2

Efficient 2-D DCT Computation from an Image Representation Point of View

G.A. Papakostas[1], D.E. Koulouriotis[2] and E.G. Karakasis[2]
[1]*Democritus University of Thrace, Department of Electrical and Computer Engineering*
[2]*Democritus University of Thrace, Department of Production and Management Engineering*
Greece

1. Introduction

Discrete Cosine Transform (DCT) constitutes a powerful tool in signal processing, since its first introduction (Ahmed et al., 1974). It belongs to a class of mathematical operations that includes the well-known Fast Fourier Transform (FFT), having as basic operation taking a signal and transforming it from one type of representation to another. More precisely, DCT transforms a signal from the spatial domain to the frequency space, with minimum information redundancy, since its kernel functions (cosines) comprise an orthogonal basis.

The main advantage of the DCT transform is its high energy compactness and thus the resulted DCT coefficients fully describe the signal in process. This benefit in conjunction with its implementation simplicity has inspired the scientists to use DCT as the basic transform in the well known image compression standard calling JPEG (ISO/IEC, 1994). Particularly, a 2-D version of the DCT is used to compute the projection of an image in the orthogonal basis of cosines functions, by resulting to a set of coefficients that constitutes the DCT image domain. According to the JPEG standard these coefficients are being compressed in a next step by applying a specific quantization table and an entropy coding procedure.

Besides the usage of the 2-D DCT as part of image compression algorithms, it is widely used as feature extraction or dimensionality reduction method in pattern recognition applications (Sanderson & Paliwal, 2003; Er et al., 2005; Jadhav & Holambe, 2008; Liu & Liu, 2008), in image watermarking and data hiding (Qi et al., 2008; Choi et al., 2008; Alturki et al., 2007; Wong et al., 2007) and in various image processing applications (See et al., 2007; Krupinski & Purczynski, 2007; Abe & Iiguni, 2007).

From the above it is obvious that the applicability range of the 2-D DCT is wide and increases continuously. This fact has motivated many researchers to investigate and develop algorithms that accelerate the computation time needed to calculate the DCT coefficients of an image. As a result of this massive research, many algorithms that present high computation rates were proposed (Zeng et al., 2001; Diab et al., 2002; Wu et al., 2008; Shao et al., 2008; Plonka & Tasche, 2005) and many hardware implementations were presented (Nikara et al., 2006; Tan et al., 2001) through the years.

Source: Image Processing, Book edited by: Yung-Sheng Chen,
ISBN 978-953-307-026-1, pp. 572, December 2009, INTECH, Croatia, downloaded from SCIYO.COM

These algorithms manage to reduce the number of operations in terms of multiplications and additions, by applying common techniques with the DFT algorithms, where optimized arithmetic methodologies and matrix operations are applied.

A completely different methodology that is making use of the image's morphology and intensities distribution, which highly improves the computational rate of the 2-D DCT, is presented in this chapter, as alternative to the conventional one. Moreover, the algorithms' nature allows the collaboration with the already existing fast methods, in order to provide high computation rates.

2. 2-D Discrete Cosine Transform

The Discrete Cosine Transform (DCT) is a popular signal transformation method, which is making use of cosine functions of different frequencies, as kernels. There are several variants of DCT with slightly modified definitions and properties, like DCT I, II, III, IV, V-VIII (Wikipedia) with the corresponding inverse formulas.

Among these types the DCT II, is usually used in image processing and compression (JPEG, MPEG), because it has strong energy compaction, meaning that a few coefficients enclose the most of the signal in process.

The *(p+q)th* order DCT-II coefficient for an *NxN* image having intensity function *f(x,y)*, is described by the following formulas

$$C_{pq} = \frac{1}{\rho(p)\rho(q)} \sum_{x=0}^{N-1} \sum_{y=0}^{N-1} D_p(x) D_q(y) f(x,y) \tag{1}$$

with 0 ≤ p,q,x,y ≤ N-1, and

$$D_n(t) = \cos\left(\frac{(2t+1)n\pi}{2N}\right) \tag{2}$$

is the cosine kernel function of the orthogonal basis, while

$$\rho(n) = \begin{cases} N & if\ n = 0 \\ N/2 & otherwise \end{cases} \tag{3}$$

is a normalization factor.

According to (1) there are a lot of computations that have to be performed in calculating a set of DCT coefficients, due to the cosine functions evaluations, but most of all due to the number of image's pixels. While in the case of blocked-DCT algorithm (1) is applied to an 8x8 image block and thus the number of pixels is quite small, in the case of the global-DCT the number of pixels is increasing and is equal to the total image pixels.

A novel methodology that attempts to reduce the complexity of the above formulas by modifying these definitions in a more simple form is presented, in the next section.

3. Improved 2-D DCT computation

As it can be seen from (1), the high computational complexity of the 2-D DCT computation process, lies on the calculation of the same quantities for the entire image and thus the computation time is highly dependent on the image size by resulting to a *O(N^2)* complexity. This fact was the main point of inspiration of investigating alternative computation methodologies to decrease the complexity by simplifying the way the operations are performed.

