J. of al-anbar university for pure science : Vol.2 : No.1 : 2008

ISSN: **1991-8941**

IWCD-PROPOSED IMAGE COMPRESSION METHOD BASED ON INTEGER WAVELET AND DCT

Abdullah.M.Awad Al-Fahdawi
University of Anbar-College of Computer

ABSTRACT:This paper describes a new lossy image compression decompression algorithm. In lossy compression techniques there are some loss of information, and image cannot be reconstructed exactly.
This algorithm will be referred to as (IWDC), which stands for integer wavelet (IWT) and discrete cosine transform (DCT) and this algorithm improves existing techniques and develops new image compressors.

(IWDC) is efficient than corresponding DCT and wavelet transform functions and incorporating DCT and integer wavelet transform are shown to improve the performance of the DCT and integer wavelet (IWT). In the new proposed compression is more efficient than the still image compression methods.

Keywords: IWCD-Proposed, Image Compression , Integer Wavelet , DCT

Introduction

The main aim of the image compression is to reduce the number of bits, which needed to represent the image.

There are correlation between the neighbors pixels in spatial domain and the more effectively technique is able to exploit the no correlations in the data in frequency domain that will be able to compress that data.

There are many methods for image compression but each method has a different characteristics depend on type of transform, which lead to increase or decrease the time and space.

The integer wavelet transform used for de-correlating image pixels has a very fast integer implementation requires only 5 additions and 2 shifts per image pixel. Image compression methods can be made by different characteristics, and produce different results.[11]Compression Algorithm

The basic motivation behind transform is to transform a set of pixels or samples from the spatial domain into another set of less correlated or more independent coefficients in the frequency domain, so a digital image is transformed by means of integer wavelet transform (IWT) into two lower resolution images a low-frequency and high-frequency images. Integer inputs sequences are transformed into integer output sequences of the same length.

Nonzero DCT coefficients are generally located close to the top/left corner of the transformed block.

Integer Wavelet Transform

The IWT can be used to compress the image either loosely by quantizing the transform coefficients or lossless by entropy coding.

The simple integer wavelet transform (IWT) can be used to decompose an image and the transform is reversible, i.e., the image can be fully reconstructed from the integer transform coefficients.

The principle of this transform is in one-dimensional case.

The vector of N integer xi

Where i=0 , 1 , ………………., N-1.

The case where N is even, the N/2 odd components y2i+1 (where i=0,1,..,k-1) are calculated as differences of the XI's. They become the detail (high frequency) transform coefficients. Each of the even component y2i (where i vary in the same range [0,……,k-1]) is calculated as a weighted average of five data items xi. These N/2 numbers become the low-frequency transform coefficients and are normally transformed again into N/4 low frequency and N/4 high-frequency coefficients.

If the signal computed in the length N is even (i.e. N = 2K), then the integer transform sequence is following two steps, odd coefficients first

The basic rule for the odd transform coefficients is.[4]

$$y_{2i+1} = \begin{cases} x_{2i+1} - [x_{2i} + x_{2i+2}]/2 \\ \quad for\ i = 0,1,...., k-2, \\ x_{2i+1} - x_{2i} \\ \quad\quad for\ i = k-1. \end{cases} .....(2-1)$$

The even transform coefficients are calculated as the weighted average

$$y_{2i} = \begin{cases} x_{2i} + y_{2i+1}/2. \\ \quad\quad for\ i = 0 \\ x_{2i} + [y_{2i-1} + y_{2i+1}]/4 \\ \quad for\ i = 1,2,.., k-1. \end{cases} ......(2-2)$$

If the signal length N is odd (i.e. N = 2K + 1), then the integer transform is computed in the following two steps, odd coefficients first

$$y_{2i+1} = \begin{cases} x_{2i+1} - [x_{2i} + x_{2i+2}]/2 \\ \quad for\ i = 0,1,...., k-1, \end{cases} ....(2-3)$$

And even coefficients second

$$y_{2i} = \begin{cases} x_{2i} + y_{2i+1}/2 \\ \quad\quad for\ i = 0 \\ x_{2i} + [y_{2i-1} + y_{2i+1}]/4 \\ \quad for\ i = 1,2,...., k-1. \end{cases} ....(2-4)$$

$$y_{2i} = \begin{cases} x_{2i} - [y_{2i-1}/2] \\ \quad for\ \ i = k \end{cases} ....(2-5)$$

This step prepares the image for the next step. [4]

The inverse transform is easy to figure out. It uses the transform coefficients yi to calculate data items zi that are identical to the original xi.

