Zeros Removal with DCT Image Compression Technique

Dr. Nushwan Yousif Baithoon

University of Baghdad/College of Science for Women /Department of Computer Science

E-Mail: nybalnakash@yahoo.com

Abstract

The discrete cosine transform (DCT) is a method for converting a signal into plain frequency components. It is extensively used in image compression. In this paper a new technique is proposed, namely ZRDCT (Zeros Removal with DCT) which is based on a lossy compression, and used to enhance image data compression.

Image quality is measured impartially, using peak signal-to-noise ratio (PSNR) or picture quality scale, and individually using perceived image quality with compression factor (CF) being the main theme of this paper, taking into consideration the preservation of well PSNR outputs.

The performance of DCT compression generally degrades low bit-rates mainly because of the underlying block-based DCT scheme. Experimental results demonstrated the effectiveness of the ZRDCT approach, which enhanced the performance of the conventional DCT image compression methods, by investigating and interrogating the whole image and hence enforcing mechanisms for finding possible redundant information and therefore the removal of unnecessary data which lead to an improvement in CF without upsetting PSNR.

The new technique also proved to have low distortions with good quality PSNR, commendable CF and good execution time, when compared to other various DCT schemes and with some wavelet based image compression.

*Keywords***:** *Discrete Cosine Transform, Image Compression, Peak Signal-to–Noise Ratio, Compression Factor.*

1. Introduction

In images, it is a known fact that neighboring pixels are correlated and therefore contain redundant information.

A typical still image contains a huge amount of spatial redundancy in plain areas where adjacent picture elements (pixels) have almost the same values. It means that the pixel values are highly correlated [1].

In addition, a still image can contain subjective redundancy, which is determined by properties of a human visual system (HVS). An HVS presents some tolerance to distortion, depending upon the image content and viewing conditions. As a result, pixels must not always be replicated exactly as originated and the HVS will not detect the difference between original image and mimicked image. The redundancy (both statistical and subjective) can be removed to achieve compression of the image data [1]. Such redundancies are the key features that this work is investigating. The basic measure for the performance of a compression

algorithm is compression factor (CF), defined as a ratio between original data size and compressed data size. In a lossy compression scheme, the image compression algorithm should achieve a trade off between CF and image quality, namely peak signal-to-noise ratio (PSNR). Higher CFs will produce lower PSNRs and vice versa and these two can also vary according to input image characteristics and content [1].

In this work, great efforts were explored to find practical blending techniques to attain high CF without affecting PSNR.

Transform coding is a widely used method of compressing image information. In a transform - based compression system two-dimensional (2-D) images are transformed from the spatial domain to the frequency domain. An effective transform will concentrate useful information into a few of the low-frequency transform coefficients. An HVS is more sensitive to energy with low spatial frequency than with high spatial frequency. Therefore, compression can be achieved by quantizing the coefficients, so that important coefficients (low-frequency coefficients) are transmitted and the remaining coefficients are discarded. Very effective and popular ways to achieve compression of image data are based on the discrete cosine transform (DCT) and discrete wavelet transform (DWT) [2].

In this paper, DCT image compression is used with an added technique, namely ZRDCT, which will elevate CF without affecting PSNR.

For lossy image compression and in general, two kinds of redundancy may be acknowledged [3]:

- Spectral Redundancy: This is the correlation between different colour planes or spectral bands.
- Spatial Redundancy: This is the correlation between neighboring pixel values.

Therefore, the idea is to reduce the number of bits needed to represent an image by removing the spectral and spatial redundancies as much as possible [3]. In this paper, the above two points were heavily exploited.

2. YUV Model

In order to take advantage of the high spectral correlation that is inherent in the YUV model and hence reduce computational complexities, the RGB input data is converted to YUV colour space. Here, Y denotes the luminance component U and V are the two chrominance components [4 and 5].

For Y, equation (1) can be determined from the RGB model using the following relation;

$$Y=0.299R+0.587G+ 0.114B \quad \ldots\ldots\ldots \quad (1)$$

From equation (1), it can be noted that the three weights associated with the three primary colours, R, G, and B, are different. The difference in the magnitudes reflects different responses of the human visual system (HVS) to various primary colours.

