International Journal of Computer and Information Technology (ISSN: 2279 – 0764)
Volume 02– Issue 04, July 2013

Extraction of Region of Interest in Compressed Domain

Vaishali A.Choudhary
Computer science and engineering
Ramdeobaba college of engineering and management.
Nagpur, India
e-mail: khushboo5sep {at} gmail.com

Preeti Voditel
Computer science and engineering
Ramdeobaba college of engineering and management.
Nagpur, India

Abstract— **Image transforms are extensively used in image processing and image analysis. Transform is basically a mathematical tool, which allows us to move from one domain to another domain .Transforms play a significant role in various image processing applications such as image analysis, image enhancement, image filtering and image compression.**

Nowadays, almost all digital images are stored in compressed format in order to save the computational cost and memory. To save the memory cost, all the image processing techniques like feature extraction, image indexing and watermarking techniques are applied in the compressed domain itself rather than in spatial domain. In this paper, for compression purpose, Discrete Cosine Transform (DCT) is used because it has excellent energy compaction. The new approach devised in this paper is that, if we will be able to find the relationship between the coefficients of a block to all of its sub-blocks in the DCT domain itself, without decompressing it then time to extract global features in compressed domain for general image processing tasks will gets minimized.

In this paper, composition of a block from all of its sub-blocks and vice versa ,directly in DCT domain is obtained and showed that the result of both operations are same and the computational complexity of the proposed algorithm is lower than that of the existing ones.

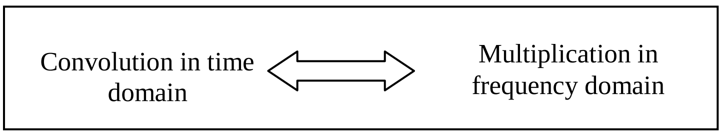
Keywords-compressed domain processing, Discrete Cosine Transform (DCT), spatial relationship.

I. INTRODUCTION

Over the years, image processing has been developed primarily in the pixel domain. As digital images stored in computers are increasingly in discrete cosine transform (DCT), direct image processing in compressed domain is a new area for research [6], [8]-[10], [11], [12].

KLT (Karhunen-Loeve transform) is the most efficient transform in terms of energy compaction. Since basis functions of KLT are image dependant, it leads to expensive computing costs. To overcome this problem, DCT is developed since it is signal independent and more compaction is achieved in small number of coefficients. Compression is achieved by applying any transform on signal so following are some reasons why we need transform domain operations.

A. Mathematical Convenience



Every action in time domain will have an impact in the frequency domain .The complex convolution operation in the time domain is equal to simple multiplication in the frequency domain.

B. To Extract More Information

Transforms allows us to extract more relevant information. To illustrate this, consider the following example:

Person X is on the left hand side of the prism, whereas the person Y is on the right hand side of the prism. Person X sees the light as white light whereas the person Y sees the white light as a combination of seven colors. Obviously, the person Y is getting more information than the person X by using the prism. Similarly, a transform is a tool that allows one to extract more information from a signal.

Image transform is basically a representation of an image. There are two reasons for transforming an image from one representation to another. First, the transformation may isolate critical components of the image pattern so that they are directly accessible for analysis. Second, the transformation may place the image data in a more compact form so that they can be stored and transmitted efficiently.

C. Why DCT

It is a fast transform. It requires real operations. It is useful in designing transform coders and Weiner filters for images. We are going to use this transform because it has excellent energy compaction for images.

1) Properties of DCT

We are going to use DCT because of following two properties:

a) Decorrelation

As discussed previously, the principle advantage of image transformation is the removal of redundancy between neighboring pixels. This leads to uncorrelated transform coefficients which can be encoded independently. Clearly, the amplitude of the autocorrelation after the DCT operation is very small at all lags. Hence, it can be inferred that DCT exhibits excellent de-correlation properties.

b) Energy Compaction

Efficacy of a transformation scheme can be directly gauged by its ability to pack input data into as few coefficients as possible. This allows the quantizer to discard coefficients with relatively small amplitudes without introducing visual distortion in the reconstructed image. DCT exhibits excellent energy compaction for highly correlated images.

