DCT IMAGE COMPRESSION USING SLIDING RLE TECHNIQUE

Nushwan Yousif Baithoon
Department of Computer, College of Science for Women, University of Baghdad. Bagdad-Iraq

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 SRLE (Sliding Run Length Encoding) 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 demonstrate the effectiveness of the SRLE approach, in terms of PSNR, CF and execution time, over different mechanisms used with DCT image compression. The new technique also proved to have favourable results in terms of PSNR and CF when compared with some wavelet based image compression.

ضغط الصور بطريقة تحويل الجيب تمام المتقطع بأستخدام تقنية الانزلاق التشفيري بحساب طول الخطوة

نشوان يوسف بيثون
قسم علوم الحاسبات، كلية العلوم للبنات، جامعة بغداد. بغداد-العراق.

الخلاصة

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

في هذا البحث تم اقتراح طريقة جديدة هي الانزلاق التشفيري بحساب طول الخطوة (Sliding Run Length Encoding) والتي تستخدم في ضغط الصور مع خاصية فقدان بعض البيانات من دون التأثير على جودة الصورة. تم قياس جودة الصورة بأستخدام نسبة قمة الاشارة الى الضوضاء (PSNR) مأخوذا بنظر الاعتبار معامل الكبس(CF) والذي يمثل الحجر الاساس لهذا البحث.

ان اداء طريقة تحويل الجيب تمام المتقطع في ضغط البيانات تعاني، وفي الاخص، من اخفاق في معاينة المعلومات ذات المعدلات الواطئة بسبب خاصية التقطيع المضمنة المستخدمة في هذه الطريقة.

لقد اظهرت النتائج المستوفاة من هذا البحث فعالية الانزلاق التشفيري بحساب طول الخطوة من ناحية قمة الاشارة الى الضوضاء (PSNR)، معامل الكبس(CF) 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).

1. Introduction

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

The primary task then is to discover a less correlated form of the image. Two main mechanisms of compression are of interest.

These are redundancy and irrelevancy reduction. Redundancy reductions intend to remove repetition from the signal source image. Irrelevancy reductions invalidate parts of the signal that will not be recognized by the signal receiver, namely the Human Visual System (HVS). [1]

Data compression is the technique to reduce the redundancies in data representation in order to decrease data storage requirements and hence communication costs. Reducing the storage requirement is equivalent to increasing the capacity of the storage medium and hence communication bandwidth. [1]

In the case of lossless compression methods, the reconstructed image, after compression, is digitally the same as that of the original image but it can only attain a modest amount of compression. [1]

However, an image reconstructed following lossy compression contains degradation relative to the original. Though, lossy schemes are capable of achieving to a large extent a higher compression than that of lossless compression methods. In sound lossy schemes, under typical viewing conditions, no visible loss is apparent.

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

- Spectral Redundancy: This is the correlation between different colour planes or spectral bands.
- Spatial Redundancy: This is the correlation between neighbouring 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. The above two points are the main pillars of this paper, which is to be relied at for their exploitation. [1]

2. DCT Compression

In DCT image compression, blocks of pixels are represented using cosine functions of different frequencies. The high frequencies, which generally contribute less information to the image, are discarded. [1]

The important feature of the DCT, which makes it useful in data compression, is that it takes correlated input data and concentrates its energy in just the first few transform coefficients. And since the input data consists of correlated quantities, then most of the *n* transform coefficients produced by the DCT are zeros or small numbers, and only a few are large (normally the first ones). [2]

Therefore and broadly speaking, compressing a set of correlated pixels with DCT may be done by:

- Compute DCT coefficients of the image.
- Quantize the coefficients.
- Entropy encoding, with the quantized coefficients being coded with variable-length or arithmetic coding.

In DCT image compression, it is common to use two-dimensional DCT which can be described by an NXN transform matrix. [3]

The most common 2D DCT is as given in equation (1). [4]

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

Where

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

With N^2 being the image size.

For matrix representation of equation (1), equation (2) may be used, giving equation (3).