Recently, the authors proposed a new algorithm to accelerate the computation of Legendre moments on binary images (Papakostas et al., 2007) and generalized it in the case of gray-scale images with application on the Geometric moments (Papakostas et al., 2008), based on a new image representation scheme called Image Slice Representation (ISR).

This new image representation method is adopted and applied on the computation of 2-D DCT coefficients, in order to improve the overall computation rate, as it will be verified by appropriate experiments.

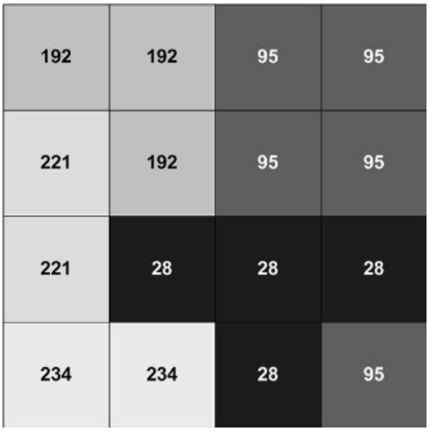
3.1 Image Slice Representation (ISR)

The main idea behind the ISR is based on a novel image perspective according to which, each gray-scale image consisting of pixels having several intensities in the range [0,255], can be completely decomposed by a finite number of image slices of different intensity.

Each slice can be defined as the following definition describes,

Definition 1: Slice of a gray-scale image, of a certain intensity f_i, is the image with the same size and the same pixels of intensity f_i as the original one, while the rest of the pixels are considered to be black.

For example let us consider the following 4x4 gray-scale image, where the pixels' intensities have been highlighted, so that the decomposition result is clear.



192	192	95	95
221	192	95	95
221	28	28	28
234	234	28	95

Fig. 1. Original gray-scale image, where the pixels' intensities are displayed as a number in the range [0,255].

If we consider the gray-scale image of Fig.1, as the resultant of non-overlapped image slices, having specific intensities in their pixels' positions, then we can decompose the original image to several slices, which can reconstruct it, by applying fundamental mathematical operations.

In this sense the image of Fig.1, can be decomposed to the following slices,

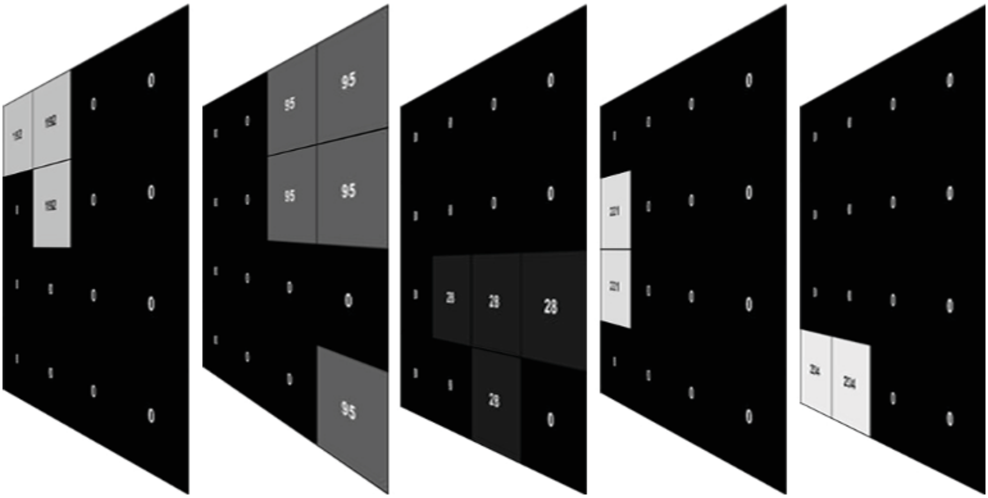


Fig. 2. Image Slice Representation of the image of Fig.1: 1st image slice of 192 intensity, 2nd image slice of 95 intensity, 3rd image slice of 28 intensity, 4th image slice of 221 intensity and 5th image slice of 234 intensity.

As a result of the Definition 1, is the following Lemma 1 and 2:

Lemma 1: Any 8-bit gray-scale image can be decomposed into a maximum of 255 slices, where each slice has pixels of one intensity value and black.

Lemma 2: The binary image as a special case of a gray-scale image consists of only one slice, the binary slice, where only the intensities of 255 and 0 are included.

The main advantage of this image decomposition is by describing a gray-scale image into a set of two-level slices of certain intensity; one can effectively apply the IBR (Papakostas et al., 2007) algorithm to compute the 2-D DCT coefficients of each binary slice and finally the total coefficients of the initial gray-scale, by superposition.

Based on the ISR representation, the intensity function *f(x,y)* of a gray-scale image, can be defined as an expansion of the slices' intensity functions,

$$f\left(x,y\right)=\sum_{i=1}^{n}f_i\left(x,y\right) \tag{4}$$

where n is the number of slices (equal to the number of different intensity values) and $f_i(x,y)$ is the intensity function of the i^{th} slice. In the case of a binary image n is equal to 1 and thus $f(x,y)=f_1(x,y)$.