It first computes the even elements. If the length N of transform signal y is even (i.e. N = 2K'), then the inverse integer transform sequence x is computed in the following two steps, even coefficients first.

$$x_{2i} = \begin{cases} y_{2i} - [y_{2i+1}/2] \\ \quad\quad for\ i = 0 \\ y_{2i} - [y_{2i-1} + y_{2i+1}]/4 \\ \quad for\ i = 1,2,...., k-1. \end{cases} ....(2-6)$$

And odd coefficients second

$$x_{2i+1} = \begin{cases} y_{2i+1} + [x_{2i} + x_{2i+2}]/2 \\ \quad for\ i = 0,1,...., k-2, \\ y_{2i+1} - x_{2i}, \\ \quad\quad for\ i = k-1. \end{cases} ....(2-7)$$

The transform coefficients calculated by Equations (2-1) and (2-2) are not generally integer, because of the divisions by 2 and 4. The same is true for the reconstructed data items of Equations (2-4 and (2-5). The main feature of the particular IWT described here is the use of transaction. Truncation is used to produce integer transform coefficients yi and also integer reconstructed data items zi. Equations (2-1) through (2-5) are modified too. If the length N of

transform signal y is odd (i.e. N = 2K + 1), then the inverse integer transform sequence x is computed in the following two steps, even coefficients first.[4]

$$x_{2i} = \begin{cases} y_{2i} - [y_{2i+1}/2] \\ \quad for\ i = 0, \\ y_{2i} - [(y_{2i-1} + y_{2i+1})/4] \\ \quad for\ i = 1,2,....,\ k-1. \end{cases} .... (2-8)$$

$$x_{2i} = \begin{cases} y_{2i} - [y_{2i-1}/2] \\ for\ \ i = k \end{cases} .... \ ..(2-9)$$

And odd coefficients second

$$x_{2i+1} = \begin{cases} y_{2i+1} + [x_{2i-1} + x_{2i+1}/2] \\ for\ i = 0.....\ k-1 \end{cases} ....(2-10)$$

Figure 1.Lena Image transformed with one pass IWT

TheDiscreteCosine Transform.

DCT breaks the source image into N x N matrix. In practice, Implementations typically break the image down into more manageable 8 x 8 blocks then we apply the discrete cosine transform on the matrix. The mathematical function for a two-dimensional DCT is:

$$y(k,l) = \frac{c(k)c(l)}{4}\sum_{i=0}^{7}\sum_{j=0}^{7}x(i,j)\cos\left\{\frac{(2i+1)kp}{16}\right\}\cos\left\{\frac{(2j+1)lp}{16}\right\}...(2-9)$$

Where l,k=0……7 and
c(l) , c(k)=1 for l,k<> 0 and
c(l) , c(k)=1 /sqrt(2) for l,k=0

The function of Inverse DCT (IDCT) can be re-written as below:

$$x(k,l) = \sum_{i=0}^{7}\sum_{j=0}^{7}y(i,j)\frac{c(k)c(l)}{4}\cos\left\{\frac{(2i+1)kp}{16}\right\}\cos\left\{\frac{(2j+1)lp}{16}\right\}....(2-10)$$

Where l,k=0……7 and
c(l) , c(k)=1 for l,k<> 0 and
c(l) , c(k)=1 /sqrt(2) for l,k=0

For an example below, the input 8 x 8 matrix from a red scale image consists of pixel values which are randomly spread around 16 to 147 range. These values are fed to the DCT algorithm, creating the output matrix below.[1]

Input matrix

65	30	18	17	34	63	62	56
66	28	20	16	34	58	94	85
42	35	28	17	16	31	38	86
37	27	15	19	29	67	58	35
48	19	21	15	11	17	56	27
49	19	17	18	14	16	67	28
21	24	16	19	61	90	40	50
25	24	24	42	60	27	39	147

