ}
We can store it using the RLE method as follows: 8R5J5R3J
So we replace the (21) elements into (8) elements.
This coding is very useful in graphics images; on the other hand, it is bad in files that do not contain much iteration of data, in this case the resultant compressed image may not be successfully compressed.

To implement the run length encoding in MATLAB following steps are executed [1]:
1. Read the gray scale image and rearrange data of image as single row vector.
2. Convert all intensities values to binary state & obtain a binary stream representation of image.
3. Count consecutive 1’s & 0’s appeared in a sequence and store them as run length encoded sequence.
4. Get the compression ratios using the original size of image and the size of run length encoded sequence.

Huffman Coding

David Huffman described Huffman coding in 1952, it is a lossless data compression technique that depends on the frequency of a symbol in a file using binary representation, and it assigns the most frequent symbol with the shortest binary bit string.

For example if we have the following data: XXXXXXYYYYZZ
Using the run length encoding we will result
{(2*6) + (2*4) + (2*2) = 24}.

However, when using Huffman coding it will result in a smaller bit output:

We assign: X by the code 0 (1 bit)
Y by the code 10 (2 bits) Z by the code 11 (2 bits)

Therefore, the resulting file will have the size

{(1*6) + (2*4) + (2*2) = 18}

Therefore, in theory, this indicates that the Huffman coding results in a smaller file size than RLE. Implementing Huffman coding on MATLAB v7.12 (R2011a), following steps are executed [1]:
1. Read a grayscale image & convert the resulting array into a single row vector.
2. Form a Huffman encoding tree using the probability of symbols in gray scale image read.
3. Encode each symbol independently using the encoding tree.
4. Get the ratio of compression from the size of the original image and the size of the Huffman coded sequence.

Discrete Cosine Transform

As we discussed previously, the main purpose of compression is the removal of the redundancy between neighbor pixels, so the DCT method is one of the common methods in Lossy compression that is used for that purpose.

We can describe the DCT Method by the following two steps:

First Step: we divide the image into 8*8 blocks of pixels.

Second Step: we apply the DCT to each block from left to right and from top to bottom.

Notes:
*Each block will be compressed because of quantization.
*The resulting array is stored in a reduced amount of space.
*To decompress the image we can use the inverse discrete cosine transform (IDCT).

Wavelet Transform

The wavelet transform method is a new concept (Founded 30 years ago) but there are a few resources of it, wavelet was proposed by Stephan Mallat in 1988, most of books and articles which written about wavelet are written by math people.

In brief, wavelet method analyzes the frequency components of the signals, and this is useful to remove the noise from the images in fields such as medical imaging.

In this paper, we used the Embedded Zero Tree wavelet algorithm.

This algorithm is simple yet very effective image compression algorithm, having the distinct property that the bits in the bit stream are generated on the basis of Importance, which produces a sequence of binary decisions which distinguishes the image. This algorithm produces effective results yet does not require any training or prior knowledge of the source thus presenting a very powerful tool for compression.

The EZW algorithm is based on four key concepts [5]:

1. Discrete wavelet transform
2. Prediction of the absence of significant information across scales by exploiting the self-similarity inherent in images.
3. Entropy-code successive approximation quantization.
4. "Universal" lossless data compression which is achieved via adaptive arithmetic coding.

Experimental Setup and Analysis

We have tested the aforementioned methods using Matlab, we have conducted the experiment on 20 gray scale images with size of 512x512, and to measure the effectiveness of each algorithm we have computed three values:

1. Compression Ratio (CR)

It is known as the ratio of the original size of the image/ data to the compressed image/data

CR= Original size / Compressed size

It gives a very good indication about the effectiveness of the compression algorithm, if the value of the CR is more than 1, then the compression algorithm is successful in producing a valid compressed file, on the other hand, if the compression ratio was less than 1 then the algorithm was unsuccessful and needs to be optimized.

2. Mean Square Error (MSE)

It is the cumulative squared error between the original image and the compressed image. MSE=

$$\frac{1}{\mathrm{MN}} * \sum_{x=0}^{n} \sum_{y=0}^{m} \{I(x,y) - I'(x,y)\}^2$$

3. Peak Signal to Noise Ratio (PNSR)

It is the ratio between the maximum power of a signal and the power of the noise, PSNR used to measure the quality of reconstruction in image compression.

