Image Block Edge Classification with Block/Subblock Conversion in the Discrete Cosine Transform Domain

Javier Vega-Pineda, José A. Salinas-Terrazas and Rafael Sandoval-Rodríguez

Instituto Tecnológico de Chihuahua, DEPI
Av. Tecnológico 2909, Chihuahua, Chih., C.P. 31310, México
{jvega, rsandova}@itchihuahua.edu.mx

Abstract. We present a method to classify the edge orientation of the blocks in images under the Discrete Cosine Transform (DCT) domain. The method includes a previous stage to join up or split down the original 8x8 blocks of a traditional JPEG compressed image forming DCT blocks with different dimensions. The change of dimensions is totally realized in the DCT domain and includes two approaches: a general spatial scheme for 2^n and non 2^n related DCT block sizes and a fast scheme just for 2^n related DCT blocks. Both approaches are capable of work dividing DCT block into their subblocks and backwards from the subblocks to the block. The edge classifier takes advantage of the block conversion process to compute the belonging class performing as the spatial domain algorithm version. The method is oriented to applications where the feature extraction from compressed domain images is important.

Keywords. Block edge classification, Discrete cosine transform (DCT), Image feature extraction.

1 Introduction

The use of compressed still digital images is increasing day by day to reduce transmission time and storage space. There are several well known formats in which the images may be coded and one of the most used is JPEG [1]. The JPEG Baseline standard is based on the Discrete Cosine Transform (DCT) [2], and numerous image databases are structured with images under this format. Therefore, new direct image processing in compress domain areas are emerging. Among others, watermarking [3], feature extraction [4-5] and error concealment [6]. Those applications may need to work with image blocks of different size to the 8x8 defined by DCT-JPEG, and therefore the conversion of their size is needed.

After the review of several methods to do the DCT block size conversion, we selected and adapted two of them as the first stage of the Block Edge (BE) classification method [7-10]. The block conversion method of Feng and Jiang [7-8] is the most general in terms of block dimensions, bidirectionality, and the number of arithmetic operations is lower than using the traditional IDCT method. The core factor in the method is to solve a set of linear equations leading to a conversion matrix with properties related to the desired block sizes conversion. The alternative method proposed by He et al [9], is based on a closely replica of the fast Discrete Fourier Transform

© S. Torres, I. López, H. Calvo. (Eds.)
Advances in Computer Science and Engineering
Research in Computing Science 27, 2007, pp. 101-111

Received 20/02/07
Accepted 08/04/07
Final version 18/04/07

(DFT) algorithms applied to the DCT to modify the blocks size. This method is bidirectional and reduces the number of arithmetic operations compared to Feng and Jiang approach, but it is applicable only to $2^n \times 2^n$ blocks. The possibility to modify the DCT image block size will permit to do processing on a global or local region.

The BE classification stage defines the edge orientation (0, 45, 90, 135 degrees) or the edge absence according to the maximum value of previously calculated measures. The edge classification is based on the first probabilistic moment (mean) computed for each of the four subblocks contained into the block under analysis. The edge classification method was proposed by Sung and Kang [10] and is defined in both domains spatial and DCT. However, taking advantage of previous results from the block conversion stage we modified the DCT approach implementing a method similar to the spatial approach but maintaining the image in the DCT domain. Li et al [11] analyze a set of DCT coefficients to determine the edge orientation which is less efficient than use the mean of subblocks proposed by [10], and more recently they [12] proposed a new scheme based on the Haar transform with a drastic reduction in the number of arithmetic operations but it is a method to be applied in the pixel domain. The modified Sung and Kang approach [10] prove to work very efficiently, and the metrics used to classify the edges based on the mean are very intuitive.

The method proposed in this paper turns to be an integration of several modified algorithms adapted and optimized to provide a two fold image processing application which performs totally in the DCT domain: the classification of the image blocks, and the modification of the relationship between the DCT coefficients of its blocks and their subblocks. Therefore, image global and local feature extraction (edges orientation) in the compress – DCT domain is possible. Experimental results are presented and the general structure of the method explained.