Also, and as an alternative of being directly related to hue and saturation, the other two chrominance components, U and V, are defined as color differences, as shown in equations (2) and (3).

$$U=0.492(B-Y)\ldots\ldots\ldots\ldots \quad (2)$$

$$V=0.877(R-Y)\ldots\ldots\ldots\ldots (3)$$

3. DCT Compression

The block-based segmentation of source image is a fundamental limitation of the DCT-based compression system. The degradation is known as the "blocking effect" and depends on block size. Larger blocks lead to more efficient coding, but require more computational power. Image distortion is less provoking for small than for large DCT blocks, but coding efficiency tends to suffer [1].

Therefore, most existing systems use blocks of 8x8 or 16x16 pixels as a compromise between coding efficiency and image quality. Each block is transformed from spatial to frequency domain using 2-D DCT basis function. For an 8x8 blocking process, it will split the image into 8x8 blocks small enough to assume high correlation between adjacent pixels and apply 8x8 DCT transform to each block to shift energy in each block to uppermost entries. The resulting frequency coefficients are quantized and finally output to a lossless entropy coder. DCT is an efficient image compression method because it can de-correlate pixels in the image, since the cosine bases are orthogonal, and condense most of the image energy into few transform coefficients. Broadly speaking, compressing a set of correlated pixel values using DCT may be done by [6]:

- Compute DCT coefficients of the image.
- Quantize the coefficients.
- coefficients either by variable-length coding or arithmetic coding.

Entropy encodes the quantized

In DCT image compression, it is common to use two-dimensional DCT which can be described by an nxn transform matrix.

The most common 2D DCT is as given in equation (4).

$$C(u,v) = \alpha(u)\alpha(v)\sum_{x=0}^{N-1}\sum_{y=0}^{N-1} I(x,y)\cos\left[\frac{(2x+1)u\pi}{2N}\right]\cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (4)$$

Where

I(x, y) = Pixel value

$$\alpha(k) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } k = 0 \\ \sqrt{\frac{2}{N}} & \text{for } k = 1,2,\ldots,N-1 \end{cases}$$

Where, N is the image size.

For matrix representation of equation (4), equation (5) may be used, giving equation (6).

$$T_{i,j} = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } i = 0 \\ \sqrt{\frac{2}{N}}\cos\left[\frac{(2j+1)i\pi}{2N}\right] & \text{for } i \neq 0 \end{cases} \quad (5)$$

$$C = TI\,T' \quad \ldots\ldots\ldots\ldots (6)$$

Where, T ' is the transpose of T.

While the inverse transform (I), equation (7), is

$$I = T'\,C\,T \quad \ldots\ldots\ldots\ldots (7)$$

4. Quantization

DCT-based image compression relies on the quantization of the image's DCT coefficients. Quantization is the process of reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it [7].

A simple example of quantization is the rounding of real into integers. To represent a real number between 0 and 7 to some specified precision takes many bits. Rounding the number to the nearest integer gives a quantity that can be represented by just three bits.

In this process, a reduction of the number of possible values of the quantity (and thus the number of bits needed to represent it) at the cost of losing information. A "finer" quantization, that allows more values and loses less information, can be obtained by dividing the number by a weight factor before rounding. Taking a larger value for the weight gives a "coarser" quantization. De-quantization, which maps the quantized value back into its original range, but not its original precision, is achieved by multiplying the value by the weight [6].

5. Huffman Coding

Huffman coding is an efficient source coding algorithm for source symbols that are not equally probable. A variable length encoding algorithm was suggested by Huffman in 1952, based on the source symbol probabilities P (xi), i=1, 2……., L. The algorithm is optimal in the sense that the average number of bits required to represent the source symbols is a minimum provided the prefix condition is met [1].

The Huffman decoding process will be a matching process, using the Huffman dictionary. Hence, the received string bits will be interrogated for its symbol and how it matches the representation given in the Huffman dictionary.

6. ZRDCT Technique

The idea behind ZRDCT is to find a way such that long strings of zeros are created after the 8x8 DCT blocking process. The foundation behind ZRDCT technique is based on the use of thresholding, which will be employed at various stages.

As an example, after converting an RGB image to its YUV color space, consider the DCT output of the Y color space, DCTY, which will be a matrix.

This matrix will be built of 8x8 blocks, which is an intrinsic process made by the DCT procedure. For a typical 8x8 sample block from a typical source image, most of the spatial frequencies have zero or near-zero amplitude and need not be encoded. In principle, the DCT introduces no loss to the source image samples; it merely transforms them to a domain in which they can be more efficiently encoded. Also, since adjacent image pixels are highly correlated, then the DCT processing step lays the foundation for achieving data compression by concentrating most of the signal in the lower spatial frequencies [6 and 7].