II. RELATED WORK

We know that most of image processing operations are developed in pixel domain. It is not only time consuming but increases computational complexity. Therefore, nowadays more research is going on image processing in compressed domain or DCT domain.

Qionghai Dai et.al.[2] proposed an efficient algorithm for computing one-dimensional (1-D) discrete cosine transform (DCT) for a signal block, given its two adjacent sub-blocks in the DCT domain and then introduce several algorithms for the fast computation of multidimensional (m-D) DCT with size N_1 N_2 …. N_m given 2^m sub-blocks of DCT coefficients with size $N_1/2$ $N_2/2$ … $N_m/2$, where N_i (i=1, 2, …. , m) are powers of 2. Besides, they have proposed a direct method by dividing the data into 2^m parts for independent fast computation, in which only two steps of r-dimensional (r = 1,2, . . ., m) IDCT and additional multiplications and additions are required. If all the dimensional sizes are the same, the number of multiplications required for the direct method is only $(2^m - 1)/$ times of that required for the row-column method, and if N , the computational efficiency of the direct method is surely superior to that of the traditional method, which employs the most efficient multidimensional DCT/IDCT algorithms.

Athanassios [3] proposed an efficient direct method for the computation of a length-N discrete cosine transform (DCT) given two adjacent length-(N=2) DCT coefficients, is presented in this letter. The computational complexity of the proposed method is lower than the traditional approach for lengths N > 8. Savings of N memory locations and 2N data transfers are also achieved.

Weidong Kou et.al. [4] Proposed a direct computational algorithm for obtaining the DCT coefficients of a signal block taken from two adjacent blocks is proposed. This algorithm reduces the number of multiplications and additions/ subtractions compared to the traditional method, which requires inverse transforms of two received coefficient blocks followed by a forward transform.

Ephraim Feig et.al.[5] introduced several fast algorithms for computing discrete cosine transforms (DCT's) and their inverses on multidimensional inputs of sizes which are powers of 2. They have also presented algorithms for computing scaled DCT's and their inverses;these have applicationns in compression of continuous tone image data,where the DCT is generally followed by scaling and quantization.

Shen *et al.* [9] proposed an algorithm for the edge extraction directly from the DCT domain. They used an ideal edge model to estimate the strength and orientation of an edge in terms of the relative values of different DCT coefficients within each data block. The experimental results show that the coarse edge information of images extracted in the DCT domain are almost 20 times faster than conventional edge detectors in the pixel domain. Similarly, Abdel-Malek *et al.* [8] have proposed a technique to detect oriented line features using DCT coefficients. A segmentation technique [6] using the local variance of DCT coefficients was proposed by Ng *et al.* in which the 3 3 DCT is computed at each pixel location using the surrounding points.

Smith *et al.* [7] proposed a feature extraction method based on the 16 DCT coefficients of 4 4blocks.The variance and the mean absolute values of each of these coefficients are computed over the whole image. The feature of the entire image is then represented by this 32-component vector, and the feature vector is further processed (such as dimension reduction) to produce indexing keys. Reeves *et al.* [10] also proposed similar indexing techniques in the DCT domain, where block size considered in their work is 8 8.

III. NEED OF COMPRESSED DOMAIN OPERATIONS

While it is emerging that increasingly more image processing algorithms are developed in the compressed or DCT domain to take advantage of reducing the computing cost and improving the processing speed, a new problem could occur from the fact that various DCT block sizes have to be used in order to ensure optimized performances. These include 8 8 blocks used in JPEG, 4 4 blocks used in image indexing, and 16 16 macro-blocks in MPEG. To deal with inter-transfer of DCT coefficients from different blocks with various sizes, the existing approach would have to decompress the pixel data in the spatial domain via the IDCT first and re-divide the pixels into new blocks with the required size to apply the DCT again and produce the DCT coefficients. It is obvious that the approach is inefficient. To this end, direct derivation of DCT coefficients for those blocks with various sizes can be made possible if the spatial relationship is fully revealed and analyzed.