The remainder of this article is organized as follows. In section 2, the basic aspects of the selected block conversion algorithms are summarized. Section 3 explains how we have simplified the selected BE algorithm and the orientation measures explained. Section 4 describes the integrated method as a tool to classify BE using the algorithms explained in sections 2 and 3. Finally, section 5 reports results in terms of image processing and draws some conclusions.

2 Algorithms for Conversion Between DCT Blocks and Subblocks

The Feng and Jiang [7] and also the He at al [9] methods are summarized, both relating the 2-D DCT blocks and their subblocks. Before presenting the two conversion algorithms, we define notation and formulate the specific problem. Given a 2-D block of pixels B, its DCT will be C_B. B can be divided into B_{ij} subblocks where the corresponding DCT for each subblock is C_{ij}. This is illustrated in Fig. 1. In the first approach [7], we will solve a set of linear equations relating the DCT basis used by the block and its sub-blocks. In the second approach [9], a fast algorithm based on DCT-II and DCT-IV which resembles the Fast Fourier Transform structure is used to relate C_B and C_{ij}.

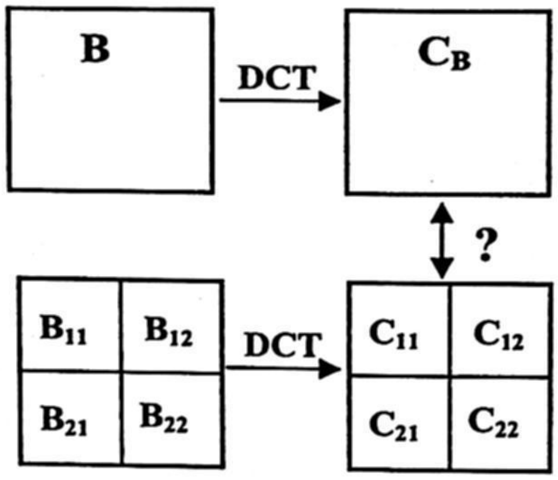

Fig. 1. The relationship between DCT a block and its subblocks

The definition for the 2-D DCT (or DCT-II) used for the algorithms development is given by

$$C_B(u,v)=\sqrt{\frac{4}{R\times C}}\alpha(u)\alpha(v) \cdot\sum_{i=0}^{R-1}\sum_{j=0}^{C-1}x(i,j)\cos\left(\frac{(2i+1)u\pi}{2R}\right)\cos\left(\frac{(2j+1)v\pi}{2C}\right) \tag{1}$$

where $\{x(i,j)\}$ are the elements of B, and $\alpha(u)=\begin{cases}\sqrt{\frac{1}{2}}, & \text{for } u=0\\ 1, & \text{otherwise.}\end{cases}$

2.1 Method Based on the Spatial Relationship Between DCT Blocks/Subblocks

The forward conversion will be defined as joining subblocks C_{ij} to reach a block C_B. The problem is reduced to solve the linear relationship between two families of DCT basis functions with equal dimensions. Assuming that C_B has dimensions of $R \times C = LN \times MN$, the subblock structure has $L \times M$ subblocks, with size each of $N \times N$. The first basis corresponds to the block and the second basis corresponds to one of the subblocks forming the block, i.e., $b_2(k,t)_{MN\times MN}=\cos(k\pi t/MN)$, $k = 0, 1,\ldots, MN-1$ and $b_1(k,t)_{N\times N}=\cos(k\pi t/N)$, respectively. $b_2(k,t)$ must have the same size than $b_1(k,t)$, then this last is expanded by zero padding forming $b(k,t)_{MN\times MN}$ a matrix with b_1 submatrices in its main diagonal. The linear relationship is established as

$$b_2(k,t)=\sum_{j=0}^{MN-1} a(k,j)b(j,t)\,. \tag{2}$$

where $t = (2i+1)/2$, ($i = 0,1,\ldots,MN$-1). (3) can be written in compact matrix form as b_2=Ab.