Now, for the rest of the discussion, and regarding each of the 8x8 blocks used in the DCT process, the first coefficient (top left) will be termed the DC (low frequency) coefficient which is large while the remaining coefficients, which are much smaller, will be called the AC (high frequency) coefficients.

ZRDCT will start by saving all of the available DC coefficients of DCTY in a special buffer called ***top array*** and a copy of DCTY is made called DCTY(1) in which its DC coefficients are made equal to zero. Note that the DC coefficients positions can be easily calculated since they always lie at the top left corner of the 8x8 block.

Then a threshold value, threshold1, is used on all AC coefficients values, to be called ***other values***, so as to zero most of them. This threshold is an empirical value and can be adjusted to give the required result. All ***other values*** that do not comply with the threshold are left in their current AC positions. The output of this stage will be called DCTY (2).

Now, a test is made to find if each row in DCTY (2) has more than 2 ***other values***, if so index that row, else remove it. The value of 2 can be adjusted by a second threshold, threshold2, which is also an empirical value and can be adjusted to give the required result. A table called ***tab array*** is formulated to save these results, noting that the indexed row numbers will be in an ascending order.

The ***tab array*** table will be applied on DCTY(1) so as to formulate a new matrix that will have the repositioned ***other values***, with their respected row numbers, thereby giving a more compact output by the elimination of the zeroed rows. The output of this stage will be called DCTY (3).

From DCTY (3), two arrays are constructed which is done by searching each row in DCTY (3) and record information about the ***othervalues***. The first array will contain the ***other values***, to be called ***other AC array***. The second array will contain their true positions, to be called ***position array***.

A quantization process, Q, to the ***top array*** and ***other AC array*** is being done by reducing the number of possible values of a quantity, thereby reducing the number of bits needed to represent it. Note that the two arrays ran at a different level of quantization. Quantization output processes for ***top array*** and ***other AC array*** to be called Q (***top array***) and Q (***other AC array)*** respectively.

A discrepancy procedure, D, based on a differencing process is being done. This process will be applicable only to Q (***top array***), ***position array*** and ***tab array***. The differencing process relies on saving the first value of a given array in a new array while consecutive values of the new array will indicate the difference between preceding and current value of the array in question. The differencing output process for Q (***top array***), ***position array*** and ***tab array*** to be called D (Q (***top array***)), D (***position array)*** and D (***tab array)*** respectively. No such a differencing process is done to the Q (***other AC array***) since its values are too random. The discrepancy procedure was done so that data will be constructed in a form more suitable to the next stage, of Huffman coding.

A Huffman coding process is applied to D (Q (***top array***)), Q ***(other AC array)***, D (***position array)*** and D (***tab array***). The output of each is a dictionary and data.

U and V colour spaces were down sampled to u and v by 2x2 adjacent pixels averaging, before DCT operation. The above ZRDCT process will be applicable to DCTu and DCTv

in the same way as shown above in the DCTY case.

Finally the whole Huffman outputs of the above are packed ready to be sent at the encoder side.

At the decoder side a reverse procedure of the above encoding process is done.

7. ZRDCT Implementation

The general steps involved in ZRDCT process are as follow;

A. Encoder Side;

1. Convert RGB image "I" to Y, U and V color space.
2. Down sample U & V color spaces only.
3. 2-D 8x8 blocks DCT operation on step 2 output, and layer Y of step 1.
4. Initiate DC, AC, position and table arrays.
5. Quantize DC and AC arrays of step 4.
6. Discrepancy to position and table arrays and to the quantization output of DC array in step 4.
7. Huffman code step 6 outputs and the quantization output of AC array in step 5.
8. Pack Huffman outputs.
9. Send encoded groups of step 8.

Figure (1) shows the general block diagram of the ZRDCT encoder.

B. Decoder Side;

A reverse procedure of the above encoding process is done.

8. Results and Discussions

The proposed ZRDCT technique was implemented using MATLAB Ver. 7 and tested on a Pentium IV 2.2 MHz Core 2 Due CPU. In all of these investigations, a coloured Lena image (256x256, 24 bits/pixel) has been used as a test material. Table (2) shows the outcome of the gained results. Note that PSNR values of these results were keep at just over 30 dBs for the sake of CF comparisons.