IV. POSSIBLE APPLICATIONS

A. Extraction of Region of Interest in Transform Domain Itself

If we have an image block which is not initially divided into blocks of any size, then we will divide it into small sub-blocks of given size. If any block is not completely inside that boundary then we will divide that block until we get our region of interest completely inside the selected region. With the help of relationship between the coefficients of DCT of a block & the DCT of its sub-blocks in DCT domain itself, it is possible to compute coefficients of any block from all of its sub-blocks and vice versa. In order to save the computation and memory cost, it is desirable to have image processing operations such as feature extraction, image indexing, and pattern classifications implemented directly in the DCT domain. It is useful in extracting global features in compressed domain for general image processing tasks such as those widely used in pyramid algorithms and image indexing.

In addition, due to the fact that the corresponding coefficient matrix of the linear combination is sparse, the computational complexity of the proposed algorithms is significantly lower than that of the existing methods.

B. Insertion of Captions and Logos

We can extend this theory of inter-transfer of DCT coefficients, for insertion of captions and logos in compressed domain itself. It is often used in images to provide additional information or to point out specifics. We can add a caption and identifying figure number to charts, smart art and pictures.

Everyday there are more videos on the net. The "Video Copy Detection" is based on detecting video copies from a video sample. Thus, we can avoid copyright violations.

We also have to consider as copied video those videos which have been recorded with a camcorder, for example, in the cinema.

We have to be aware that videos can be modified. They can have a logo, some color transforms, black borders, quality decreasing, etc.

The watermarking technique is used to avoid copyright of the images and videos.

V. ROI EXTRACTION STEPS

A. Boundary Detection

It can be done by:

- Impoly
- Imrect
- Roipoly
- Getline functions.

Among these, we have used roipoly and getline functions.

1) Roipoly

Specify polygonal region of interest (ROI).

Use roipoly to specify a polygonal region of interest (ROI) within an image. Roipoly returns a binary image that you can use as a mask for masked filtering.

BW = roipoly creates an interactive polygon tool, associated with the image displayed in the current figure, called the target image. With the polygon tool active, the pointer changes to cross hairs when we move the pointer over the image in the figure. Using the mouse, you specify the region by selecting vertices of the polygon. You can move or resize the polygon using the mouse. When you are finished positioning and sizing the polygon, create the mask by double-clicking or by right-clicking inside the region and selecting create mask from the context menu. Roipoly returns the mask as a binary image, BW, the same size as I. In the mask image, roipoly sets pixels inside the region to 1 and pixels outside the region to 0.

Syntax

BW = roipoly (I, c, r), where I is input image, and vectors c and r represents the X and Y coordinates of selected polygon.

2) Getline

Select polyline with mouse.

Syntax

[x, y] = getline (closed), lets you select a polyline of polygon until it is closed.

B. ROI Extraction in Spatial Domain

Figure 1. Input image

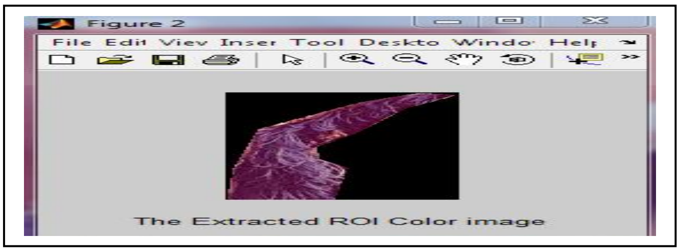


Figure 2. ROI extracted in spatial domain

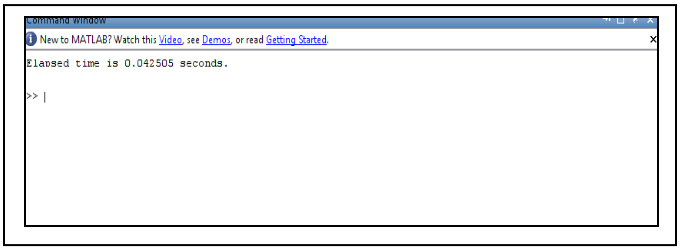


Figure 3. Time required to extract ROI in spatial domain

Fig. 1 shows input image which is Lena image. Fig. 2 shows Region of Interest (ROI) extracted in spatial domain and fig. 3 shows the time required to extract that ROI.