Solving for a unique $a(k,j)_{MN \times MN}$ (A) and after applying the normalization constants $\alpha_n(.)$ [7], will give us the forward conversion matrix A^* between subblocks to one block. A^{*-1} is the backward conversion matrix. A numerical example follows: $N = 3$, $L = 2$ and $M = 2$, then four 3 x 3 subblocks (2 by row and 2 by column) will be converted to one 9 x 9 block. First, compute $A^*=\alpha_n b_2 b^{-1}$, and second, apply A^* to the rows and columns of C_{ij}. The transformation can be expressed as,

$$C_B=\sqrt{\frac{1}{LM}}A^*C_{ij}A^{*T}\,. \tag{3}$$

The forward conversion matrix A* is computed as

$$A^*=\begin{bmatrix} 1 & 0 & 0 & 1 & 0 & 0 \\ 0.9107 & 0.4082 & -0.0632 & -0.9107 & 0.4082 & 0.0632 \\ 0 & 1 & 0 & 0 & -1 & 0 \\ -0.3333 & 0.8165 & 0.4717 & 0.3333 & 0.8165 & -0.4717 \\ 0 & 0 & 1 & 0 & 0 & 1 \\ 0.244 & -0.4082 & 0.8797 & -0.244 & -0.4082 & -0.8797 \end{bmatrix},$$

with an input set of subblocks C_{ij}, results in a block C_B:

$$C_{ij}=\begin{bmatrix} \begin{bmatrix} 167 & 2 & 0 \\ -4 & -1 & 0 \\ 2 & -1 & 0 \end{bmatrix} & \begin{bmatrix} 176 & -12 & 5 \\ -18 & 8 & 1 \\ 5 & -1 & -1 \end{bmatrix} \\ \begin{bmatrix} 192 & 3 & -1 \\ -10 & -3 & 0 \\ 0 & 1 & 1 \end{bmatrix} & \begin{bmatrix} 204 & -20 & 3 \\ 18 & -5 & -2 \\ -3 & 1 & -3 \end{bmatrix} \end{bmatrix} \quad C_B=\begin{bmatrix} 370 & -15 & 18 & -10 & 3 & -1 \\ -27 & 0 & -6 & 3 & 1 & -2 \\ -15 & 22 & -5 & -2 & 2 & 1 \\ 5 & -8 & -1 & 0 & -1 & -1 \\ 2 & 0 & 0 & 1 & -2 & 2 \\ 1 & 0 & 0 & -2 & 1 & -1 \end{bmatrix}.$$

Solving (3) for C_{ij} will define the block to subblocks transformation. The algorithm is valid for blocks with dimensions equal and different to $2^n \times 2^n$, and also there is a definition for 1-D vectors.

2.2 Method Based on the Fast Computation Between DCT Blocks/Subblocks

This method is mainly based on the structure of decimation-in-frequency FFT algorithm [9]. Mapping the structure to the DCT, a transformation matrix T_N can be decomposed into a straight forward combination of the submatrix $T_{(N/2)}$, where N is the number of values to transform. The conversion is applied only for $2^n \times 2^n$ blocks. For 2-D DCT forward subblocks to block conversion, the related equations are given as

$$C_B = \frac{1}{2} T_N \begin{bmatrix} B_{11} & B_{12} \\ B_{21} & B_{22} \end{bmatrix} T_N^T = \frac{1}{2} P_N \begin{bmatrix} Y_1 & Y_2 \\ Y_3 & Y_4 \end{bmatrix} P_N^T, \tag{4}$$

$$\begin{aligned}
Y_1 &= C_{11} + S_{\frac{N}{2}} C_{21} + C_{12} S_{\frac{N}{2}} + S_{\frac{N}{2}} C_{22} S_{\frac{N}{2}} \\
Y_2 &= \left(C_{11} + S_{\frac{N}{2}} C_{21} - C_{12} S_{\frac{N}{2}} - S_{\frac{N}{2}} C_{22} S_{\frac{N}{2}} \right) B_4 \\
Y_3 &= B_4^T \left(C_{11} + C_{12} S_{\frac{N}{2}} - S_{\frac{N}{2}} C_{21} - S_{\frac{N}{2}} C_{22} S_{\frac{N}{2}} \right) \\
Y_4 &= B_4^T \left(C_{11} - C_{12} S_{\frac{N}{2}} + S_{\frac{N}{2}} C_{22} S_{\frac{N}{2}} - S_{\frac{N}{2}} C_{21} \right) B_4
\end{aligned} \tag{5}$$