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

$$C = T\, I\, T' \quad (3)$$

While the inverse transform (I), equation (4), is

$$I = T'\, C\, T \quad (4)$$

3. YUV Model

In order to lower computational complexities, the RGB input data is converted to YUV colour space. In this model, Y denotes the luminance component; U and V are the two chrominance components. [5]

The luminance Y, equation (5), can be determined from the RGB model using the following relation:

$$Y = 0.299R + 0.587G + 0.114B \quad (5)$$

From equation (5), it can be noted that the three weights associated with the three primary colours, R, G, and B, are not the same. Their different magnitudes reflect different responses of the HVS to different primary colours.
Also, instead of being directly related to hue and saturation, the other two chrominance components, U and V, are defined as colour differences, as shown in equations (6) and (7).

$$U=0.492(B-Y) \quad (6)$$

$$V=0.877(R-Y) \quad (7)$$

4. The Sliding Consequence

It can be realized that the various compression algorithms in use today are based on and involve the knowledge of diverse physical and mathematical models and subjects. Lossy image compression is considered here, where a source model-based coding algorithm may be applied in the entropy, which is the average information content per symbol, encoding phase of the source encoding step after transformation and quantization. Moreover, upon the characteristics of the input data; each algorithm may give different compression performance.
A typical run-length encoder identifies runs of the same symbol and replaces each run with a token that includes the symbol and the length of the run. If the run is shorter than a token, the raw symbols are output, but the encoder has to make sure that the decoder can distinguish between tokens and raw symbols. [6]
The intention of this paper is the realization of a hybrid technique, namely SRLE, which will combine the goodness of RLE with the effectiveness of the proposed technique. The word ***sliding*** comes about from the process of interrogating (sliding over) the randomness of the entire RLE output which will be at the encoder side.

5. The Proposed Hybrid Technique

Since, RLE output stream is a combinational pairs of a value (coefficient **C**) and its run (repetition **R**); we will then be interested on how well such a pair is represented as compactly as possible at the transmitter side so as to further decrease the amount of bits to be sent thus enhancing overall bandwidth requirements by increasing CF, without affecting PSNR.
SRLE will calculate the probability distribution of **C** for a given RLE output, hence, the repetition probabilities of the **C** values are found. The **C** values with the highest probability will then be assigned the required amount of bits needed for its representation; also, this bit assignment is used to find the highest value that can represent other **C's**. A search of other possible values in **C**, ***SS***, which can be represented with the same number of bits, is then sought.
The rest of the **C** values that does not fit the above criterion, are assigned with a variable bit length, needed for the highest value, or ***HV***, of **C**, which will adequately represent them.
For example, suppose a sequence of **C** coefficients is given as 100, 70, 60, 30, 7, 4, 4, 4 and 3. Normally, the 100 value will require 7 bits for its representation, thus 7 bits multiplied by the number of coefficients 9 giving 63 bits total number of bits required to convey the whole bit stream of the **C** values.
Whereas with SRLE, since the value of 4 has the highest probability of occurrence then the number of bits needed for its representation is found as 3 bits. The 3 bits also will cover the values of 3 and 7; hence, five out of the nine values will be represented with 3 bits giving 15 bits. The other values of **C** will be represented with 7 bits i.e.7 multiplied by 4 giving 28 bits. For now, the total number of bits **T** will be 28 bits plus 15 bits giving 43 bits.
The **T** bits if transmitted will be meaningless to the decoder and thus it cannot reconstruct the original **C** values. Therefore a further technique is devised, to correctly symbolize the **T** string, by the use of tags.
Tagging will be used in such a way that the ***SS*** values may be given a 0 tag bit each, while the ***HV*** values are given a 1 tag bit each. In essence, the tags will act as a header to the trail of actual data values of **C**.
Consequently, these tags are used here to differentiate the 4 bit assignment values from that of the 7 bits, when processed by the decoder. Hence, and since there are 9 **C** values, 9 tag bits will be required. The total number of bits required to convey the whole bit stream of the **C** values will now be 43 bits plus 9 tag bits giving 52 bits.
Comparing the above two results of 63 bits with that of 52 bits, it can thus be seen that there is a reduction of ((63-52)/63)% which is about 17.4% of the total amount of data intended for transmission. It is therefore true to say that in a more practical condition with **C** being much longer and highly correlated; the results will be even more superior.

6. SRLE Implementation

As mentioned before, the input image is divided into 8-by-8 blocks, and the two-dimensional DCT is computed for each block. The DCT coefficients are then quantized, encoded and made ready for transmission. The receiver decodes the quantized DCT coefficients, computes the inverse two-dimensional DCT of each block, and then puts the blocks back together into a single image.
Figure 1. shows the block diagram of the SRLE encoder.

