Journal of Al-Nahrain University Vol.12 (3), September, 2009, pp.177-183 Science

DIGITAL IMAGE COMPRESSION USING INTEGER BASED TRANSFORM ALGORITHM

Bushra A. Sulttan
Department of Computer, Collage of Science, Baghdad University.
E-mail:bushra542001@yahoo.com.

Abstract

One of the most problems in computer application is the storage and transmission of images that makes the field of developing the image compression. For that, various compression methods have been proposed using different techniques to achieve high compression ratios and high image quality. One of the image compression methods is transformed based image compression using integer wavelet transform (IWT), which is the main purpose of this paper. This paper describes an algorithm of compression and decompression using (IWT). In IWT, we perform all possible decomposition of an image (i.e. until image size 2×2), after that the resultant coefficients are arranged according to their impotency using zigzag arrangement and divided on factor related to their impotency (increase the factor with low impotency), to make low impotency coefficient related to zero. Remap the values to the range (1.255) (i.e. byte). Then perform lossy or lossless Run Length Coding (RLC) to construct the codebook of the compressed image. The effects of different RLC and different initial quantization factor, image contents and compression ratios assessed. The results provide a good reference for application developers to choose a good wavelet compression system for their application.

Keywords: integer wavelet transforms IWT, Run Length Coding RLC, transform based image compression.

1-Introduction

Image compression has been pushed to the forefront of the image processing field this is largely a result of the rapid growth in computer power the corresponding growth in the multimedia market, and the advent of the world wide web (WWW), which makes the internet easily accessible for every one, additionally, the advances in video technology create a demand for new, better and faster image compression algorithms [1]. The typical image contains a large amount of spatial redundancy in plain areas where adjacent picture elements (pixels, pels) have almost the same values. It means that the pixels values are highly correlated. In addition, an image can contain subjective redundancy, which determined by properties of a Human Vision System (HVS).

An HVS presents some tolerance to distribution, depending upon the image content and viewing conditions. Consequentially, pixels no always be reproduced exactly as originated and the HVS will not detect the difference between original image and reproduced image. The redundancy (both statistical and subjective) can be removed to achieve compression of the image data, the basic measure for the performance of a compression algorithm is compression ratio (CR), defines as a ration between original data sized and compressed data size [2].

Based on the difference between original and reconstructed version, data compression schemes can be divided into two broad classes (see Fig.(1): the first class is lossless compression in which the reconstructed image exactly as the original image, the second class is lossy compression techniques involves some loss of information and the data can not be recovered or reconstructed exactly as the original image [4]. In lossy compression scheme, the image compression algorithm should achieve a tradeoff between compression ratio and image quality. Higher compression rations will reduce lower image quality and vise versa. Quality and compression can also vary according to input image characteristics and content [2].

The rest of the paper organized as follows Section 2 describes the transform based image compression. Section 3 describes the IWT. Section 4 describes the RLC. Section 5 describes the proposed algorithm. Section 6 describes the experimental results and finally section seven lists the conclusions.

2-Transform Based Image Compression

The transform coding compression methods consist of two main phases the compression phase and decompression phase. In compression phase, the image transforms from spatial domain to frequency domain and then compressed. In the second phase which is the reconstruction phase the output of the first phase is used to reconstruct the original image by using inverse transform operation on compressed image (Fig.(2))[3].An effective transform will concentrate useful information into a few of low-frequency transform coefficients. An HVS is more sensitive to energy with low spatial frequency than high spatial frequency. Therefore, compression can achieve by quantizing the coefficient, so the important coefficients (low-frequency coefficient) transmitted and the remaining discarded. Very effective and popular way to achieve compression of image data based on the discrete cosine transform (DCT) and discrete wavelet transform (DWT) [2]. The DWT is simple but has an important drawback, namely, it uses non-integer filter coefficients, which results in non-integer transform coefficients. There is several ways to modify the base DWT such that to produce integer transform coefficients named Integer Wavelet Transform (IWT). The transform is reversible, i.e. the image can fully reconstructed from the (integer) transform coefficients. This IWT can use to compress the image either lossly (by quantizing the transform coefficients) or losslessly (by entropy the transform coefficients) [5].

A wavelet image compression system can be created by selecting type of wavelet function (perform IWT of an image), quantizer (the data transformed by the transformation quantized to reduce the amount of information. Quantization is irreversible operation because of its lossy property) and statistical coder (Encoding the result from quantization. This step can be error free by using run length encoding RLE or Huffman coding. It can also be lossy if it optimizes the representation of the information to reduce the bit rate further) [3].