where C_{ii} = 2-D DCT(B_{ii}), T_N is the N-values DCT transformation matrix, P_N is the $N \times N$ permutation matrix (rows in order 1, 3,..., N-1, 2, 4,..., N), $S_{N/2} = T_{N/2} \overline{I_{N/2}} T_{N/2}^T$, $\overline{I_N}$ is the reverse identity matrix, $B_4 = T_{N/2} \left(T_{N/2}^{IV} \right)^T \overline{I_{N/2}}$, $T_{N/2}^{IV}$ is the N/2-values DCT-IV (see Appendix A) and Y_i are linear combinations of the four transform matrices (DCT sections).

The linear combinations for 2-D DCT backward block to subblocks conversion are given as

$$\begin{aligned}
C_{11} &= \frac{1}{4} \left(K_1 + K_2 + K_3 + K_4 \right) \\
C_{12} &= \frac{1}{4} \left(K_1 - K_2 + K_3 - K_4 \right) S_{\frac{N}{2}} \\
C_{21} &= \frac{1}{4} S_{\frac{N}{2}} \left(K_1 + K_2 - K_3 - K_4 \right) \\
C_{22} &= \frac{1}{4} S_{\frac{N}{2}} \left(K_1 - K_2 - K_3 + K_4 \right) S_{\frac{N}{2}},
\end{aligned} \tag{6}$$

where the $(N/2)\times(N/2)$ matrices K_i are given as

$$K_1 = Y_1, \quad K_2 = Y_2 B_4^T, \quad K_3 = B_4 Y_3, \quad K_4 = B_4 Y_4 B_4^T. \tag{7}$$

A numerical example follows: converting a 4 x 4 block into its four 2 x 2 subblocks $N = 4$ (backward conversion). Solving (4) for Y_i's and then substituting in (6) through (7) for K_i's:

$$C_B=\begin{bmatrix}218 & 45.63 & 22 & -4.8\\ 74.7 & -66.56 & -24.14 & -26.8\\ -58.5 & 30.76 & 0.5 & -13.29\\ -3.89 & -33.3 & -7.32 & 14.06\end{bmatrix}\quad P_4=\begin{bmatrix}1&0&0&0\\0&0&0&1\\0&1&0&0\\0&0&1&0\end{bmatrix}\quad \begin{bmatrix}Y_1 & Y_2\\ Y_3 & Y_4\end{bmatrix}=\begin{bmatrix}436 & 44 & -9.66 & 91.25\\ -117 & 1 & -26.56 & 61.52\\ -7.76 & -14.64 & 28.11 & -66.6\\ 149.40 & -48.28 & -53.6 & -133.11\end{bmatrix}$$

$$T_2=\begin{bmatrix}0.7071 & 0.7071\\ 0.7071 & -0.7071\end{bmatrix}\quad S_2=\begin{bmatrix}1 & 0\\ 0 & -1\end{bmatrix}\quad T_2^{IV}=\begin{bmatrix}0.9239 & 0.3827\\ 0.3827 & -9239\end{bmatrix}\quad B_4=\begin{bmatrix}-0.383 & 0.924\\ 0.924 & 0.383\end{bmatrix}$$

$$\begin{bmatrix}K_1 & K_2\\ K_3 & K_4\end{bmatrix}=\begin{bmatrix}436 & 44 & 88 & 26\\ -117 & 1 & 67 & -1\\ 141 & -39 & -67 & -93\\ 50 & -32 & -106 & -38\end{bmatrix}\quad \begin{bmatrix}C_{11} & C_{12}\\ C_{21} & C_{22}\end{bmatrix}=\begin{bmatrix}149.5 & -15.5 & 139 & -18\\ -26.5 & -17.5 & -7 & -2\\ 112.5 & 50.5 & 35 & 9\\ -1.5 & -17.5 & 85 & -1\end{bmatrix}$$

In this method most of the matrices can be computed before the block conversion. Even when the reordering operations have not computational cost [9], they are required to arrive to the correct values position. There is also a 1-D version of the algorithm.

