DCT Transform and Wavelet Transform in Image Compression: Applications

Corban Radu, Ungureanu Iulian
Technical University of Iasi, Iasi, Romania, Email: rcorban@etc.tuiasi.ro, uiulian@usa.net

In image processing, an important part is the compression. This means the reducing the dimentions of the images, to a level that can be easily used or processed. There are many methods used for realizing a compression process, and the choice of any method depends on user's requests, reffering to algorithmic complexity of the compression process, the quality of the image obtained after compression, and the compression rate. In this paper is presented a comparative study of the most used compression methods: the Wavelet transform, and the DCT transform.

Keywords: image compression, Wavelet transform, DCT transform, algorithmic complexity, compression rate.

INTRODUCTION

The compression is a process that should realize a compact digital representation of a signal. If the data is an audio, video or image signal, the compression problem is to minimize the bit rate of the digital representation of the signal. In the most applications, it's good the data to be available in compressed form; or else, these applications could not be feasible.

The audio, video and image signals are available to be compressed, because the next factors [1]:

1. There is a considerable redundancy in these signals:
   - There is a spatial correlation between the next samples in a signal;
   - For data acquired from many sensors, there is a spectral correlation between the samples from different sensors;
   - For data as video signals, exists a temporal correlation between the samples.
2. There is a significant quantity of information in the signals that is irrelevant from a perceptual point of view.

In the Fig. 1 is presented the scheme of a compression system:

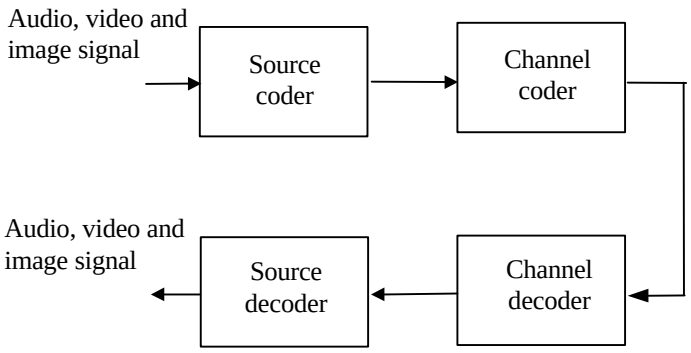

Fig. 1. General scheme for a compression system

From a design point of view, the compression problem can be seen as a problem of minimization the bit rate, including several constraints, as the signal quality level, the implementing (or algorithmic) complexity and the communications delays.

DCT BASED CODING

The DCT based coding is the base for all image and video compression standards. In a DCT based system, the basic computing is the translation of an image block with a NxN dimension (in pixels), from the spatial domain in the DCT domain. In the compression standards, N=8. This number is chosen because, from an implementation point of view, an 8x8 image block does not need special requests of memory; moreover, algorithmic complexity for such a block is feasible on the most computing platforms. From the compression rate point of view, if we use an N bigger than 8, we will not obtain significant improvements.

DCT is used because next reasons:

- For high correlated data, the compression rate obtained by DCT is getting close to that obtained using the optimum Karhunen – Loeve transform.
- DCT is an orthogonal transform. So, if in a matrix form, the DCT output is $Y=TXT^t$, then the inverse transform is $X=T^tYT$. The $X \rightarrow Y$ is named the direct DCT, and is given by [1]

$$y_{kl} = \frac{c_k c_l}{4} \sum_{i=0}^{7} \sum_{j=0}^{7} x_{ij} \cos\left(\frac{(2i+1)k\pi}{16}\right) \cos\left(\frac{(2j+1)l\pi}{16}\right) \quad (1)$$

where k, l=0, …, 7 and $c_k = \begin{cases} \frac{1}{\sqrt{2}}, & k = 0 \\ 0, & k \neq 0. \end{cases}$

DCT can be written in matrix form, as $y=Tx$, where $x=\{x_{00},\ldots, x_{07},x_{10},\ldots,x_{17},\ldots, x_{77}\}$, and T is a matrix, whose elements are the products of the cosine functions defined before.

The inverse DCT transform can be written as:

$$x_{ij} = \sum_{k=0}^{7} \sum_{l=0}^{7} y_{kl} \frac{c(k)c(l)}{4} \cos\left(\frac{(2i+1)k\pi}{16}\right) \cos\left(\frac{(2j+1)l\pi}{16}\right) \quad (2)$$

An important feature of 2-D DCT and of 2-D IDCT is separability. This means these 2 bidimensional transforms, written in matrix form, can be computed by performing 1-D DCT first on the rows, then on the columns of this matrix.

The 1-D DCT is:

$$z_k = \frac{c(k)}{2} \sum_{i=0}^{7} x_i \cos\left(\frac{(2i+1)k\pi}{16}\right) \quad (3)$$

This equation can be written in matrix form as $z=TX$, where T is an 8x8 matrix, that have its elements equal to the cosine functions defined before; $x=\{x_0,\ldots, x_7\}$ is a row matrix, and z is a column matrix.