Table 1. 8*8 blocks of image data

Output matrix

308	-90	81	30	14	7	10	5
12	13	30	25	-22	24	2	0
52	-40	6	-11	32	-35	12	-5
-26	35	-9	40	-30	36	-18	11
0	3	-4	-9	11	-5	-2	-4
-6	2	-40	35	-23	-21	16	-19
-19	15	20	-40	22	4	-4	3
-15	9	-5	-11	5	-14	13	-3

Table 2. 8*8 blocks of image data applied with 2- dimensional DCT

The output matrix consists of DCT Coefficients, which is ordered in a way that coefficients containing useful and important data for representation of the image are in the upper left of the matrix and in the lower right

coefficients containing less useful information.

The transform is rarely applied to the whole image; it usually deals with small regions or image blocks independently.

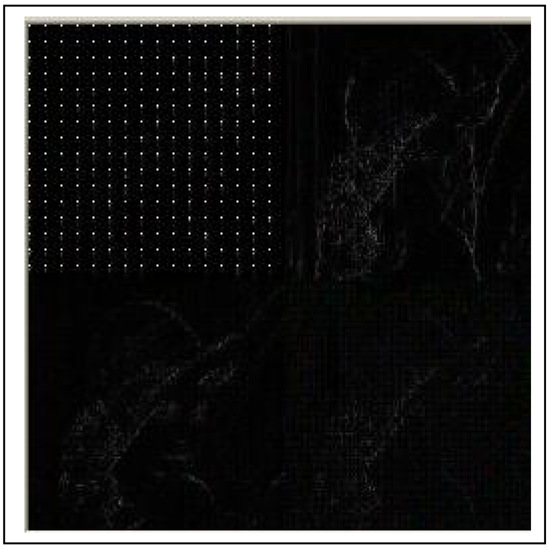

Figure 2.Lena Image transformed with 2-dimention DCT

Quantizations

Quantization means the process of representing a large possibly infinite set of values with a much smaller set. After the FDCT is computed, each of the 64 coefficients is quantized in conjunction with a 64-element quantization table.

ISO has developed a standard set of quantization values that cause impressive degrees of compression.

To calculate the quantization matrix we use this formula:

$$Q(i,\ j) = 1 + (1 + i + \ j) * Q.f \text{......}(\ 2 - 11)$$

Quantization is defined as division of each DCT coefficient by its corresponding quantize step size followed by rounding to the nearest integer, i.e. the purpose of quantization is to achieve further compression by discarding information which is not visually significant or high spatial frequency information.

3- (Integer Wavelets and DCT) Image Compression

This proposed algorithm used to apply (INTEGER and DCT) transform in image compression that after we apply the Integer transform on image, the values in the first quarter of image are used only to apply the DCT and all the values in (LH,HL and HH) are quantized The DCT transform is applied to this quarter. It means apply this transform to every image pixel. These coefficients represent the information varying from pixel to pixel.

The image that is to be compressed is processed in 8x8 blocks. The DCT is applied to each of these blocks. The coefficient set that is obtained has to be quantized; this means the scaling of each coefficient block with a known matrix, so the numeric value of these coefficients is reduced. The real compression process is realized by truncating these new values after the quantization process. This truncating process also produces the loss of information.

After scaling of coefficients, many of these will be zero, that meaning they are no more important. The rest of coefficients represent the image in compressed form. The compression rate obtained using this method depends on the values of the elements of the quantization matrix. As much as these elements have high values, we will obtain more coefficients with values equals to zero, so the compression rate will be higher. This will increase also the loss of information in compression process.

The coefficients of image array are reordered from 2-dimensional array to one dimensional array for the values of the coefficient matrix and RLE are applied for the zero's values and the shift-coding are applied to get image data in stream of bits in a file.