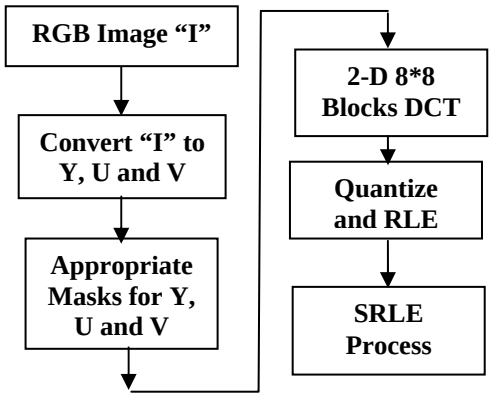

Figure 1: SRLE Encoder Block Diagram

To implement the SRLE scheme, the following algorithm has been put to practice:

A. Encoder Side;

1. A BMP RGB image "I" is loaded.
2. Convert I to Y, U and V colour space.
3. Prepare 8*8 masks for Y, U and V plans. These masks will serve as a filter so that only a number of DCT coefficients are allowed for further processing. Coarse filters are used for U and V components, while a light filter is used for Y component. This can be done, since; losing information in the U and V components compresses the image which will introduce distortions that cannot be detected by the eye, while losing information in the Y component is very obvious to the eye. The tailoring of such filters is image dependent.
4. 2-D 8-by-8 blocks DCT is computed for step 3 output.
5. Quantize and RLE DCT output of step 4.
6. SRLE output of step 5.

B. Decoder Side;

A reverse procedure of the above encoding process is done.

7. Results and Discussion

The proposed scheme 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 Lena image (256x256, 24 bits/pixel) has been used as a test material. Table (1) shows the outcome of the gained results.

Table 1: Test Outputs Obtained from Comparing Non-SRLE with SRLE Schemes

#	DCT Scheme	PSNR (dB)	CF	ET (Sec)
1.	DCT (No SRLE), without Huffman Coding	30. 005	10.252	1.7784
2.	SRLE DCT with Huffman Coding	30.005	14.780	28.950
3.	SRLE DCT without Huffman Coding	30.005	13.005	1.8252

PSNR and CF, equations (8) and (9), are calculated as follows. [4]

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

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.

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

Note that in table (1), and regarding the Huffman coding process in the second result, SRLE was only applied to the Huffman dictionary.
The first result of table (1) suffers from low CF as compared to the other two, although it surpasses them by its superior execution time (ET).
The second result has good PSNR and the highest CF of the three results, but it suffers from ET. This was due to the labours computational density of the Huffman coding scheme.
However, a closer look at result 3 reveals that much better trade-off outputs of PSNR, CF with excellent ET can be achieved with the use of the SRLE technique.
Figure 2. shows the original image with output images of table (1) results.

Figure 2: Original Image with Output Images of Table (1)

Ibraheem [7] used DWT for image compression. His test results indicated a PSNR of 24 to 40 with CF of 2 to 4. Saha [8], in his comparison of wavelet compression methods**,** showed that using DWT with fixed length coders can achieve a PSNR of just lower than 30 at a CF of about 5.

When these results are compared to table (1), SRLE has clearly the upper hand.

8. Conclusions

A variety of compression techniques have been put forward to achieve high image qualities and compression ratios bearing in mind and to some extent, low computational complexities. In this paper a SRLE, which is a hybrid technique, was proposed. Such a technique showed low distortions with favourable PSNR, encouraging CF and an approving ET, when compared to other various schemes.

The gains accomplished using this technique was due to the fact that SRLE improved the performance of the traditional RLE technique by further exploiting its probabilistic randomness.

References

1. Mano, J. **1999.** *Compressed Image File Formats JPEG, PNG, GIF, XBM, BMP*, Addison Wesley Longman, Inc. pp. 121-148.
2. Salomon, D. **2008.** *A Concise Introduction to Data Compression*, Springer-Verlag London Limited. pp. 144-155.
3. Pu, I. **2006.** *Fundamental Data Compression*, Butterworth - Heinemann Publications. pp. 163-165.
4. Salomon, D. **2007**. *Data Compression the Complete Reference*, Springer-Verlag London Limited. pp. 337-353.
5. Shi, Y. and Sun, H. **2000**. *Image and Video Compression for Multimedia Engineering*, CRC Press. pp. 36-38.
6. Salomon, D. **2007**. *Variable-Length Codes for Data Compression*, Springer-Verlag London Limited. pp. 9-12.
7. Ibraheem, N. I. **2004**. Image Compression Using Wavelet Transform, M.Sc. thesis, Baghdad University. pp. 80-85.
8. Saha, S. **2004**. *Image Compression – from DCT to Wavelets: A Review*, The Association for Computing Machinery, Inc. pp. 15-16.