3-Integer Wavelet Transform (IWT)

Wavelet transform (WT) represents an image as a sum of wavelet function (wavelets) with different locations and scales. Any decompression of an image into wavelet involvers a pair of wave forms: one represent high frequencies corresponding to the detailed parts of an image (wavelet function) and one for the low frequencies or smooth part of an image (scaling function)[2].

The principle of this transform is simple and illustrated here for one-dimensional case. Given a data vector of length N integer (X_i), where i=0…N-1, we define k=N/2 and the transform vector (Yi) is computed, by calculating the odd component and even component of y separately. When the length N is even (i.e. N=2K), then the integer transform sequence is computed by implementing the following steps [5]:

1. Determine the odd coefficients:

$$\left.\begin{array}{ll} Y_{2i+1} = X_{2i+1} - \lfloor (X_{2i} + X_{2i+2})/2 \rfloor, & i = 0..K-2 \\ Y_{2i+1} = X_{2i+1} - X_{2i} & i = K-1 \end{array}\right\} \quad \text{..............................(1)}$$

2. Determine the even coefficients:

$$\left.\begin{array}{ll} Y_{2i} = X_{2i} + \lfloor Y_{2i+1}/2 \rfloor, & i=0 \\ Y_{2i} = X_{2i} + \lfloor (Y_{2i-1} + Y_{2i+1})/4 \rfloor & ,i=1..K-1 \end{array}\right\} \quad \text{..............................(2)}$$

Notice that the (IWT) produce the vector Y_i where the detail coefficient and weighted average are interleaved. The algorithm should be modify to place the average in the first half and the details in the second half [5].

Inverse IWT applied to reconstruct the original image. When the length N of the transformed signal Y is even (i.e. N=2K), then the integer transforms sequence X computed in the following two steps [5]:

1. determine the even coefficients:

$$\left.\begin{array}{ll} X_{2i} = Y_{2i} - \lfloor Y_{2i+1}/2 \rfloor, & i=0 \\ X_{2i} = Y_{2i} - \lfloor (Y_{2i-1} + Y_{2i+1})/4 \rfloor & ,i=1..K\text{-}1 \end{array}\right\}. \quad \text{................................(3)}$$

2. determine the odd coefficients:

$$\left.\begin{array}{ll} X_{2i+1} = Y_{2i+1} + \lfloor (X_{2i} + X_{2i+2})/2 \rfloor, & i=0..K\text{-}2 \\ X_{2i+1} = Y_{2i+1} + X_{2i} & i=K\text{-}1 \end{array}\right\}. \quad \text{................................(4)}$$

4-Run Length Coding (RLC)

Run Length Coding RLC is an image compression method works by counting the number of adjacent pixels with same gray level value this count called run length, is coded and stored. The pair (G,L) correspond to the gray level value and the run length L. [1].

5-The Proposed Algorithm

In this section, the proposed developed transform based image compression algorithm is described, the main concerns in this design are:

a. Keep the codebook content just byte (i.e range [0,1,..255]) where the zero is the special character in the codebook file. Special character means the value had been began after it was the occurrences of the value placed in front of it, as will be explain in the following example:
Suppose the data:
1 1 1 1 1 2 2 2 2 2 2 2 4 5 6 6 6 6 6 6
RLC (1,5) (2,7) (4,1) (5,1) (6,6)
The codebook content will be as follows:
1 0 5 2 0 7 4 5 6 0 6

b. Get benefit form integer wavelet transformation and then mapped it to the byte range [1..255].

c. The arrangement of the transformed coefficients according to their impotency to exploit this arrangement in the quantization process.

d. According to the difference between the adjacent pixels the RLC is classified to lossy or lossless (i.e. compute the run length to the pixels with the same gray values (lossless) or the pixels differ in (1 or 2) of gray values (lossy)).

The detail of the proposed algorithm is show in algorithm 1.

Algorithm1:
Inputs: (image file).
Outputs: (codebook file).
Step1: perform IWT on an image with all possible decompositions.
Step2: set the values of the subband HH_1 in the resulted IWT from step 1 to zero
Step3: perform the linear mapping to resulted coefficients to convert it to the range [1..255].
Step4: arrange the resulted mapped coefficients from step3 using Zigzag arrangement (i.e. LLn, LHn, HLn, Hn-1 …….,LH1,HL1). Therefore, sequential file will contain information arrange in descending order from high to low impotency, see Fig. (3).
Step5: quantized the arranged file, dividing it by the factor K. This factor will increase when the impotency decrease, except from the quantizing process the subband LLn
Step6: perform the linear mapping to resulted quantized values to keep it in the range [1.255].
Step7: apply lossless or lossy RLC algorithm on the quantized file to produce the codebook file. To get benefits form the data declaration (byte) we use (0) number was used as special character in the codebook file.