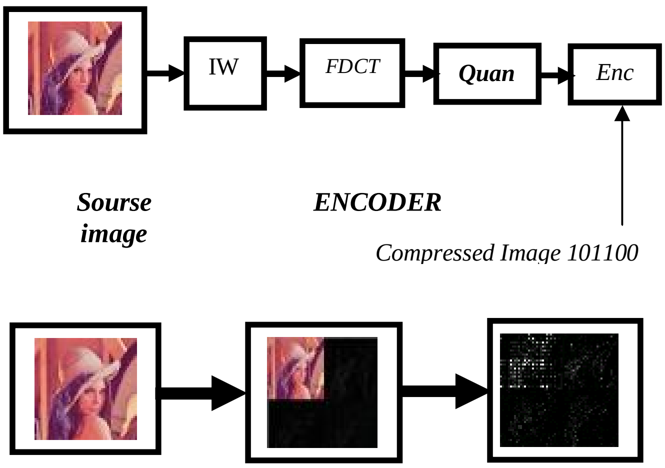

Figure 3.IWCD Encoder block diagram

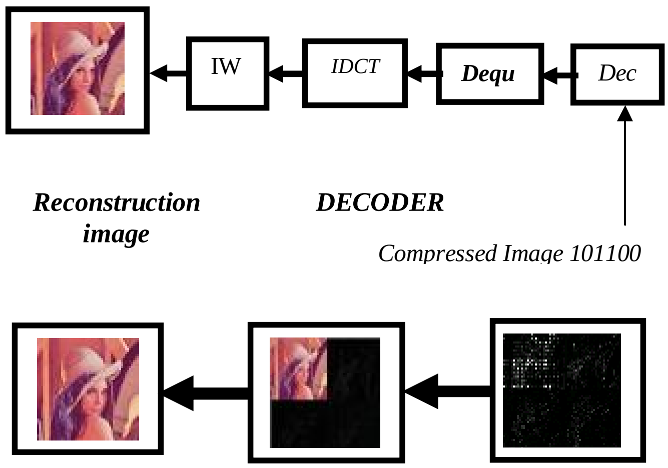

Figure 4.IWCD Decoder block diagram

The reconstruction of the original image is realized with the reversed algorithm.
First, the coefficient from the binary file are received, the shift-decoding are applied to get image data in decimal representation and decoding RLE for the zero's value the coefficients of the image array are reordered from 1-dimensional array to 2-dimensional array.

Now the matrix representing the compressed image is multiplied by the quantization matrix, resulting a matrix corresponding to the image, then inverse DCT and inverse IWT are applied to this matrix. Now we get a data matrix, the elements of this matrix are a little different versus the elements corresponding to the original image, due to the information loss that appears in compression process. These losses can be seen in the differences between the original and the compressed image.

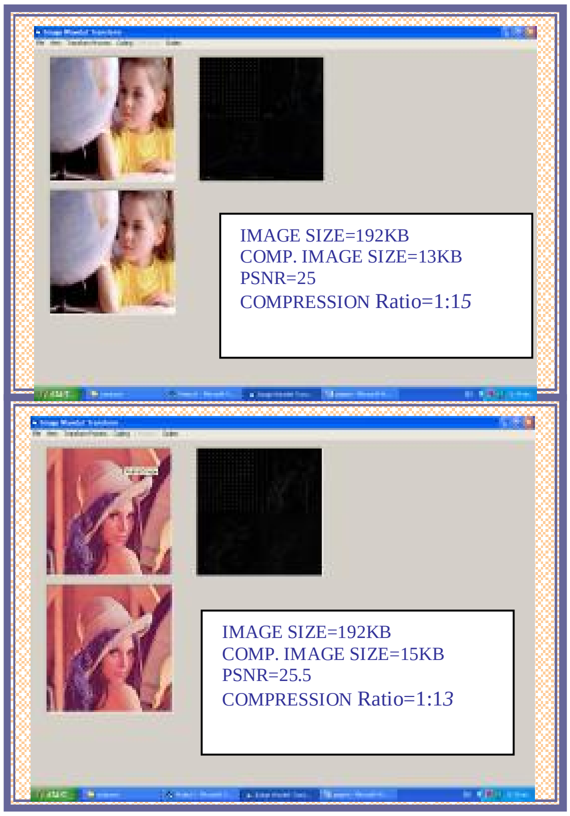

Figure 5.Compressed Images by IWCD

Discussion

Two of the error metrics used to compare the various image compression techniques are the Mean Square Error (MSE) and the Peak Signal to Noise Ratio (PSNR). The MSE is the cumulative squared error between the compressed and the original image, whereas PSNR is a measure of the peak error. The mathematical formulae are:

MSE

$$= \frac{1}{MN} \sum_{y=1}^{M} \sum_{x=1}^{N} \left[I(x,y) - I'(x,y) \right]^2$$

PSNR = 20 * log10 (Imax (x, y) / sqrt(MSE))

Where I(x, y) is the original image, I'(x,y) is the approximated version (which is actually the decompressed image) and M,N are the dimensions of the images.