C. Mathematical Derivation for Mapping of a Block to its Sub-blocks

Given a block of pixels, *B,* its number of rows and columns can be represented as a product of two integers such as *L N* rows and *M N* columns, in order to obtain a convenient representation of its sub-blocks. Correspondingly, this block *B* can be divided into *L M* sub-blocks represented as with the size of *N* x *N* (*l =0, 1,..., L-1, m=0,1, . .., M-1)* pixels. Assuming that the DCT coefficients of the block *B* and its sub-blocks are represented as C_B and (u ,v) respectively.(l=0,1,…, *L-1; m=,1,..., M-1;* u ,v=0,1 ,... *N-1),* the problem to be formulated is to determine the spatial relationship between C_B, and (u, v), i.e., the relationship between the DCT coefficients of the block *B* and that of its sub - blocks .

A 2-dimensional block *B* with *LN* rows and *MN* columns would have its DCT being defined as follows:

$$C_B(u,v) = \sqrt{\frac{4}{LN*MN}}\,\alpha(u)\alpha(v)\sum_{i=0}^{LN-1}\sum_{j=0}^{MN-1} x(i,j)\cos\left(\frac{(2i+1)u\pi}{2LN}\right)\cos\left(\frac{(2j+1)v\pi}{2MN}\right) \quad (1)$$

Where,

Here, for convenience, we normalize α *(u)* =1 for all u. For the image block B_{lm}, its corresponding DCT can be expressed as

$$C_{lm}(u,v) = DCT(B_{lm}) = \frac{2}{N}\sum_{i=0}^{N-1}\sum_{j=0}^{N-1} x(lN+i, mN+j)\cos\left(\frac{(2i+1)u\pi}{2N}\right)\cos\left(\frac{(2j+1)v\pi}{2N}\right)$$

(l=0,1,…….,L-1;m=0,1,….,M-1;u,v=0,1,……,N-1). (2)

Denote 2-D basis functions co!

as (i, j). In fact, it is the product of 2 1-D basis functions

(i) = and (j)

In the same domain as above, we reconstruct new basis functions as follows.

$$\Psi_{uv}(i,j) = \begin{cases} \cos\left(\frac{2(i \bmod N)(u \bmod N)\prod}{2N}\right)\cos\left(\frac{2(j \bmod N}{}\right. & \text{When } [i/N]=[u/N] \text{ And } [j/N]=[v/N] \\ 0 & \text{Otherwise} \end{cases}$$

(3)

Similarly, which can also expressed as the product of 2 1-D basis functions

= When [i /N]=[u/N]
Otherwise

(i, u = 0,1 ,…., LN-1) (4)

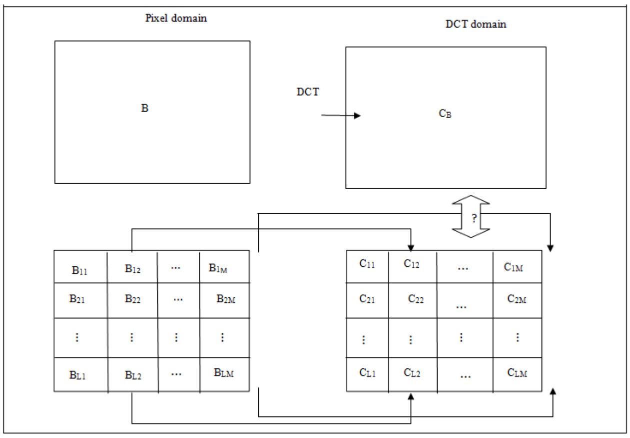


Figure 4. Schematic illustration of problem to be solved

$$\Psi_v^2(j) = \begin{cases} \text{c(} & \text{When } [j/N]=[v/N] \\ & \text{Otherwise} \end{cases}$$

(j, v= 0,1,….,LM-1) (5)