3 Compress Domain Block Edge (BE) Classification Method

The BE classification method in the pixel domain working on an image can be easily described as follows:

i. The image $N \times N$ blocks are divided in their symmetrical four subblocks.
ii. The mean (first probabilistic moment) is computed for each subblock: S_{11}, S_{12}, S_{21}, S_{22}.
iii. Define the threshold value for edge or not edge block (heuristically), d_{NE}.
iv. The measures for directional edge pattern are computed using equations in Table 1.
v. The measure with the maximum value including the edge threshold value, defines the edge class.

Table 1. Measures for block directional patterns

Edge Direction (*radians*)	*Measure*
No Edge	d_{NE} (set by user)
0	$d_0=\frac{1}{2}\lvert S_{11}+S_{12}-(S_{21}+S_{22})\rvert$
$\pi/4$	$d_{45}=\max\left\{\frac{1}{3}\lvert 3S_{11}-(S_{12}+S_{21}+S_{22})\rvert,\frac{1}{3}\lvert 3S_{22}-(S_{11}+S_{12}+S_{21})\rvert\right\}$
$\pi/2$	$d_{90}=\frac{1}{2}\lvert S_{11}+S_{21}-(S_{12}+S_{22})\rvert$
$3\pi/4$	$d_{135}=\max\left\{\frac{1}{3}\lvert 3S_{12}-(S_{11}+S_{21}+S_{22})\rvert,\frac{1}{3}\lvert 3S_{21}-(S_{11}+S_{12}+S_{22})\rvert\right\}$

The computation of the measures in the DCT domain also is defined in [10]. However, because the directionality measures are based only on the average subblock values, we proposed a modification to the method. Is well known that the DC coefficient in a DCT block is N times the whole block average value [13], this can be written as

$$\text{(In pixel domain): } m = \frac{1}{N^2}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(i,j)\text{ ; (In DCT domain): } m = \frac{1}{N} C_B(0,0). \quad (8)$$

Therefore, using the DCT-DC coefficient from each subblock inside the block the five measures at Table 1 can be computed and a BE class defined. The DCT subblocks generation may then be performed by any of the two block/subblock conversion methods described in section 2.

4 Image Feature Extraction Tool: Integrating Block/Subblock Conversion and BE Classification

In Fig. 2, the methods explained in sections 2 and 3 are integrated as an image global or local region feature extraction tool. The DCT blocks or subblocks are converted to the desired size according to the final application using the tool. The converted blocks are 2-D subsampled (ND-ND) leaving the DC coefficients to compute the BE directional measures. The measure with the maximum value defines a pointer to the edge class and the corresponding tag is assigned to the block.

An important difference between the methods for block conversion presented at section 2 is the scope obtained in the size conversion. The fast computation method (section 2b), works in a recursive structure, the previous $2^n \times 2^n$ subblocks are required to the next conversion, i.e., the conversion from 8^2 to 2^2 blocks needs to compute previously the 8^2 to 4^2 conversion.

With the spatial relationship method (section 2a), an A* matrix can be previously calculated and the conversion 8^2 to 2^2 be performed in one step. Be A* the N^2 to $(N/2)^2$ conversion matrix and be D^* the $(N/2)^2$ to $(N/4)^2$ conversion matrix. Notice that A* is an $N \times N$ matrix and D^* is $(N/2) \times (N/2)$ matrix. Therefore D^* is resized to $N \times N$ by zero padding out of the main block diagonal. Solving (3) for C_{ij}, the N^2 to $(N/2)^2$ conversion is defined as

$$C_{ij(N/2)^2} = \sqrt{LM}\, A^{*-1} C_{B(N^2)} \left(A^{*-1}\right)^T, \quad (9)$$

zero padded D^* is applied to (9),

$$C_{ij(N/4)^2} = LM \cdot D_Z^{*-1} A^{*-1} C_{B(N^2)} \left(A^{*-1}\right)^T \left(D_Z^{*-1}\right)^T, \text{where } D_Z^* = \begin{bmatrix} D^* & 0 \\ 0 & D^* \end{bmatrix}_{N^2}. \quad (10)$$

The 8^2 to 2^2 conversion matrix is then computed as $A_{8-2}^* = D_Z^{*-1} A^{*-1} = \left(A^* D_Z^*\right)^{-1}$.