Let be $z_{il} = \frac{c(k)}{2} \sum_{j=0}^{7} x_{ij} \cos\left(\frac{(2j+1)l\pi}{16}\right)$ (4)

the result of the 1-D DCT on x_{ij} rows. The previous equations suppose that the 2-D DCT can be obtained by performing the 1-D DCT on x_{ij} rows, then performing the 1-D DCT on z_{il} columns. As matrix notation, $Y=TXT^t$ can be seen as $Z=TX^t$; $Y=TZ^t$.

From an implementation point of view, this row-columns computing solution can simplify the hardware necessities, the price being an easy growth of the total number of performed operations.

WAVELET REPRESENTATION

Recent years, many scientists contributed to the development of Wavelet theories [2] and their applications. The Wavelet functions are “local” functions. In applications, the Wavelet based functions are located over the fluctuations of some signal analysis, and they provide a local analysis for the signal. A nonlinear arbitrary function can be represented as a Wavelet series. The Wavelet coefficients can be determined analytically – for some classes of Wavelet basis functions, and for some signal types – , or can be calculated . Using classic methods, the calculations can be very long. For the applications, the time needed by these calculations has to be reduces as much as possible.

WAVELET FUNCTIONS

Let be a mother Wavelet function $\Psi(t) = 0$, la $t = \pm\infty$. A basic Wavelet function is defined [2] as $\Psi_{a,b} = \frac{1}{\sqrt{a}}\Psi\left(\frac{t-b}{a}\right)$. *a* is named scaling parameter, and represents the dilation of the basic function; *b* is named shift parameter, and represents the function position. A function f(t) is expressed using a basic Wavelet function as $f(t) = \sum_a \sum_b W_{a,b}\Psi_{a,b}(t)$. where $W_{a,b}$ is called Wavelet coefficient.

For a Wavelet analysis, a Wavelet function has to provide 3 important features: admisibility, needed by the inverse transform; orthogonality, that is necessary to obtained the Wavelet coefficients analitically, and compactity (function have to be defined on a finite domain). If the compactity condition is satisfied, the total number of operations needed to compute all the Wavelet coefficients is getting reduced.

THE DWT

A class of Wavelets functions is specified by a number of Wavelet filter coefficients. We will present only the Wavelet filters included in Daubechies class. The most simple and located member of this class is called DAUB4, and it has only 4 coefficients c_0, c_1, c_2, c_3.

Let us consider the next transform matrix [3]:

$$\begin{bmatrix}
c_0 & c_1 & c_2 & c_3 & & & & & & & & \\
c_3 & -c_2 & c_1 & -c_0 & & & & & & & & \\
\cdot & \cdot & c_0 & c_1 & c_2 & c_3 & & & & & & \\
\cdot & \cdot & c_3 & -c_2 & c_1 & -c_0 & & & & & & \\
\cdot & \cdot & & & & & \cdot & & & & & \\
\cdot & \cdot & & & & & & c_0 & c_1 & c_2 & c_3 & \\
\cdot & \cdot & & & & & & c_3 & -c_2 & c_1 & -c_0 & \\
c_2 & c_3 & & & & & & & & c_0 & -c_1 & \\
c_1 & -c_0 & & & & & & & c_3 & -c_2 & &
\end{bmatrix} \quad (5)$$

DWT means the successive application of a Wavelet matrix (as 5), first to the whole N-_length vector, then to the N /2 length smooth-vector, and so one, until it remains a small number (regularly 2) of smooth components. This algorithm is called the pyramidal algorithm. The DWT output means the remaining component, plus all the detailing components, accumulated along the algorithm. The next diagram will explain the algorithm:

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \\ y_9 \\ y_{10} \\ y_{11} \\ y_{12} \\ y_{13} \\ y_{14} \\ y_{15} \\ y_{16} \end{bmatrix} \xrightarrow{3.1} \begin{bmatrix} S_1 \\ d_1 \\ S_2 \\ d_2 \\ S_3 \\ d_3 \\ S_4 \\ d_4 \\ S_5 \\ d_5 \\ S_6 \\ d_6 \\ S_7 \\ d_7 \\ S_8 \\ d_8 \end{bmatrix} \xrightarrow{permute} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ S_4 \\ S_5 \\ S_6 \\ S_7 \\ \underline{S_8} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \xrightarrow{3.1} \begin{bmatrix} S_1 \\ D_1 \\ S_2 \\ D_2 \\ S_3 \\ D_3 \\ S_4 \\ \underline{D_4} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \xrightarrow{permute} \begin{bmatrix} S_1 \\ S_2 \\ S_3 \\ \underline{S_4} \\ D_1 \\ D_2 \\ D_3 \\ \underline{D_4} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \xrightarrow{etc} \begin{bmatrix} S_1 \\ \underline{S_2} \\ D_1 \\ \underline{D_2} \\ D_1 \\ D_2 \\ D_3 \\ \underline{D_4} \\ d_1 \\ d_2 \\ d_3 \\ d_4 \\ d_5 \\ d_6 \\ d_7 \\ d_8 \end{bmatrix} \tag{6}$$

The final result will always be a vector with S values, and a values hierarchy D, d etc. Once generated the d values, these are propagating in all next steps of the algorithm.