A lower value for MSE means lesser error, and as seen from the inverse relation between the MSE and PSNR, this translates to a high value of PSNR. Logically, a higher value of PSNR is good because it means that the ratio of Signal to Noise is higher. Here, the 'signal' is the original image, and the 'noise' is the error in reconstruction. So, if we find a compression scheme having a lower MSE (and a high PSNR), we can recognize that it is a better one.

Table 3. Compression Factor and PSNR

Image Name	Image File size	Comp. File size	PSNR (db)	Comp factor
Lena	192 KB	13 KB	25	15
Child	192 KB	15 KB	25.5	3
Baboon	192 KB	15 KB	18	3

PSNR is the most commonly used value to evaluate the objective image compression quality.

We have implemented a progressive IWT and DCT based coder and demonstrated good results. Our results show that the DCT remains a competitive compression technology when used with good quantization strategies, so that full advantage can be taken of the significant investment that has occurred in fast DCT algorithms.

The experimental results for this new algorithm show that the algorithm using IWCD can achieve a much higher compression ratio than image compression methods using only DCT or integer Wavelet transforms.

There are several important points to notice in this comparison.

1-The DCT is one of efficient method for image compression but it needs the longer time.

2-IWCD is the best compression ratio and it needs the time less than DCT but this time more than Integer Wavelet transforms and this time is acceptable if we look from compression ratio view.

REFERENCES

[1]. David Bethel,"Optimization of Still Image Compression Techniques". PHD Thesis, 1997, University of path.

[2]. Nelson, Mark. "The Data Compression Book" .P. 347-373/M&T Publishing 1991.

[3]. Scott E Umbaugh. " Computer Vision and Image Processing", Prentice Hall NJ, 1998.

[4]. David Salmon, "Data Compression ". The Couplet Reference Prentice Hall, Third Edition-2003.

[5]. Eric E Majani," Algorithm for Lossy /Progressive Loss less Image Compression", NASA Tech. Brief Vol. -21.NO.12, Item # 146, Dec 1997.

[6]. Panrong Xiao. "Image Compression by Wavelet Transform". Master Thesis, August 2001, East Tennessee State University.

[7] K. Sayood, "Introduction to Data ompression",Second edition, Academic Press, 2000.

[8] C.S Burrus R.A. Gopinath, and H.Guo. "Introduction to Wavelets and Wavelet Transforms", Englewood Cliffs, NJ: Prentice Hall, 1998.

[9] H. GU, "Image Compression Using the Haar Wavelet Transform", Masters Thesis, East Tennessee State University, 2000.

[10] G. Strang, "The Discrete Cosine Transform", Society for Industrial and Applied Mathematics, 1999.

[11] P. Morton and A. Petersen, "Image Compression Using the Haar Wavelet Transform", College of the Redwoods, 1997.

IWCD طريقة مقترحة لكبس الصور الملونة باستخدام تحويلات Integer Wavelet و DCT

عبدالله محمد عواد الفهداوي
E-Mail: abalawad2000@yahoo.com

الخلاصة:

هذه الدراسة تصف طريقة جديدة لكبس الصور الملونة والتي هي بامتداد(BMP)و هـذه الخوارزميـة تعتمد على مبدأ فقدان جزء من البيانات التي يعتبر فقدانها غير مؤثر عمليا على نوعية الصورة و قـد سـميت (IWDC) لأنها تعتمد تحويلات Discrete) Cosine Transform و Wavelet) من المعروف بان الكبس باستخدام تحويلات Discrete Cosine Transform يعطي نتائج عالية في عملية الكبس و لكن الفترة الزمنية اللازمة للتحويل عالية و من جانب أخر فان الفترة اللازمة للتحويل فيWavelet Transform اقـل كثيـرا ولكن نسبة الكبس اقل كذلك والخوارزمية المقترحة يتم الاستفادة فيها من محاسن الطريقتين اعلاة