By using the ISR representation scheme, the computation of the $(p+q)^{th}$ 2-D DCT coefficient (1) of a gray-scale image *f(x,y)*, can be performed according to,

$$
\begin{aligned}
C_{pq} &= \frac{1}{\rho(p)\rho(q)}\sum_{x}\sum_{y} D_p(x)D_q(y)f(x,y) = \frac{1}{\rho(p)\rho(q)}\sum_{x}\sum_{y} D_p(x)D_q(y)\left(\sum_{i=1}^{s} f_i(x,y)\right) \\
&= \frac{1}{\rho(p)\rho(q)}\sum_{x}\sum_{y} D_p(x)D_q(y)*\left(f_1(x,y)+f_2(x,y)+\ldots+f_s(x,y)\right) \\
&= \frac{1}{\rho(p)\rho(q)}\sum_{x}\sum_{y} D_p(x)D_q(y)f_1(x,y) + \frac{1}{\rho(p)\rho(q)}\sum_{x}\sum_{y} D_p(x)D_q(y)f_2(x,y)+\ldots \\
&\quad + \frac{1}{\rho(p)\rho(q)}\sum_{x}\sum_{y} D_p(x)D_q(y)f_s(x,y) \\
&= f_1\left[\frac{1}{\rho(p)\rho(q)}\sum_{x_1}\sum_{y_1} D_p^1(x_1)D_q^1(y_1)\right] + f_2\left[\frac{1}{\rho(p)\rho(q)}\sum_{x_2}\sum_{y_2} D_p^2(x_2)D_q^2(y_2)\right]+\ldots \\
&\quad + f_s\left[\frac{1}{\rho(p)\rho(q)}\sum_{x_s}\sum_{y_s} D_p^s(x_s)D_q^s(y_s)\right] \\
&= f_1C_{pq}^1 + f_2C_{pq}^2 + \ldots + f_sC_{pq}^s
\end{aligned}
\tag{5}
$$

where f_i and C_{pq}^i , *i=1,2,…,s* are the intensity functions of the slices and the corresponding *(p+q)th* order coefficient of the *ith* binary slice, respectively.

The corresponding coefficient of a binary slice C_{pq}^i , is the coefficient computed by considering a block representation of the image (Papakostas et al., 2007), as follows

$$
C_{pq}^i = \sum_{j=0}^{k-1} C_{pq}(b_j) = \sum_{j=0}^{k-1}\sum_{x=x_{1,b_j}}^{x_{2,b_j}}\sum_{y=y_{1,b_j}}^{y_{2,b_j}} D_p(x)D_q(y) = \sum_{j=0}^{k-1}\left(\sum_{x=x_{1,b_j}}^{x_{2,b_j}} D_p(x)\right)\left(\sum_{y=y_{1,b_j}}^{y_{2,b_j}} D_q(y)\right) \tag{6}
$$

where x_{1,b_i}, x_{2,b_i} and y_{1,b_i}, y_{2,b_i} are the coordinates of the block b_i, with respect to the horizontal and vertical axis, respectively.

The proposed methodology has already been applied to the computation of binary and gray-scale images' moments in (Papakostas et al., 2007) and (Papakostas et al., 2008) respectively, with remarkable results.

As a result of the above analysis (5) and (6), is the following Proposition 1,

Proposition 1: The *(p+q)th* order 2-D DCT coefficient of a gray-scale image is equal to the *"intensity-weighted"* sum of the same order coefficients of a number of binary slices.

3.2 Partial Image Block Representation algorithm – PIBR

Once a gray-scale image is decomposed to several slices according to the ISR method presented above, each slice can be considered as a two-level image where the IBR algorithm can be effectively applied.

While the separate application of the IBR algorithm to each slice significantly increases the overall block extraction time, a modified IBR algorithm, which extracts the image blocks in a single pass, is introduced in the sequel. This algorithm is called PIBR (Partial Image Block Representation) and consists of the same steps with the IBR, but with the extra process of extracting blocks of all intensity values.

The PIBR algorithm consists of one pass of the image and a bookkeeping process. More precisely, the algorithm is described by the following steps:

Algorithm – PIBR

Step 1. Consider each line y of the image f and find the object level intervals for each intensity value that exists in line y.

Step 2. Compare the intervals and blocks that have the same intensity in line y-1.

Step 3. If an interval does not match with any block of the same intensity, then it is the beginning of a new block.

Step 4. If a block matches with an interval of the same intensity, the end of the block is in the line y.

■

As a result of the application of the above algorithm, are several sets of rectangular blocks, having the same intensity value. After the blocks extraction, the image can be redefined in terms of blocks of different intensities, instead of individual pixels as,

$$f(x,y)=\left\{f_i\left(x_i,y_i\right):i=1,2,3,..,n\right\} \ where \ \ f_i\left(x_i,y_i\right)=\left\{b_{ij}:j=0,1,...,k_i-1\right\} \quad (7)$$

where n is the number of image slices (maximum 255) and k_i is the number of image blocks having intensity i. Each block is described by two pixels, the upper left and down right corner pixels of the block.