To reconstruct the decompressed image the decompression algorithm will apply, passing to it the codebook file and reverse all the steps in compression algorithm from step 7 down to step1.

To enhance the performance of the proposed system apply the median filter on the reconstructed decompressed image to remove any noise or blur.

6-Experimental Results

To evaluate and enhance the performance of the proposed algorithm, some experiments performed.

Two gray scale images of size (256x256) used in these experiments as uncompressed image. The first “winter” was a rough image (i.e. contains more details) and the second “Lena” was a smooth and flat image.

Experiments: Figs. (4) and (5) shows these experiments where: (a) is the original image (before compression) and (b), applying the proposed compression algorithm where the quantization factor begin from two and increase by one and using lossless RLC. In (c), applying median filter on the resulted reconstructed decompressed image from (b). In (d), apply the algorithm with quantization factor begin from one and increase by on and using lossy RLC. Finally in (e), applying median filter on the resulted reconstructed decompressed from (d).

Form the previous two experiments, mean square error (MSE) used between the original and the reconstructed decompressed images to evaluate the proposed algorithm. MSE is calculated using equation (5) [1].

$$MSE = \frac{\sum_{i=0}^{n-1}\sum_{j=0}^{m-1}\left(p_o(i,j)-p_d(i,j)\right)^2}{m \times n} \quad ......5$$

Where

n: depth of image.

m: width of image.

$P_o(i,j)$: pixel value of original image

$P_d(i,j)$:pixel value of the reconstructed decompressed image.

Table (1) summarizes the results in error measurement (MSE) and CR.

Table (1)
Comparison between the MSE and CR in reconstructed decompressed images and the type of applied RLC.

	Lena		winter	
	CR	MSE	CR	MSE
Lossless RLC	1.41948	0.0101238	1.28990	0.03498
Lossy RLC	6.71475	0.0165425	4.44236	0.04983

7-Conclusions

From the results of the proposed algorithm, one can deduce the following:

a. The implementation of IWT adds some power and flexibility to the compression performance. The coefficients take integer values for both the forward and their inverse transform of the IWT.

b. It is noticed that good CR and un noticeable error in an image when it is smooth and flat also with the rough images.

c. The rounding of the division in all transformation or mapping to keep the value in integer range [1..255] do not reduce the performance of the system.

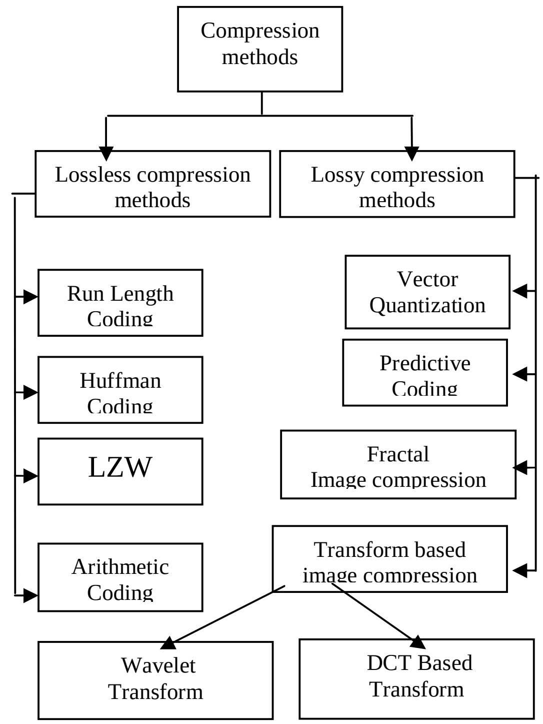
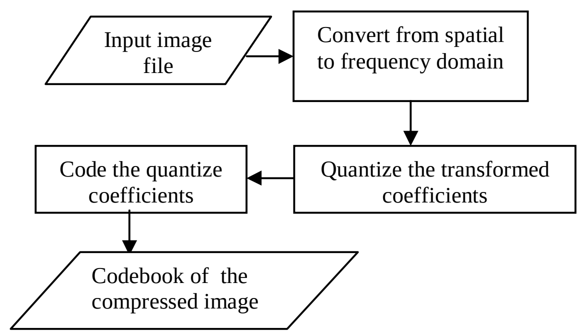
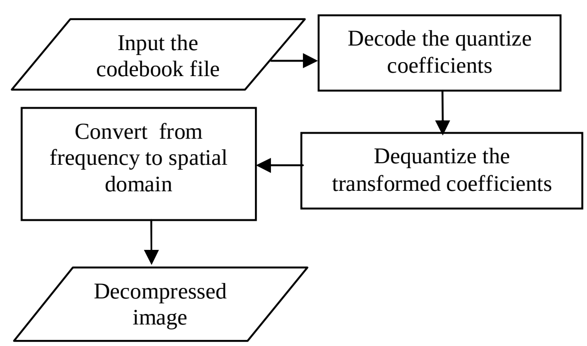


Fig. (1) : The most popular image compression methods.



a. The block diagram of the compression process.



b. The block diagram of the decompression process.

