Mathematics and Computers in Simulation 27 (1985) 599–608
North-Holland

IMAGE DATA COMPRESSION BY THE DISCRETE COSINE TRANSFORM

Vito CAPPELLINI and Enrico DEL RE

Dipartimento di Ingegneria Elettronica, Università di Firenze, Firenze, Italy

The application of the Discrete Cosine Transform to image data compression is presented. In particular a fast algorithm is described for an efficient implementation of the transform. Both zonal and threshold sampling methods are considered and their performance is evaluated. Also a comparison of the performance of the Discrete Cosine Transform and the Fast Fourier Transform is outlined. Quantitative and qualitative results are presented for evaluating the various alternatives for the application of the Discrete Cosine Transform. The results have shown that a convenient choice is the threshold sampling method applied to transformed blocks of size ranging from 8 x 8 to 32 x 32.

INTRODUCTION

The inherent large amount of data carried by the images requires efficient techniques to reduce the total number of bits necessary to represent the images without information loss or, more practically, with an acceptable and controlled level of information loss (image distortion).

The operation of reduction of the amount of data is called data compression. The operation of image data compression applies to many and very different areas of practical interest: communications, remote sensing, bioengineering, robotics, artificial intelligence and so on. Data compression can be carried out before storing the image data, in order to reduce the required memory size or, alternatively, it can be performed before transmitting the image data to the receiving terminal side. In this way, a reduction of the transmission rate and, consequently, of the required transmission bandwidth, of an image is also achieved. Of course, this solution calls for an increase in the processing capabilities at the transmitting side. However, the present trend of the VLSI (very-large-scale-integration) digital circuit technology suggests this approach as a technically feasible solution, even in the very near future.

Many methods are known in the literature for image data compression. In [1] for example, the most important approaches to image data compression are reviewed and an extensive bibliography is also reported. Basically, three different approaches are available for image data compression [1],[2]:
- methods based on the image processing in the spatial domain (predictors, interpolators, DPCM, DM);

0378-4754/85/$3.30

- methods based on a preliminary mapping of the image data in a transform domain and on a successive suitable selection of the transformed data (transform coding, e.g., based on the Karhunen-Loeve, Discrete Fourier, Hadamard, Haar, etc., transforms);
- hybrid methods employing a combination of transform and predictive coding (e.g. transforming the image data with respect to one coordinate and then applying one of the first-class methods to the other coordinate of the data obtained after the previous transformation).

Generally, transform coding for image data compression achieves better results in terms of a higher reduction of the total amount of data and of a smaller reconstruction error of the data after the compression operation. On the other hand, it generally requires an increased computational and implementation complexity with respect to the other methods. However, the efficiency of transform coding image data compression and the present trend of the VLSI digital circuit technology suggest this method as a possible candidate for future image data compression systems.

Many unitary transforms have been proposed and applied (e.g. the Discrete Fourier Transform, the Hadamard Transform, the Haar Transform, etc.), each approximating to a different degree the behaviour of the optimal Karhunen-Loeve Transform. It has been also shown [1],[2] that the Discrete Cosine Transform (DCT) closely behaves like the Karhunen-Loeve Transform for highly correlated images.

This paper considers the application of the DCT to images of practical interest, that in general has a high degree of spatial correlation. In particular, a fast algorithm, based on an appropriate use of the Fast Fourier Transform (FFT) algorithm, has been used for an efficient computation of the DCT. Different solutions have been studied and will be compared in terms of the overall efficiency and the computational load.

BIDIMENSIONAL DCT

The bidimensional (2D) Discrete Cosine Transform of a bidimensional sequence (image) $x(n,m), n,m = 0,\dots,N-1$, is defined as [2],[3]

$$C(\ell,k) = \frac{4b(\ell)\,b(k)}{N^2} \sum_{n=0}^{N-1} \sum_{m=0}^{N-1} x(n,m) \cos\frac{\pi(2n+1)\ell}{2N} \cos\frac{\pi(2m+1)k}{2N} \qquad (1)$$

$$\ell,k=0,\dots,N-1$$

with

$$b(p) = 1 \qquad \text{for } p = \ell,k \neq 0$$

$$b(p) = 1/\sqrt{2} \qquad \text{for } p = \ell,k = 0$$

The original image can be recovered by the Inverse Discrete Cosine Transform (IDCT) through the relation