Using the fast DCT approach there is an important reduction on the computational cost [9]. Furthermore, the position of the subblocks DC coefficients in (6) remains the same before and after the multiplications of the linear combinations by $S_{N/2}$, therefore, in our application the complexity is reduced avoiding such multiplications. Results applying the image feature extraction tool are presented in the next section.

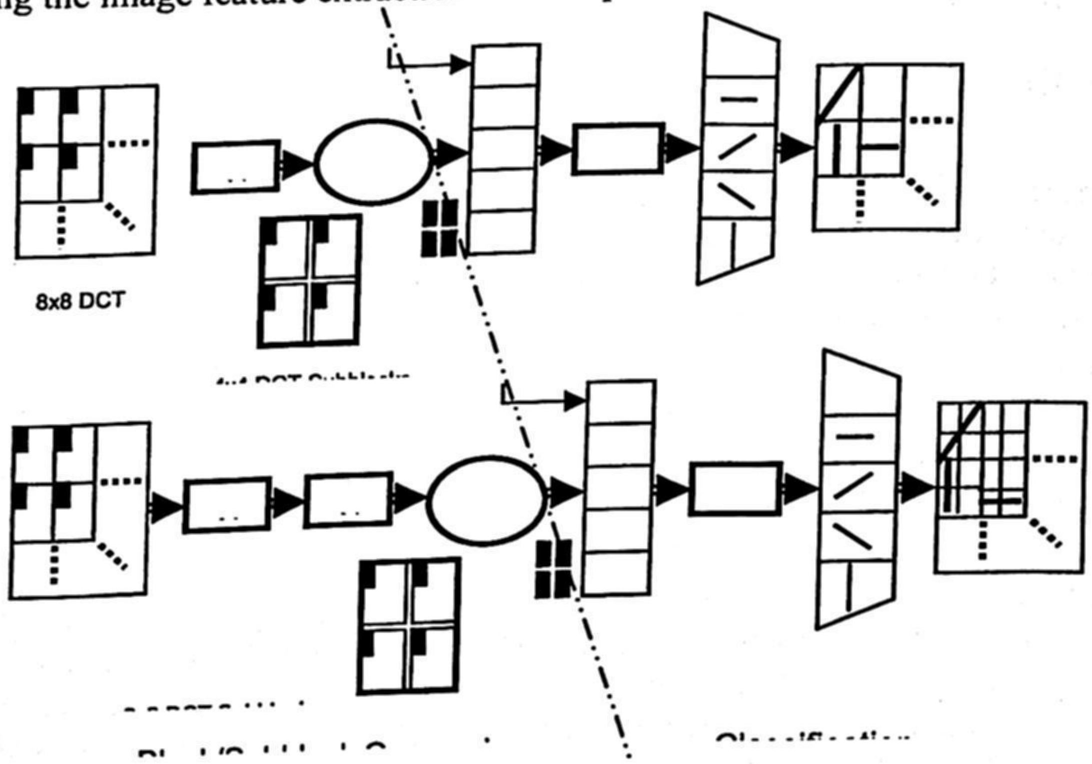

Fig. 2. Integrating Block/Subblock Conversion and BE Classification

5 Experimental Results and Conclusions

The results of analyzing Barbara image is presented at Fig. 3, for threshold values d_{NE} of 10 and 20, and for 8^2 and 4^2 blocks with 4^2 and 2^2 subblocks respectively [14]. Those blocks tagged as an oriented edge where substituted by its corresponding edge image, and those blocks below the d_{NE} value where left blanked. The classified edges where assembled as an image to compare the results with the original Barbara. For small threshold value d_{NE} much more blocks are classified as BE, this can be easily detected at the head and the table cloth of Barbara. The change of block size from 8^2 to 4^2 as was expected, increases the image details and therefore the number of blocks classified as BE. A set of images with the detail of Barbara face are show in Fig. 4, supporting the previous comment.

The BE Classification Method proposed in [10] is defined to work in the DCT domain with blocks of size 8^2 or smaller. With the spatial-like approach proposed in this article it is possible to apply the method to blocks of dimensions above of 8^2, because just the four subblocks DC coefficients are needed to compute the measures for direc tional patterns. In order to compute the BE pattern in a 16^2 block, the DC coefficients of its four corresponding 8^2 subblocks are needed.