Fig. (2): Transform coding compression and decompression phases.

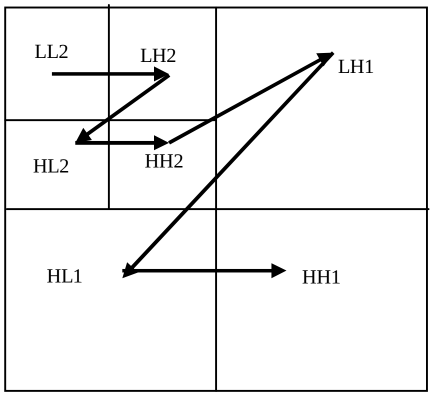


Fig.(3): The Zigzag arrangement of the wavelets coefficients.

c

a

d

b

e

Fig.(4): The Proposed Algorithm Results.

a. Original image (lena).
b. Reconstructed compressed image of (a) using lossless RLC.
c. Filtered image of (b).
d. Reconstructed compressed image of (a) using lossy RLC.
e. Filtered image of (d).

a

b

c

d

e

Fig. (5) : The Proposed Algorithm Results.

a. ***Original image (winter).***
b. ***Reconstructed compressed image of (a) using lossless RLC.***
c. ***Filtered image of (b).***
d. ***Reconstructed compressed image of (a) using lossy RLC.***
e. ***Filtered image of (d).***

References

[1] S.E Umbangh, ***"**Computer Vision and Image Processing**"***, prentice-Hall PTR, Inc, USA, 1998, pp. 237-263.

[2] S. Gregic, M. Gregic, Member IEEE and B. Zovko-Cihlar, Member, IEEE," Performance Analysis of Image Compression Using Wavelets", IEEE Transaction Industrial Electronic, VOL 48, No.3, JUNE 2001, pp. 682-683.

[3] W. S. Ahmed, " The Effects Of Quantization On Image Compression Using Transform Coding", M.Sc. Thesis, Dep. Computer Since, Al-Nahrain Unvirsity,2007,pp. 11-28

[4] J. Addel, "Lossy and Lossless Image Compression", Prentice Hall PTR,2000, pp 21-22.

[5] D. Salaman, "Data Compression the Complete Reference", Addition Wesley Company, Second Edition, 2000, pp. 343-346.

الخلاصة

واحدة من المشاكل الهامة في تطبيقات الحاسوب هو نقل وخزن المعلومات ومنها الصور الرقمية, والتي تطلبت الاهتمام بتطوير تقنيات الضغط. لذلك تم أقتراح طرق مختلفة للظغط بأستخدام عدة تقنيات لتحقيق نسب ضغط عالية وكذلك جوده عالية للصور. تعتبر طريقة الضغط المبنيـــــة علـــى اســــاس التحويــــل المويجي الصحيح (integer wavelet transform IWT) واحدة من هذه الطرق التي هي الطريقة الاساسية في هذا البحث. تم في هذا البحث وصف خوارزمية الضغط وفك الضغط بأستخدام التحويل المويجي الصحيح (IWT). في هذا التحويل المويجي يتم الاستمرار بتحويل الصورة بكل التقسيمات الممكنة الى ان يصبح حجم الصورة (2x2). بعد ذلك المعاملات الناتجة تترتب على اساس اهميتها (الاهم فالاقل اهمية) باستخدام طريقة الترتيب المتعرج (zigzag) ثم تقسم على معامل مرتبط بأهمية هذه المعاملات (يزداد المعامل بنقصان الاهمية) لجعل هذه العاملات غير المهمة تقترب من الصفر. يتم اعادة تعديل القيم الناتجة لتصبح ضمن المدى(255..1) ثم يتم تطبيق الترميز بطريقة (Run Length) لبناء كتاب الترمـــيز (codebook).

تم اختبار تأثير استخدام نوعين من الترميز (الفاقدة للمعلومات وغير الفاقدة للمعلومات) و قيم ابتدائية مختلفة لمعامل التكميم على نوع الصور الناتجة وعلى نسبة الضغط. النتائج اعطت مصادر جيده لمطوري التطبيقات لأختيار التحويل المويجي الجيد لضغط الصور في تطبيقاتهم.