It is defined as: PSNR □
10 □log 1 0*(255 ^2) / MSE

2.RESULANTS AND DISCUSSION

After applied RLE and Huffman algorithm, we noticed the following result as table 1 and figure 1.

Table 1:.Comparison between RLE and Huffman

Image Number	RLE Compression Ratio	Huffman Compression Ratio
1	1.533	5.2372
2	1.459	5.9013
3	1.967	5.9349
4	2.727	6.0394
5	1.668	5.1258
6	0.879	6.1344
7	2.128	6.2183
8	4.54	5.5511
9	1.4	5.8884
10	1.085	5.6892
11	1.226	6.2286
12	3.34	6.0715
13	2.64	5.7255
14	1.234	6.4297
15	1.698	6.4232
16	1.903	6.1935
17	2.398	6.617
18	5.789	6.1798
19	1.179	6.3007
20	1.297	5.9734

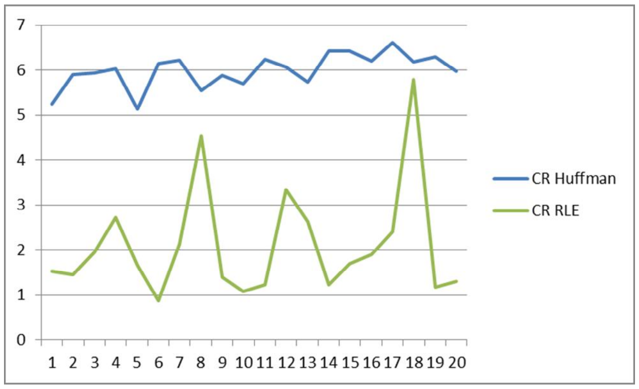

Figure 1: Huffman CR vs. RLE CR

The figure shows the consistency and effectiveness of the Huffman coding technique over the RLE.

We notice that the RLE was unsuccessful in compressing image #6, which has CR of less than one which shows the limitations of this algorithm which depends on the repetition of certain pattern.

On other hand, when compare between DCT and Wavelet we observed the following result as in table 2 and figure 2.

Table 2:.Comparison between DCT and Wavelet

Image	Wavelet CR	DCT CR	Wavelet PSNR	DCT PSNR
1	21.4931	7.5	5.9466	8.25
2	14.2479	7.558	6.4203	8.2458
3	13.3236	8.151	10.2715	8.2346
4	14.2822	7.348	9.4872	8.2381
5	13.7398	7.367	7.2996	8.2439
6	19.6129	6.815	10.4359	8.2349
7	17.0795	5.379	2.8824	8.2641
8	14.2307	8.338	9.2843	8.2327
9	21.0651	6.83	6.9841	8.2427
10	6.4064	8.435	2.8468	8.268
11	16.4349	5.855	2.2254	8.269
12	19.1391	6.767	8.4534	8.2378
13	7.9037	7.386	3.7651	8.2611

14	38.7005	5.052	7.974	8.2413
15	27.4773	5.903	9.2622	8.2362
16	35.0212	5.785	7.1855	8.2413
17	32.4295	5.079	8.3029	8.2352
18	35.3481	5.31	8.3086	8.2371
19	21.1323	6.354	10.1245	8.231
20	24.5094	6.178	8.2929	8.2471

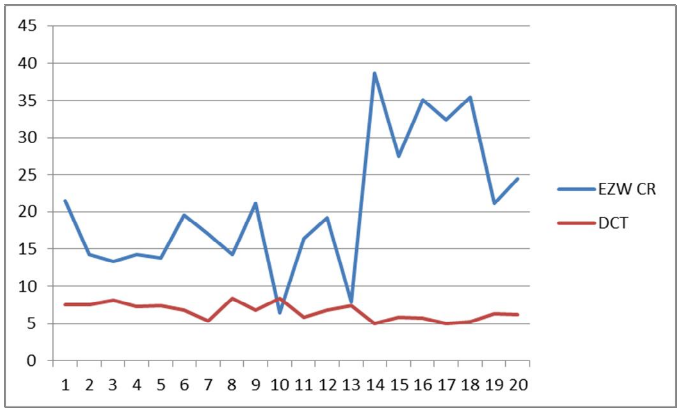

Figure 2: DCT CR vs. EZW CR

We notice the overall superiority of the EZW wavelet algorithm in comparison to the DCT in regards of compression ratio as figure 3.