Once the blocks of all slices are extracted, (6) can be applied to compute the 2-D DCT coefficients of each slice and then by using (5) the corresponding coefficient of the original gray-scale image.

It has to be noted that the procedure of extracting image's slices and slice's blocks, does not add significant overhead in the overall process, something which is verified by the experimental study taken place and presented in the next section.

3.3 Complexity analysis

Before proceeding with the experiments to investigate the performance of the proposed methodology in comparison with the conventional one, a detailed analysis of the algorithm's complexity is somewhat necessary in order to study its computational efficiency.

The following analysis, considers the worst case of image's intensities distribution which corresponds to the image called Chessboard defined as: *f(i,j)=0 (black) or f(i,j)=1 (white), f(i,j)≠{f(i-1,j), f(i+1,j), f(i,j-1), f(i,j+1)*.

In the following Table 1, the computational complexity of the proposed algorithm in computing a single 2-D DCT coefficient of an *NxN* image, in the worst case, in comparison with the conventional method, in terms of the number of fundamental operations, is presented.

	Direct Method	**Proposed Method**
Multiplications	$10*N^2+2$	$9*N^2+4$
Additions	$3*N^{*M}$	$3*N^2+1$
Divisions	$2*N^2+1$	$2*N^2+1$
Kernel Computations	$2*N^2$	$2*N^2$

Table 1. Algorithms' computational complexity.

From Table 1, it is clear that the use of the ISR can significantly reduce the number of multiplications required in order to compute a single coefficient, while the rest operations remain quite the same.

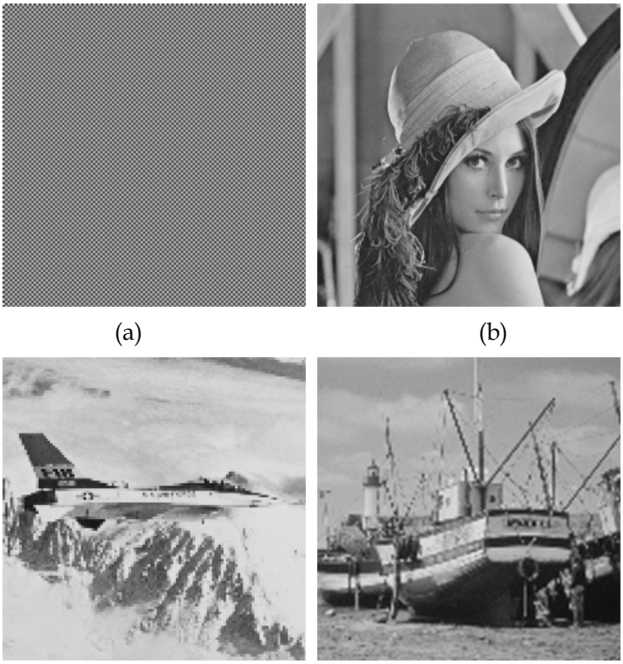
This outperformance, is reinforced in the case of multiple coefficients computation and in arbitrary images where the number of extracted blocks are smaller but with wider area.

4. Experimental study

In order to investigate the computational performance of the proposed methodology, appropriate experiments have taken place, in a 3.2 GHz PC having 1GB RAM, in C++ programming language and their results are presented in this section.

For the sake of the experiments three well known benchmark images and an artificial one have been selected, the Lena, F16, Boat and Chessboard images having 128x128 pixels size, as depicted in the following Fig.3. The images of Fig.3, is in gray-scale format but the same experiments are repeated and for their binary versions, since the usage of binary images are more suitable in real-time pattern recognition systems, where there are constraints on the amount of information that can be processed in real-time sense.

The experiments are performed in two directions: the performance of the proposed computation scheme in computing the 2-D DCT of the whole image, a common practice in pattern recognition applications where the resulted coefficients are used as discriminative features, called global-mode is investigated firstly.



(a) (b)

(c) (d)

Fig. 3. Benchmark images (a) Chessboard, (b) Lena, (c) F16 and (d) Boat in gray-scale format.

In a second state, the efficiency of computing the 2-D DCT in a block-mode, meaning that the algorithm is applied in a finite number of image sub-blocks, which is a common policy in the case of image compression applications such as JPEG, and the dependency of the computation speed on the block size is derived. It has to be remarked that the complexity of the two algorithms, as already presented in section 3.3, is independent of the order *(p,q)* of the coefficient and thus the complexity of computing a certain number of coefficients is multiple of the corresponding complexity of one.

Direction I - *global-mode*

According to this experimental direction, the novel algorithm is applied in the entire image and its performance is compared to the corresponding of the conventional method. In the following Table 2, the computation statistics for the case of the test images are presented in details.

Operation Type	Method	Chessboard Image	Lena Image	F16 Image	Boat Image
Multiplications	***Direct***	163842	163842	163842	163842
	Proposed	147460	139263	136100	138204
Additions	***Direct***	49152	49152	49152	49152
	Proposed	49153	49103	48904	49129
Divisions	***Direct***	32769	32769	32769	32769
	Proposed	32769	31088	30449	30873
Kernel Computations	***Direct***	32768	32768	32768	32768
	Proposed	32768	31087	30448	30872
Total	***Direct***	278531	278531	278531	278531
	Proposed	262150	250655	245901	249078
Total Reduction (%)		**5.88%**	**10.00%**	**11.71%**	**10.57%**

Table 2. Performance (in number of operations) of the algorithms of computing a single DCT coefficient, for the set of all gray-scale test images.