It is observed that the whole DWT is an orthogonal linear operator.

For the inverse DCT, we will reverse the algorithm, beginning with the smallest level in hierarchy, working from the right to the left. We will use the transpose of the matrix (5).

THE METHOD

In order to realize a practically comparison of these 2 compression methods, we have developed 2 image compression programs, using, for each of them, one of the 2 methods. These 2 programs have been realized using the Microsoft Visual C++6.0 compiler.

DCT compression

Applying the DCT to a image, in order to compress it, means to apply this transform to every image pixel, the result being a Fourier coefficients set. These coefficients represent the information varying speed, 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. In fact, 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 reconstruction of the original image is realized with the reversed algorithm. First, the matrix representing the compressed image is multiplied by the quantization matrix, resulting a matrix corresponding to the image. 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.

Wavelet compression

In order to realize a compression process using Wavelet transform, we use a matrix like that from eq. 5. The elements of this matrix are the DAUB4 coefficients. The unwritten values are equal to zero.

Each 8x8 image block is multiplicated with the transform matrix, the result being a matrix that is containing Wavelet coefficients.

The compression process is realized by quantizing the Wavelet coefficients matrix. Then, choosing a threshold value, all the coefficients that are smaller than this threshold are equalized to zero. It's obviously that the compression rate is depending on this threshold value, and also is the information loss.

For obtaining the compressed image, we have to apply the inverse Wavelet transform, that meaning to multiplicate the Wavelet coefficient matrix with the transpose of the matrix (5). Due to the losses, it can be seen some differences between the original image and the compressed one.

In figure 2, you can see the differences between the original image and the images compressed with that 2 methods.

Fig. 2. a) Original image

Fig. 2. b) Image compressed with DCT
Compression rate 50%

Fig. 2. c) Image compressed with Wavelet transform
Compressed rate 50%

Conclusions

The algorithmic complexity of the 2 methods

From Eq. 1 results that for computing only one coefficient DCT in an 8x8 image block, we need to perform 64 multiplications, and 64 addition. This means, for the whole block 8x8 we need to perform 4096 multiplications /additions.

If we use an algorithm 1-D, performing the calculations first on the rows, then on the columns of an 8x8 block, we will need to perform only 1024 operations. There are scientists who developed algorithms that need a smaller number of operations [Chan, Lee].

On the other side, using the Wavelet transform, the algorithmic complexity is reducing pretty much. To perform the operations for one single Wavelet coefficient, it is need just 4 multiplications and 4 additions. This means, for an 8x8 image block, only 512 operations, just a half of that are needed to perform the DCT.

Wavelet transform and DCT are both linear functions, that generate a data structure that contains $\log_2 n$ segments with variable length; usually, these segments are presented as a 2^n length vector with different elements. The mathematical properties of the matrix used by these transforms are similar. The inverse transforms FFT and DWT matrix are the transposed original matrix. As result, both transforms can be seen as a rotation in the function space, in different domains. For FFT this new domain contains the basic functions sine and cosine. For the Wavelet, the new domain contains basic functions more complicated, called Wavelet functions, mother Wavelet, or analysis Wavelet.

The basic functions are defined in frequency, representing a category of mathematical tools, as the power spectrum.

On the other side, as a difference versus Fourier transform, the Wavelet transform is defined in space. This makes many functions and operators that are using the Wavelet transform to get “sparsed”, when they are translated from the space domain in the frequency domain. This “sparsening” is very useful in many applications, as signal detection or noise filtering [4, 5].

A way to see the differences in space and time between the 2 transform is to analyze the spaces covered by these functions, in the space – time plane, as in next figure.

It’s well to remember that the Wavelet transform have not just a single set of basic functions, like the Fourier transform, that uses only sine and cosine as basic functions, but presents an infinite set of basic functions. So, the Wavelet transform offers an immediately access to information, access that could not be obtained using other methods, like Fourier transform.

References:

1.Bhaskaran, V., Konstantinides, K., "Image and video compression standards", Kluwer Academic Publishers, USA, 2000

2.Uchino, E., Samatsu, T., Yamakawa, T., "Wavelet network with convex wavelets: applications to nonlinear systems modeling and feature extraction in vector cardiography", *Soft computing in human related sciences*, Teodorescu, H. N., Kandel, A., Jain, L. C., CRC Press, USA, 1999

3.Press, W. H., Teukolski, S. A., Vetterling, W. T., Flannery, B. P., "Numerical recipes in C", Cambridge University Press, 1992, 2-nd edition

4.Wachowiak, M. P., Rash, G. S., Quesada, P. M., Desoky, A. H., "Wavelet – Based Noise Removal for Biomechanical Signals: A Case Study", *IEEE Trans. on Biomedical Engineering*, Vol. 47, no. 3, 2000, pp. 360-369

5.Kadambe, S., Murray, R., Boudreaux – Bartels, G. F., "Wavelet Transform Based QRS Detector", *IEEE Trans. on Biomedical Engineering*, Vol. 46, no. 7, 1999, pp. 838-848