$$x(n,m) = \frac{1}{N^2} \sum_{\ell=0}^{N-1} \sum_{k=0}^{N-1} b(\ell)\,b(k)\,C(\ell,k) \cos\frac{\pi(2n+1)\ell}{2N} \cos\frac{\pi(2m+1)k}{2N} \qquad (2)$$

$$n,m=0,\dots,N-1$$

The direct implementation of the DCT and IDCT by means of the relations (1) and (2) would be very time-consuming and highly inefficient. A computational improvement could be obtained by noting that the DCT and IDCT can be derived through the application of the Discrete Fourier Transform (DFT) to a symmetrical extension of the original sequence x(n,m) in the two coordinates. This approach would lead to the computation of DFTs (by means, of course, of the Fast Fourier Transform algorithm) of size 2N by 2N. In spite of the transform size increase, the use of the FFT algorithms would greatly reduce the computational complexity of this solution with respect to the straightforward implementation of (1) and (2). However, a far better implementation of the DCT and IDCT is possible by using FFT algorithms of size N by N [2],[4],[5]. In the following section a fast procedure for the implementation of a DCT (and an IDCT) of size N by N by means of a FFT algorithm of the same size will be outlined. The formal justification of this procedure can be found in the cited references.

FAST IMPLEMENTATION OF THE DCT AND IDCT

First we note that the relations (1) and (2) of the 2D DCT and IDCT, respectively, define a transformation with a separable kernel. Therefore their implementation can be equivalently realized by transforming first along one coordinate (e.g. rows) through a one-dimensional (1D) DCT (or IDCT) and then along the other coordinate (e.g. columns) again through a 1D DCT (or IDCT). Consequently, a fast implementation of the 2D DCT and IDCT may be obtained through a fast implementation of the 1D DCT and IDCT. In the following, the procedure for a fast implementation of the 1D DCT and IDCT by means of the FFT algorithm will be described.

Fast DCT (FCT)

Consider a 1D sequence x(n),n=0,...,N-1, with N a power of 2, for example the generic row of the image to be transformed. Then:

a) form the sequence v(n) defined as

$$v(n) = \begin{cases} x(2n) & n = 0,\ldots,\left\lceil \frac{N-1}{2} \right\rceil \\ x(2N-2n-1) & n = \left\lceil \frac{N+1}{2} \right\rceil,\ldots,N-1 \end{cases} \tag{3}$$

where $\lceil x \rceil$ denotes the 'integer part of x'. In other words, the sequence v(n) is obtained through a reordering of the sequence x(n): first the even-numbered samples of x(n) followed by the odd-numbered samples in a reverse order;

b) compute the DFT of v(n)

$$V(k) = \sum_{n=0}^{N-1} v(n)\, W_N^{nk} \quad , \; k=0,\ldots,N-1 \tag{4}$$

(where $W_N = e^{-j2\pi/N}$ as usual) by using the FFT algorithm;

c) evaluate the DCT C(k) of x(n) through the relation

$$C(k) = \mathrm{Re}\{2\, W_{4N}^{k}\, V(k)\} \tag{5}$$

where Re{x} denotes the real part of the complex quantity x.

Fast IDCT (IFCT)

The original sequence x(n) is obtained from its DCT C(k) through the following steps:

a) form the DFT V(k) of the sequence v(n) through the relation

$$V(k) = \begin{cases} \frac{1}{2} W_{4N}^{-k} [C(k) - j\, C(N-k)], & k=0,\dots,N/2 \\ V^{*}(N-k), & k=\frac{N}{2}+1,\dots,N-1 \end{cases} \qquad (6)$$

where * denotes complex conjugation;

b) compute v(n) as the IDFT of V(k)

$$v(n) = \frac{1}{N} \sum_{k=0}^{N-1} V(k)\, W_N^{-nk} \qquad (7)$$

through an IFFT algorithm;

c) obtain the sequence x(n) according to the relation (3).

The procedures outlined above for the implementation of the DCT and IDCT can be further improved by noting that the sequences to be transformed, x(n) and C(k), are real sequences. Therefore, the standard techniques [6] for implementing the transform of two real sequences by means of one complex transform (FFT or IFFT) can be applied in this case, obtaining a further computational gain of a factor of 2.