The above results justify the theoretical analysis about the algorithms' complexity presented in section 3.3 and show that the introduced methodology requires less fundamental operations in comparison to the conventional method. Chessboard image represents the worst case image for the ISR since the number of extracted slices is very high equal to the half of image's pixels. Even in the case of the chessboard image the proposed method outperforms the traditional one, by reducing the total operations of about 5.88%, while for the other images this reduction approaches 10% in almost all cases.

Table 3, illustrates some useful performance statistics of applying ISR to the gray-scale images, such as the time needed to extract image's blocks, the number of extracted slices and the total blocks. As it can be seen from this table, the block extraction time is a fast procedure that does not add significant overhead in the whole computation and the number of extracted blocks varies depending on the image's intensities distribution.

	Chessboard Image	Lena Image	F16 Image	Boat Image
Block Extraction Time(msecs)	0.6434	0.6584	0.6912	0.6891
Num. of Blocks	16384	14708	14095	14501
Num. of Intensity Slices	2	205	211	213

Table 3. Performance statistics of applying ISR to the gray-scale benchmark images.

Since DCT coefficients are widely used as discriminative features to pattern classification applications, there is an increasing interest in implementing the process of coefficients extraction in an embedded system having a microprocessor as basic processing unit.

		Chessboard Image	Lena Image	F16 Image	Boat Image
Multiplications	*Direct*	163842	163842	163842	163842
	Proposed	147460	55330	60283	60107
Additions	*Direct*	49152	49152	49152	49152
	Proposed	49153	26881	29322	29152
Divisions	*Direct*	32769	32769	32769	32769
	Proposed	32769	13484	14746	14703
Kernel Computations	*Direct*	32768	32768	32768	32768
	Proposed	32768	13483	14745	14702
Total	*Direct*	278531	278531	278531	278531
	Proposed	262150	109178	119096	118664
Total Reduction (%)		**5.88%**	**60.80%**	**57.27%**	**57.39%**

Table 4. Performance (in number of operations) of the algorithms of computing a single DCT coefficient, for the set of all binary test images.

In these applications the need of real-time processing and resources restrictions (memory resources) constitute major issues of the software developer in implementing any time consuming algorithm such as 2-D DCT. Binary images comprise a good choice that satisfies the two above requirements and thus the study of the proposed algorithm's performance on bi-level images is of significant importance.

In this way, the gray-scale images of Fig.3 are initially converted to binary format by applying the well known Otsu's thresholding algorithm (Otsu, 1979) and the same experiments are executed in the resulted binary images. The corresponding results are presented in the above Table 4.

Table 4, shows that the efficiency of using the proposed method for binary images is significant, since it achieves 60% reduction of the total operations in comparison with the conventional method. This result is justified by the fact that in the case of binary images the number of extracted blocks is low while blocks' area is increased and thus according to (5)-(6), the whole processing is less time consuming.

The corresponding statistics of applying ISR to the binary images are depicted in the following Table 5, where it can be observed that the number of extracted blocks is small indeed.

	Chessboard Image	**Lena Image**	**F16 Image**	**Boat Image**
Block Extraction Time(msecs)	0.6434	0.1966	0.1778	0.1810
Num. of Blocks	16384	1394	1299	1295
Num. of Intensity Slices	2	2	2	2

Table 5. Performance statistics of applying ISR to the binary benchmark images.

Conclusively, it can be declared that the proposed methodology can be effectively applied to accelerate the computation of the 2-D DCT when the transform is applied on the entire image. While the proposed algorithm presents remarkable performance in gray-scale images, in the special case of binary images, the algorithm outperforms the conventional method totally, by making it appropriate for real-time, performance demanding applications.