The integration of both stages, the block/subblocks conversion and the BE classification all in the DCT domain performed successfully. The modifications to the block/subblocks methods simplified the whole tool complexity keeping the flexibility of choice between the general [7] and the fast DCT computation schemes [9].

Fig. 3. Barbara images BE classified. Top left, Original. Top center and right, 8^2 blocks, d_{NE}= 20 and d_{NE}= 10, respectively. Bottom left and right, 4^2 blocks, d_{NE}= 20 and d_{NE}= 10

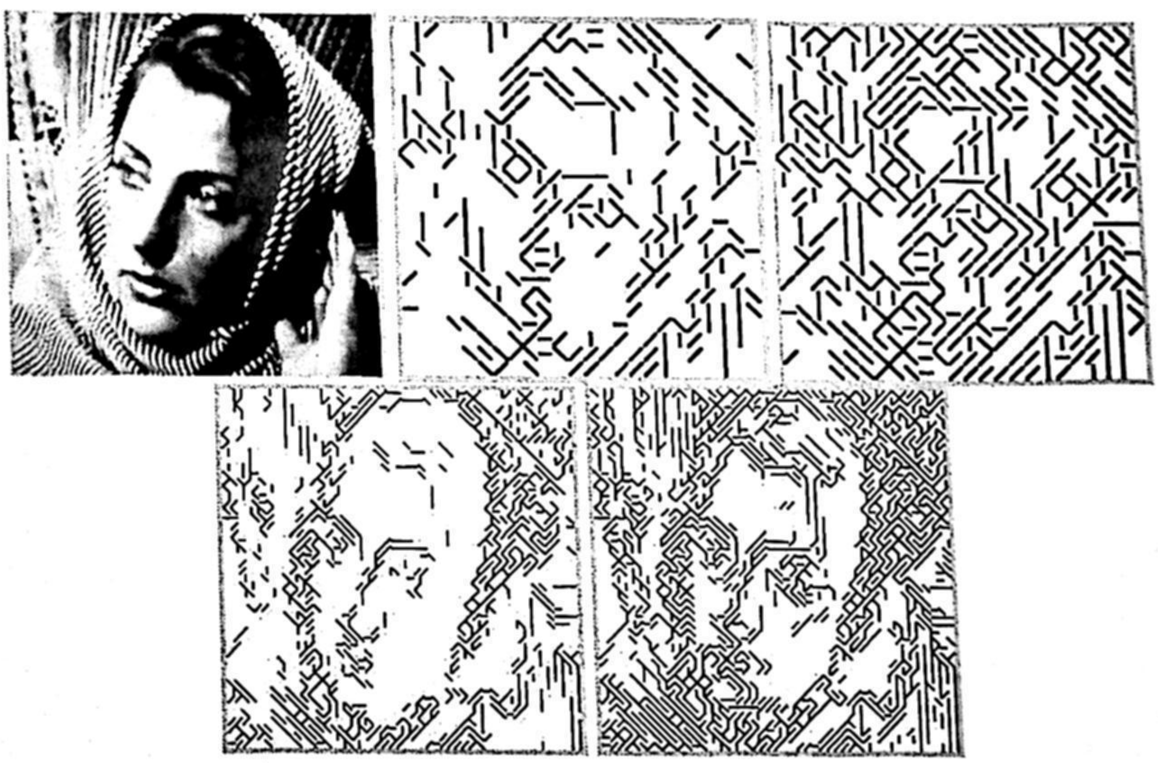

Fig. 4. Barbara face images BE classified. Top left, Original. Top center and right, 8^2 blocks, d_{NE}= 20 and d_{NE}= 10, respectively. Bottom left and right, 4^2 blocks, d_{NE}= 20 and d_{NE}= 10

The BE classifier resembling closely the pixel scheme [10] in place of the DCT version, takes advantage of the preprocessing DCT blocks stages simplifying even further the edge classification. Pattern recognition [14] and video analysis [10] in the DCT domain are other research directions for the work here presented. The digital image processing area in the compress domain is an emerging and interesting area, this article is a small example.