According to the definition of the basis functions Ψ_{uv}, Ψ_u^1 and Ψ_v^2. We have

$$\sum_{i=0}^{LN-1}\sum_{j=0}^{MN-1} x(i,j)\Psi_{uv}(i,j) = \sum_{i=0}^{LN-1}\sum_{j=0}^{MN-1} x(i,j)\Psi_u^1(i)\Psi_v^2(j) = \frac{N}{2}C_{[u/N]} \quad (6)$$

And furthermore, because (i) and (i) are 1-D basis functions in the same space, (i) can be represented the linear combination of (i) as follows .

$$\Phi_u^1(i) = \sum_{k=0}^{LN-1}\beta_k^u\Psi_k^1(i)\ (u = \quad (7)$$

where are the parameters with respect to and , which can be uniquely determined by substituting the value of i (i=0,l,. .. , *LN-1).* Similarly for (j),

(j). (v= 0, 1, … , MN-1) (8)

Denote matrices { } and { } as $\mathbf{D}_{LN \quad LN}$ and $\mathbf{F}_{MN \quad MN}$. Substituting the results obtained in equation (7) and (8) into equation (l), and then equation (1) can be rearranged into equation (9) and then to (10).

(9)

$$= \sqrt{\frac{4}{LN*MN}} \sum_{i=0}^{LN-1} \sum_{j=0}^{MN-1} \left(x(i,j) (\beta_0^u \quad \beta_1^u \quad \cdots \quad \beta_{LN-1}^u) \begin{pmatrix} \Psi_0^1(i) \\ \Psi_1^1(i) \\ \vdots \\ \Psi_{LN-1}^1(i) \end{pmatrix} (\Psi_0^2(j) \quad \Psi_1^2(j) \quad \cdots \right.$$

$$= \sqrt{\frac{4}{LN*MN}} (\beta_0^u \quad \beta_1^u \quad \cdots \quad \beta_{LN-1}^u) \sum_{i=0}^{LN-1} \sum_{j=0}^{MN-1} \left(x(i,j) \begin{pmatrix} \Psi_0^1(i)\Psi_0^2(j) & \Psi_0^1(i)\Psi_1^2(j) & \cdots & \Psi_0^1(i)\Psi_{MN-1}^2(j \\ \Psi_1^1(i)\Psi_0^2(j) & \Psi_1^1(i)\Psi_1^2(j) & \cdots & \Psi_1^1(i)\Psi_{MN-1}^2(j \\ \vdots & \vdots & \cdots & \vdots \\ \Psi_{LN-1}^1(i)\Psi_0^2(j) & \Psi_{LN-1}^1(i)\Psi_1^2(j) & \cdots & \Psi_{LN-1}^1(i)\Psi_{MN-1}^2 \end{pmatrix} \right.$$

Exploiting the results in equation (6) into above equations, we have

$$C_B(u, v) = \sqrt{\frac{1}{L*M}} (\beta_0^u \quad \beta_1^u \quad \cdots \quad \beta_{LN-1}^u) \begin{pmatrix} C_{0,0}(0,0) & \cdots & C_{0,0}(0,N-1) & C_{0,M-1}(0,0) & \cdots & C_{0,M-1}(0,N-1) \\ \vdots & \cdots & & \vdots \ \cdots \ \vdots & \cdots & \vdots \\ C_{0,0}(0,0) & \cdots & C_{0,0}(N-1,N-1) & C_{0,M-1}(0,0) & \cdots & C_{0,M-1}(N-1,N- \\ C_{L-1,0}(0,0) & \cdots & C_{L-1,0}(0,N-1) & C_{L-1,M-1}(0,0) & \cdots & C_{L-1,M-1}(0,N-1 \\ \vdots & \cdots & \vdots & \vdots & \cdots & \vdots \\ C_{L-1,0}(0,0) & \cdots & C_{L-1,0}(N-1,N-1) & C_{L-1,M-1}(0,0) & \cdots & C_{L-1,M-1}(N-1,N- \end{pmatrix} \quad (10)$$