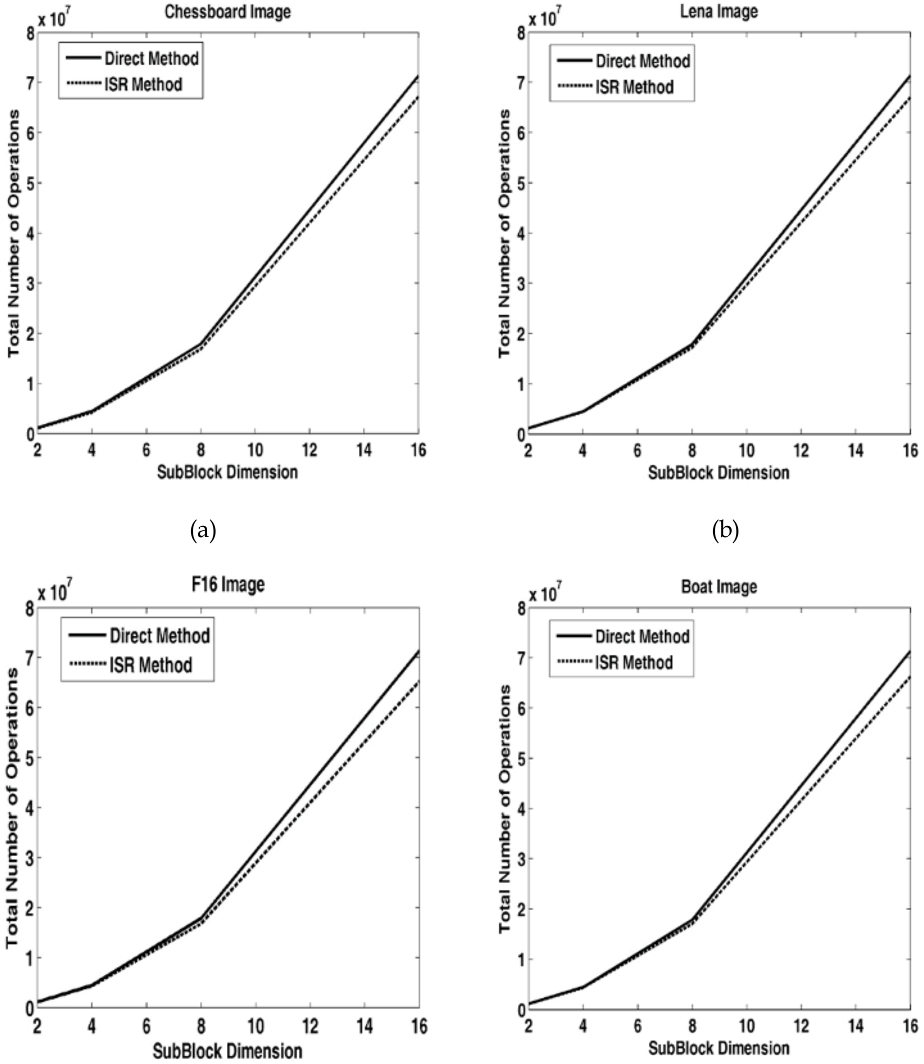
Direction II – *block-mode*

Due to the high popularity of DCT in image compression methods, it is of significant importance to investigate the performance of the proposed methodology when applied on a sub-block of the image (block-mode), as it is performed in the JPEG standard.

As part of the JPEG standard, the 2-D DCT is applied on 8x8 image sub-blocks, but it is important to investigate the behaviour of the method in several dimensions of image's sub-blocks. Therefore, it is decided to study the algorithm for 2x2, 4x4, 8x8, and 16x16, sub-blocks in both gray-scale and binary benchmark images and the results are illustrated in Fig.4 and Fig.5 respectively.

Significant conclusions can be derived from these figures, about the performance of the proposed methodology when it is applied in a block-mode. The computation of 2-D DCT of an image decomposed to sub-blocks of *nxn* size, is accelerated by using the ISR for sub-block size greater than 8x8 considering a gray-scale format, while in binary images this acceleration occurs earlier.

Therefore, the use of the proposed methodology in the case of binary images improves the overall computation rate but when applied on gray-scale images the advantages of the method are restricted for sub-blocks greater than the 8x8 size, which is the official size used in the standard JPEG. This fact generates some questions about the efficiency of the introduced algorithm, since the usefulness of sub-block size greater than 8x8 in JPEG compression has to be studied.



(a)

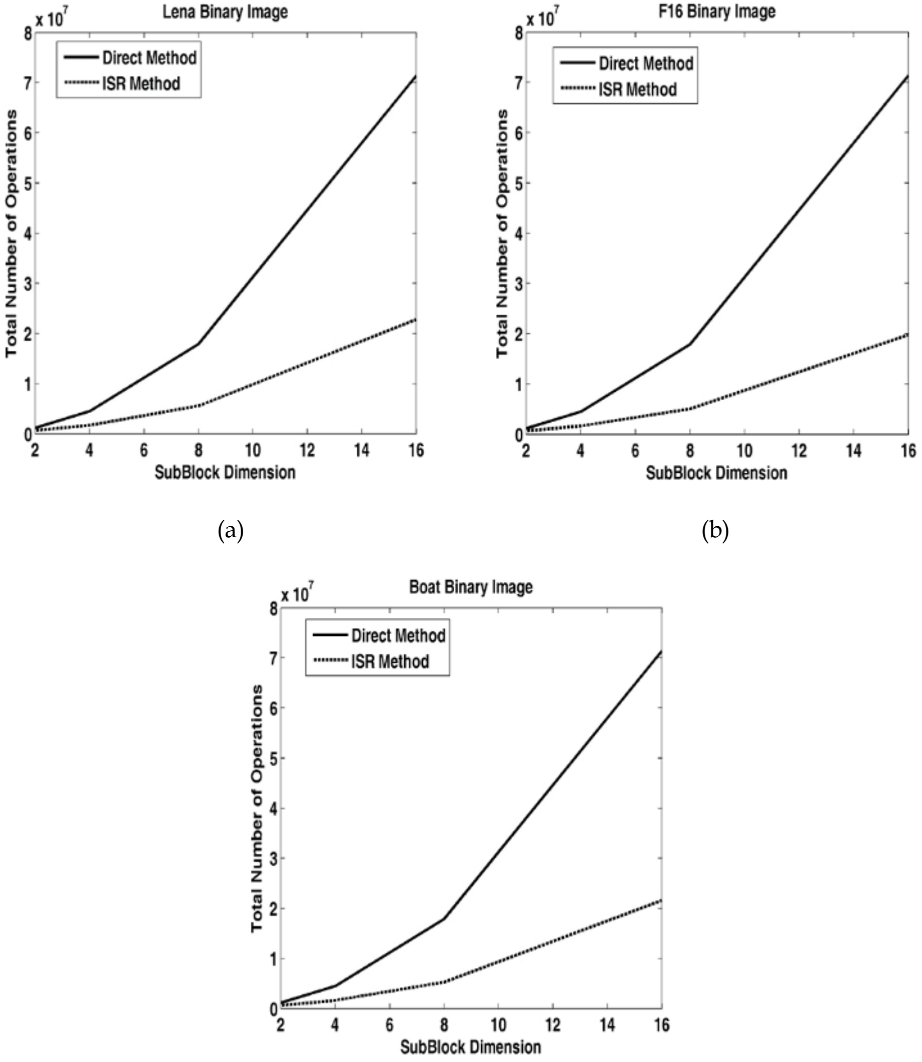
(b)