Acknowledgments. The authors are grateful to the support given by the Dirección General de Institutos Tecnológicos (DGEST), the CONACYT, and the Instituto Tecnológico de Chihuahua.

References

1. Pennebaker W.E., Mitchell J.L.: JPEG: Still Image Data Compression Standard. Kluwer Academic Publishers, Norwell, MA, (1992)
2. Rao K.R., Yip P.: Discrete Cosine Transform: Algorithms, Advantages, Applications. Academic Press, San Diego, CA (1990)
3. Wen-Nung L., Guo-Chiang L., Chih-Liang W., Ta-Chun W.: Robust Image Watermarking On the DCT Domain, Proc. IEEE International Symposium On Circuits and Systems, Vol. I. Geneva, Switzerland, May 28-31 (2000) 228-231
4. Jiang J., Armstrong A., Feng G.C.: Direct Content Access and Extraction form JPEG Compressed Images, Pattern Recognition, Vol. 35. Elsevier Science Ltd (2002) 2511-2519
5. Beetnara B., Sun W.Y., Yong M.R.: Fast MPEG-7 Visual Description Extraction Using DCT Coefficient, Proc. IEEE Conference of Convergence Technologies for Asia-Pacific Region, Vol. 3. Bangalore, India, Oct. 15-17, (2003) 1136-1139
6. Hemami S.S., Gray R.M.: Subband-Coded Image Reconstruction for Lossy Packet Network, IEEE Trans. on Signal Processing, Vol. 6. No. 8. (1999) 202-204
7. Jiang J., Feng G.: The Spatial Relationship of DCT Coefficients Between a Block and Its Sub-blocks, IEEE Trans. on Signal Processing, Vol. 50. No. 5. (2002) 1160-1169
8. Jiang J., Feng G.: Image Spatial Transformation in DCT Domain, Proc. IEEE International Conference on Image Processing, Vol. 3. Thessaloniki, Greece, Oct 7-10 (2001) 836-839
9. He Z., Bystrom M., Nawab H.: Bidirectional Conversion Between DCT Coefficients of Blocks and Their Subblocks, IEEE Trans. on Signal Processing, Vol. 53. No. 8. (2005) 2835-2841
10. Sung C.H., Kang K.: A Compressed Domain Scheme for Classifying Block Edge Patterns, IEEE Trans. on Signal Processing, Vol. 14. No. 2. (2005) 145-151
11. Li H., Liu G., Li Y.: An Effective Approach to Edge Classification from DCT Domain, Proc. IEEE International Conference on Image Processing, Vol. 1. Rochester, NY, Sep 22-25 (2002) 940-943
12. Li H., Ngan K.N.: Fast and Efficient Method for Block Edge Classification, Proc. International Conference on Communications and Mobile Computing, ACM Press, New York, NY, July 03-06, (2006) 67-72
13. Liu J.G., Liu Y.Z., Wang G.Y.: Fast DCT-I, DCT-III, and DCT-IV via Moments, Eurasip Journal on Applied Signal Processing, Vol. 12. Hindawi Pub. Co., (2005) 1902-1909
14. Salinas-Terrazas J. A.: Búsqueda de Patrones Geométricos en Imágenes en el Dominio de la Transformada Discreta del Coseno, M. Sc. Thesis, Instituto Tecnológico de Chihuahua, Chihuahua, Chih., México, (2007)

Appendix: A

The Discrete Cosine Transform II and IV are defined by different equations [2]. The 2-D DCT-II is the one in equation (1). The 1-D DCT-IV is defined as

$$C_B^{IV}(k) = \sqrt{\frac{2}{N}} \sum_{n=0}^{N-1} x(n) \cos \frac{\pi(2n+1)(2k+1)}{4N} \tag{11}$$

where $k = 0,\ldots,N\text{-}1$. The $N \times N$ transformation matrix for the DCT-IV is defined as

$$T_N^{IV}(i,j) = \sqrt{\frac{2}{N}} \cos \frac{\pi(2i+1)(2j+1)}{4N}, \quad i,j \in [0,\ldots,N-1]. \tag{12}$$