PSNR and CF, equations (8) and (9), are calculated as follows [8 and 9].

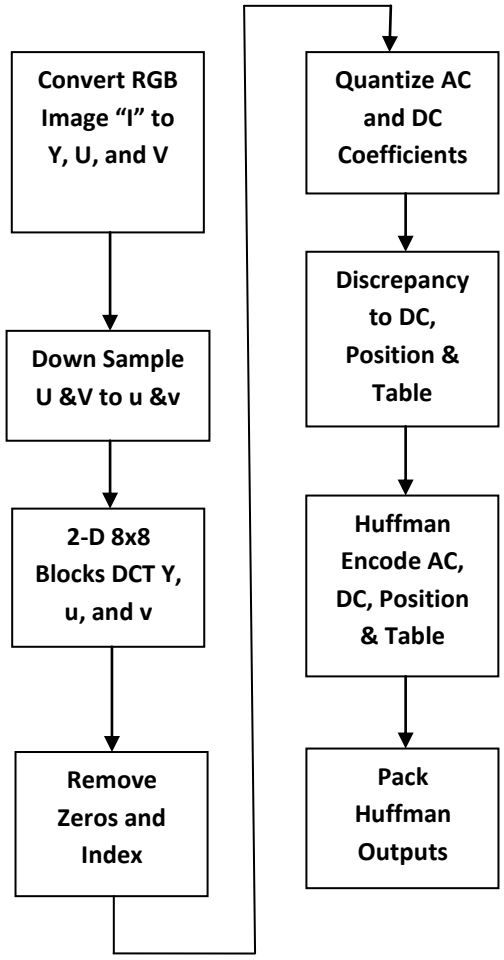

Figure (1): ZRDCT General Encoder Block Diagram

$$PSNR = 10\log_{10}\frac{(L-1)^2}{RMS} \quad (8)$$

Where, $RMS = \frac{1}{m \cdot n}\sum_{i=1}^{m}\sum_{j=1}^{n}\left(x_{i,j} - x'_{i,j}\right)^2$

L is the number of gray levels, X is the original image, X' is the compressed image, m and n are the image width and height respectively.

$$CF = \frac{\text{Image size before compression}}{\text{Image size after compression}} \quad \ldots\ldots\ldots \quad (9)$$

The first result of table (2) suffers from low CF as compared to the other two, although it has the second lowest execution time (ET).

The second result has a better CF than its predecessor but it suffers from ET. This was due to the labours computational density of the Huffman coding scheme being done on large data sets.

However, a closer look at result 3 reveals that a much better CF and ET can be achieved, as compared to the above results, with the use of the ZRDCT technique. The removal of redundant and therefore unwanted data boosted CF without affecting PSNR.

Figure (2) shows the original image with output images of table (2) results.

Saha [10], in his comparison of wavelet compression methods**,** showed that using DWT with fixed length coders can achieve a PSNR of just lower than 30 dBs at a CF of about 5.

Grgic and Grgic [11], in their performance analysis of image compression using wavelets, showed that using DCT or DWT with variable length coders can achieve a PSNR of 30 dBs at a CF of just lower than 20.

Amerijckx, Legat and Verleysen [12], proposed a compression scheme, based on the use of the organization property of Kohonen maps, and achieved a PSNR of 24.7 dBs at a CF of 25.22.

When these results are compared to that of ZRDCT in table (2), the proposed ZRDCT technique performance has clearly the initiative.

Table (2): Test Results

	DCT Scheme	**PSNR (dBs)**	**CF**	**ET (Sec)**
1.	DCT (no Huffman Coding)	30. 0052	10.073	2.387
2.	DCT with Huffman Coding	30.0050	14.78	32.98
3.	ZRDCT	30.0033	35.540	١.٤٩٧٦

Original Image

Figure (2): Original Image with Output Images of Table (2)

9. Conclusions

DCT-based image coders perform well at modest bit rates. However, at higher compression ratios, image quality degrades because of the artefacts resulting from the DCT blocking process.

An enhancement to DCT based image compression technique has been proposed to achieve high image qualities and good compression factors with low computational complexities.

In this paper a ZRDCT technique was proposed. Such a technique showed low distortions with good quality PSNR, commendable CF and good ET, when compared to other various schemes.

The improvement attained using the proposed technique was due to the fact that ZRDCT enhanced the performance of the conventional DCT image compression methods by investigating and interrogating the whole image and hence enforcing mechanisms for finding possible redundant information and therefore the removal of unnecessary data which lead to an improvement in CF without upsetting PSNR.