(c)

(d)

Fig. 4. Total operations in respect to sub-block size for (a) Chessboard, (b) Lena, (c) F16 and (d) Boat gray-scale images.

For this reason, it is decided to investigate the compression performance of the JPEG under variable sub-block sizes. The compression ratio for a fixed image quality (MSE – Mean Squared Error) and for 2x2, 4x4, 8x8 and 16x16 sub-block size is computed and the results are displayed in the following Fig.6. These results testify that the 8x8 sub-block size is the optimum choice regarding the compression ratio and the image quality, an observation which deals with the JPEG standard.



(a)

(b)

(c)

Fig. 5. Total operations in respect to sub-block size for (a) Lena, (b) F16 and (c) Boat binary images.

5. Conclusion

A novel methodology that ensures the computation of 2-D DCT coefficients in gray-scale images as well as in binary ones, with high computation rates, was presented in the previous sections. Through a new image representation scheme, called ISR (Image Slice Representation) the 2-D DCT coefficients can be computed in significantly reduced time, with the same accuracy.

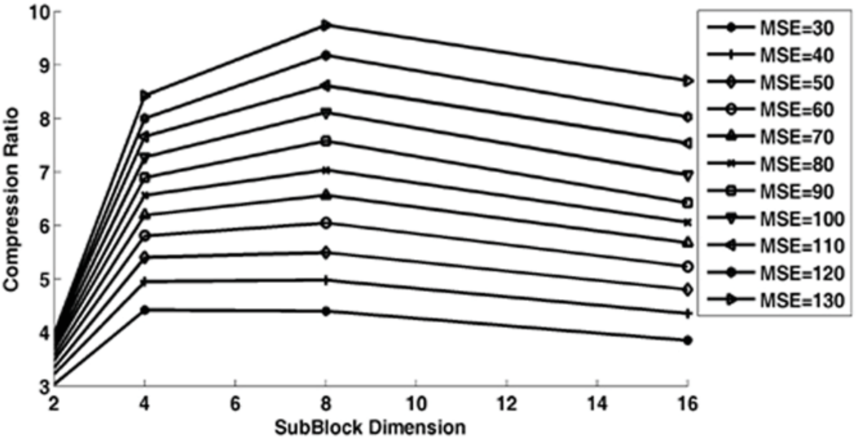


Fig. 6. Compression performance of JPEG for various sub-block sizes.

Appropriate experiments prove that the new algorithm totally outperforms the conventional method, in the case the algorithm applied on the entire image area for any image type, while in its blocked version the advantages were obvious for sub-block sizes greater than 8x8.

The experimental results are very promising and in conjunction to the capability of the methodology to collaborate with any of the existent fast algorithms, it can be a good choice for real-time and time demanding applications.

6. References

Abe, Y. & Iiguni, Y. (2007). Image restoration from a downsampled image by using the DCT. *Signal Processing*, Vol. 87, No. 10, pp. 2370–2380.

Ahmed, N.; Natarajan, T. & Rao, K.R. (1974). Discrete cosine transform. *IEEE Trans. Computers*, Vol. 23, No. 1, pp. 90-93.

Alturki, F.T.; Almutairi, A.F. & Mersereauu, R.M. (2007). Analysis of blind data hiding using discrete cosine transform phase modulation. *Signal Processing: Image Communication*, Vol. 22, No. 4, pp. 347-362.

Choi, H.J.; Seo, Y.H.; Yoo, J.S. & Kim, D.W. (2008). Digital watermarking technique for holography interference patterns in a transform domain. *Optics and Lasers in Engineering*, Vol. 46, No. 4, pp. 343-348.