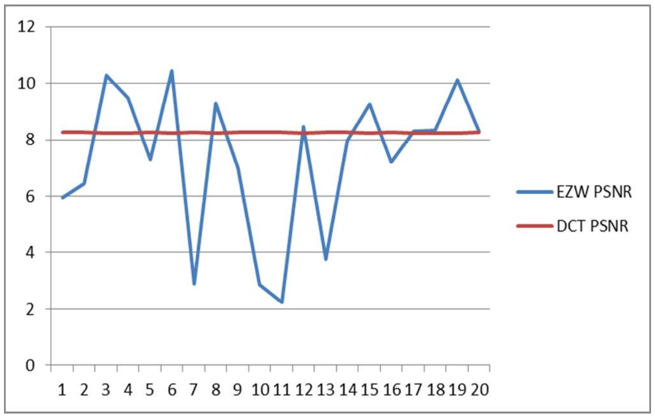

Figure 3: EZW PSNR vs. DCT PSNR

We notice that the PSNR for the DCT is consistent while the PSNR for the EZW wavelet is varying which is expected for the varying degrees of Compression ratios produced by this algorithm compared with the DCT which had somewhat modest CR values in comparison.

But overall, the higher PSNR values produced in EZW indicate a higher signal to noise ratio which is a good indication of the effectiveness of the algorithm since the signal here is the original image and the noise is the reconstruction of the compressed image.

3.CONCLUSION

In this paper, we tested four kinds of image compression algorithms. The first two are the RLE and Huffman coding algorithms, they represent the lossless image compression class, we have founded that the Huffman coding outperforms The Run Length Encoding algorithm based on the compression ratio, but RLE is very simple to implement and fast to perform so it's suitable for applications that require simplicity over size of compressed data size

The second comparison was conducted between two types of lossy image compression algorithms, we've tested the Discrete Cosine Transform and the Wavelet Transform represented by EZW algorithm, based on the results, the EZW outperforms the DCT and gives better compression results.

REFERENCES

1. Patel, D., Bhogan, V., & Janson, A. (2013). Simulation and comparison of various lossless data compression techniques based on compression ratio and processing delay. *International Journal of Computer Applications*, *81*(14) https://doi.org/10.5120/14186-2423
2. Gonzalez, R. C., & Woods, R. E. (2012). Digital image processing
3. Telagarapu, P., Naveen, V. J., Prasanthi, A. L., & Santhi, G. V. (2011). Image compression using DCT and wavelet transformations. International Journal of Signal Processing, Image Processing and Pattern Recognition, 4(3), 61-74
4. Parmar, H. M., & Scholar, P. G. (2014). Comparison of DCT and wavelet based image compression techniques. *International Journal Engineering Development and Research*, *2*(1), 664-669.
5. J. M. Shapiro, "Embedded Image Coding Using Zero trees of Wavelet Coefficients," IEEE Trans. on Signal Processing, Vol. 41, No. 12, pp. 3445 – 3462, Dec. 1993 https://doi.org/10.1109/78.258085
6. Dr. Amin Mubarak Alamin Ibrahim, Dr. Mustafa Elgili Mustafa, "Comparison between (RLE and Huffman) Algorithms for Lossless Data Compression" December – January 2015, 1808 – 1812.

7. Abualigah, L. M. Q. (2019). *Feature selection and enhanced krill herd algorithm for text document clustering*. Berlin: Springer. https://doi.org/10.1007/978-3-030-10674-4
8. Varsha Bansal, Pratishtha Gupta, Suhail Tomar,” The Implementation of Run Length Encoding for RGB Image Compression” International Journal of Advanced Research in Computer Engineering & Technology (IJARCET) Volume 3 Issue12, December2014