SELECTION CRITERIA

As is known, the coefficients of the transformed image can be selected according to two different criteria:

a) zonal sampling, where the retained coefficients are selected according to their belonging to a (generally prefixed) zone;

b) threshold sampling, where the retained coefficients are selected according to their amplitude (i.e. the coefficients below a prefixed threshold are discarded).

The two selection criteria differ in two main respects:

i) threshold sampling necessitates the transformed coefficients to be identified by their associated 'addresses', which have to be stored or transmitted, while zonal sampling does not require the transformed coefficient identification;

ii) by defining the selection zone, zonal sampling determines a priori the number of coefficients to be retained in the transformed image, while threshold sampling cannot determine a priori the number of transformed coefficients above the prefixed value.

PERFORMANCE OF THE DCT APPLIED TO IMAGE DATA COMPRESSION

Different types of images have been processed and compressed by the DCT and both zonal sampling and threshold sampling have been applied to the transformed images.

The performance of each method has been evaluated according to the following main parameters:

1) The compression ratio CR, defined as the percentage of the retained transformed coefficients with respect to the original image samples. It should be noted that this parameter does not consider either any efficient coding or any identification of the transformed coefficients.

2) The mean square error e^2_{rms} defined as

$$e^2_{rms} = \frac{1}{M^2} \sum_{i=1}^{M} \sum_{j=1}^{M} [x(i,j) - \hat{x}(i,j)]^2 \tag{8}$$

where $x(i,j)$ and $\hat{x}(i,j)$ are the M x M sequences of the original image and of the reconstructed image after the compression operation, respectively.

3) The peak error e_p defined as

$$e_p = \max_{i,j} |x(i,j) - \hat{x}(i,j)| \tag{9}$$

Original images of size M = 256 have been chosen. They have been processed by applying the DCT to blocks of size N x N. The values chosen for the block size have been N = 4, 8, 16, 32, 64, 128, 256. For each choice of the block size the previous performance parameters (together with other important quantities) have been obtained for both zonal and threshold sampling. Six different extensions of the selection zone for the zonal sampling and six different thresholds for the threshold sampling have been considered.

Some important conclusions can be drawn from the extensive results obtained from this study [7]. The comparison between the performance achieved by the zonal and threshold samplings has shown that threshold sampling gives better results than zonal sampling. For example, for a fixed compression ratio, threshold sampling gives much smaller errors, e_{rms} and e_p, than zonal sampling. Alternatively, prefixed errors, e_{rms} and e_p, can be obtained by the threshold sampling with a compression ratio about half that of zonal sampling. This is true quite independently of the transform block size N. Indeed the dependence of the zonal sampling on the block size N is quite negligible. On the other hand, the block size N has some influence on the performance of the threshold sampling, indicating a preference for large transform block sizes (N = 64, 128, 256). However, taking into account both the performance achieved and the implementation computational complexity (which increases with the block size), a suitable compromise choice for N ranges from 8 to 32.

Another common and widely used transform suitable for image data compression is the FFT. A comparison between the performance obtained by the FFT and the DCT has been also carried out. The results have shown a far better performance of the DCT (threshold sampling) with respect to the FFT (threshold sampling). The performance degradation (in terms of the errors, e_{rms} and e_p) of the FFT with respect to the DCT is of the same order as the degradation of the DCT zonal sampling with respect to the DCT threshold sampling.

EXAMPLES OF IMAGE DATA COMPRESSION BY THE DCT

Some examples of the results obtained from the application of the DCT to the image data compression problem are shown in the following. The DCT has been implemented by the described fast DCT algorithm (FCT). Figs. 1 and 2 show the application of the FCT to a Landsat image (band 4) of a region in the South of Italy (Sele River in Campania). Fig. 1 refers to the threshold sampling, whereas Fig. 2 refers to the zonal sampling. In both cases the original image is the upper-left image. Fig. 1 shows the reconstructed images with a different selection of the threshold T (in percent of the maximum transformed coefficient) and the obtained compression ratio CR. Fig. 2 reports the reconstructed images by the zonal sampling obtained at approximately the same CR. The block size of the transformation is N = 16. For the same block size Fig. 3 shows the transformed image, i.e. the coefficients obtained after the application of the DCT to the original image in blocks 16 x 16. Energy concentration can be observed near the origin of each block, as expected by the DCT property of closely approximating the optimal Karhunen-Loeve transform.