Diab, C.; Oueidat, M. & Prost, R. (2002). A new IDCT-DFT relationship reducing the IDCT computational cost. *IEEE Trans. Signal Processing,* Vol. 50, No. 7, pp. 1681-1684.

Er, M.J.; Chen, W. & Wu, S. (2005). High-Speed face recognition based on discrete cosine transform and RBF neural networks. *IEEE Trans. on Neural Networks*, Vol. 16, No. 3, pp. 679-691.

ISO/IEC 10918-1, (1994). Digital compression and coding of continuous-tone still images: Requirements and guidelines.

Jadhav, D.V. & Holambe, R.S. (2008). Radon and discrete cosine transforms based feature extraction and dimensionality reduction approach for face recognition. *Signal Processing*, Vol. 88, No. 10, pp. 2604-2609.

Krupinski, R. & Purczynski, J. (2007). Modeling the distribution of DCT coefficients for JPEG reconstruction. *Signal Processing: Image Communication*, Vol. 22, No. 5, pp. 439-447.

Liu, Z. & Liu, C. (2008). Fusion of the complementary Discrete Cosine Features in the YIQ color space for face recognition. *Computer Vision and Image Understanding,* Vol. 111, No. 3, pp. 249-262.

Nikara, J.A.; Takala, J.H. & Astola, J.T. (2006). Discrete cosine and sine transforms-regular algorithms and pipeline architectures. *Signal Processing*, Vol. 86, No. 2, pp. 230–249.

Otsu, N. (1979). A threshold selection method from gray-level histograms. *IEEE Trans. Systems Man Cybernet.*, Vol. 9, pp. 62– 66.

Papakostas, G.A.; Karakasis, E.G. & Koulouriotis, D.E. (2008). Efficient and accurate computation of geometric moments on gray-scale images. *Pattern Recognition*, Vol. 41, No. 6, pp. 1895-1904.

Papakostas, G.A.; Karakasis, E.G. & Koulouriotis D.E. (2007). Exact and speedy computation of Legendre moments on binary Images. *Proceedings of 8th International Workshop on Image Analysis for Multimedia Interactive Services (WIAMIS'07)*, p.48, ISBN, Santorini – Greece, 2007.

Plonka, G. & Tasche, M. (2005). Fast and numerically stable algorithms for discrete cosine transforms. *Linear Algebra and its Applications*, Vol. 394, pp. 309–345.

Qi, H.; Zheng, D. & Zhao, J. (2008). Human visual system based adaptive digital image watermarking. *Signal Processing*, Vol. 88, No. 1, pp. 174-178.

Sanderson, C. & Paliwal, K.K. (2003). Features for robust face-based identity verification. *Signal Processing*, Vol. 83, No. 5, pp. 931-940.

See, K.W.; Loke, K.S.; Lee, P.A. & Loe, K.F. (2007). Image reconstruction using various discrete orthogonal polynomials in comparison with DCT. *Applied Mathematics and Computation*, Vol. 193, No. 2, pp. 346-359.

Shao, X. & Johnson, S.G. (2008). Type-IV DCT, DST, and MDCT algorithms with reduced numbers of arithmetic operations. *Signal Processing,* Vol. 88, No. 6, pp. 1313–1326.

Tan, T.C.; Bi, G.; Zeng, Y. & Tan, H.N. (2001). DCT hardware structure for sequentially presented data. *Signal Processing*, Vol. 81, No. 11, pp. 2333–2342.

Wikipedia contributors, Discrete cosine transform, Wikipedia, The Free Encyclopedia, http://en.wikipedia.org/w/index.php.

Wong, K.; Qi, X. & Tanaka, K. (2007). A DCT-based Mod4 steganographic method. *Signal Processing*, Vol. 87, No. 6, pp. 1251-1263.

Wu, J.S.; Shu, H.Z.; Senhadji, L. & Luo, L.M. (2008). A fast algorithm for the computation of 2-D forward and inverse MDCT. *Signal Processing*, Vol. 88, No. 6, pp. 1436–1446.

Zeng, Y.; Cheng, L.; Bi, G. & Kot, A.C. (2001). Integer DCTs and Fast Algorithms. *IEEE Tans. Signal Processing*, Vol. 49, No. 11, pp. 2774-2782.

Image Processing

Edited by Yung-Sheng Chen

ISBN 978-953-307-026-1
Hard cover, 516 pages
Publisher InTech
Published online 01, December, 2009
Published in print edition December, 2009

There are six sections in this book. The first section presents basic image processing techniques, such as image acquisition, storage, retrieval, transformation, filtering, and parallel computing. Then, some applications, such as road sign recognition, air quality monitoring, remote sensed image analysis, and diagnosis of industrial parts are considered. Subsequently, the application of image processing for the special eye examination and a newly three-dimensional digital camera are introduced. On the other hand, the section of medical imaging will show the applications of nuclear imaging, ultrasound imaging, and biology. The section of neural fuzzy presents the topics of image recognition, self-learning, image restoration, as well as evolutionary. The final section will show how to implement the hardware design based on the SoC or FPGA to accelerate image processing.

InTech Europe
University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China
Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821