References

[1]. Salomon, D. **2007.** Data Compression the Complete Reference, *Springer-Verlag London Limited*. pp. 337-353.

[2]. Mano, J. **1999.** Compressed Image File Formats JPEG, PNG, GIF, XBM, BMP, *Addison Wesley Longman, Inc*. pp. 121-148.

[3]. Nelson, M. and Gally, J. **1996**. The Data Compression Book, *New York NY: M&T Books*. pp. 326-344.

Archarya, T., Tsai, P. **2005**. JPEG2000 Standard for Image Compression: Concepts, Algorithms and VLSI Architectures, *Wiley, Hoboken*. pp. 60-64.

[4]. Pu, I. **2006.** Fundamental Data Compression, *Butterworth-Heinemann Publications*. pp. 163-165.

[5]. Salomon, D. **2007.** Variable-Length Codes for Data Compression, *Springer-Verlag London Limited*. pp. 9-12.

[6]. Ahmed, N., Natrajan, T. and Rao, K.R. **1974.** Discrete Cosine Transform, *IEEE Transactions on Computers*, vol. C-32. pp. 90-93.

[7]. Salomon, D. **2008.** A Concise Introduction to Data Compression, *Springer-Verlag London Limited*. pp. 144-155.

Watson, A. **1994**. Image Compression Using the Discrete Cosine Transform, *NASA Ames Research Center*. pp. 81-88.

[8]. Saha, S. **2004**. Image Compression – from DCT to Wavelets: A Review, *The Association for Computing Machinery*, Inc. pp. 15-16.

[9]. Grgic, S. Grgic, M., **2001**. Performance Analysis of Image Compression Using Wavelets. *Member, IEEE, and Branka Zovko-Cihlar, Member, IEEE*. VOL. 48, NO. 3, JUNE 2001. pp. 682-689.

[10]. Amerijckx, C., Legat, J., Verleysen, M. **2003.** Image Compression Using Self-Organizing Maps, *Systems Analysis Modeling Simulation*, Vol. 43, No. 11, November 2003. pp. 1529–1543.

الغاء الاصفار مع تحويل الجيب تمام المتقطع لضغط الصور الرقمية

الخلاصة

تستخدم طريقة تحويل الجيب تمام المتقطع (DCT)، لتغيير اشارة معينة الى الصيغة المتقطعة، و هي طريقة مشاع استخدامها في عمليات ضغط الصور الرقمية.

في هذا البحث تم اقتراح طريقة جديدة هي الغاء الاصفار مع تحويل الجيب تمام المتقطع لضغط الصور الرقمية (Zeros Removal with DCT) والتي تستخدم في ضغط الصور الرقمية مع خاصية فقدان بعض البيانات من دون التأثيرعلى جودة الصورة. تم قياس جودة الصورة بأستخدام نسبة قمة الاشارة الى الضوضاء (PSNR) مأخوذا بنظر الاعتبار معامل الضغط (CF) والذي يمثل الحجر الاساس لهذا البحث.

ان اداء طريقة تحويل الجيب تمام المتقطع في ضغط البيانات تعاني، و في الاخص، من اخفاق في معاينة المعلومات ذات المعدلات الواطئة بسبب خاصية التقطيع المضمنة المستخدمة في هذه الطريقة. لقد اعتمدت طريقة ZRDCT على فحص وتمحيص الصورة باكملها وبالتالي اضفاء اساليب معينه لاحتمالية ايجاد معلومات من الممكن الغائها نهائيا وبالتالي رفع اداء معامل الكبس من دون التأثيرعلى نسبة قمة الاشارة الى الضوضاء .

لقد اظهرت النتائج المستوفاة من هذا البحث فعالية طريقة الغاء الاصفار مع تحويل الجيب تمام المتقطع لضغط الصور الرقمية من ناحية نسبة قمة الاشارة الى الضوضاء، معامل الضغط وزمن التنفيذ مقارنة مع مثيلاتها من الطرق الاخرى التي تستخدم طريقة تحويل الجيب تمام المتقطع و كذلك مقارنة مع بعض الطرق التي تستخدم طريقة التحويل المويجي (DWT).

المفاتيح : تحويل الجيب تمام المتقطع، ضغط الصور، نسبة قمة الاشارة الى الضوضاء، معامل الضغط.