Applying the form of matrix and block matrix, the equation can be shown in concise form as follows.

$$C_B(u, v) = \sqrt{\frac{1}{LM}} D \begin{pmatrix} C_{0,0} & C_{0,1} & \cdots \\ C_{1,0} & C_{1,1} & \cdots \\ \vdots & \vdots & \cdots \\ C_{L-1,0} & C_{L-1,1} & \cdots \end{pmatrix} \quad (11)$$

Where **D** and **F** are square matrices of the parameters with dimensions LN X LN and MN *X MN* respectively, which can be uniquely determined by equation (7) or (8) in advance since they are only relevant to those basis functions. Note that each element represents the set of DCT coefficients for sub-block , and thus itself is a matrix with N x N elements. For the special case of *L=M,* we have: **D=F,** and hence equation (11**)** can be further simplified as:

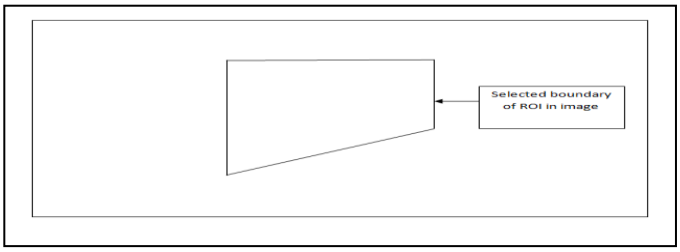
C_B = (12)

Finally, Since there exists inverse matrix for matrices D or F, the block matrix { }also can be expressed into the linear combination of CB .This means, for any block of DCT coefficients, the DCT of its sub-blocks can also be obtained directly in DCT domain. The result reveals that any DCT block can be decomposed into sub-blocks in an iterative manner similar to pyramid algorithms**.**

D. Our Application of Block to Sub-block Theory

Figures 5 to 7 illustrate how to use the relationship of DCT coefficients exchanges between a block and its sub-blocks. The region surrounded by black lines is selected polygon or boundary of the ROI. This boundary is detected with the help of mouse and this has been possible with the getline function in addition to roipoly function. Among impoly, imrect and roipoly functions, we have used roipoly function because with help of this, we can choose any region exactly what we want. For example, if we want to select hairs of a girl of a "Lena" image and if we would use imrect function, the rectangle surrounding hair portion will also include some extra region which we don't want. So we have used roipoly function to detect the boundary. In our proposed work, we have devised two methods to compare time requirements. In the first method, initially, obtain the boundary of a region of an image as shown in figure 5. After extracting that region, obtain the DCT of it by using 2 sub-blocks which is shown by figure 6. Again obtain the original cropped image by using IDCT and again obtain DCT of it by using 4 4 sub-blocks (shown in fig. 7). It is obvious that this procedure is inefficient and time required to extract that region will also be more. Therefore, direct derivation of DCT coefficients for those blocks with various sizes can be made possible if we will be able to find out the spatial relationship between the coefficients of different block sizes. Our approach for ROI extraction is as follows:

Instead of going back to pixel domain with the help of IDCT, we can obtain coefficients of any block sizes (for example, from 2 2 to 4 4) in DCT domain only. This will be possible with the help of (12) and by applying the the DCT coefficients exchange example as shown in figure 11. After extracting ROI's by using both the methods, compare the time requirements of both, respectively. In this way, you will find lower computational complexity, lower computing cost and also the time required in DCT domain will be less than that of those in pixel domain i.e. by IDCT method.



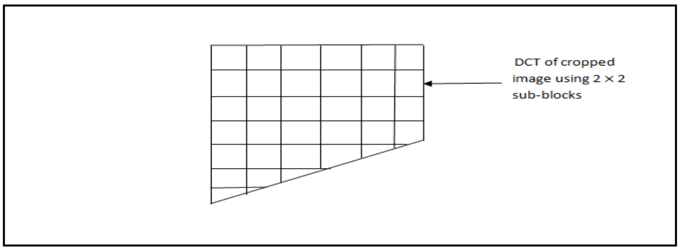


Figure 5. Boundary detection of ROI in image

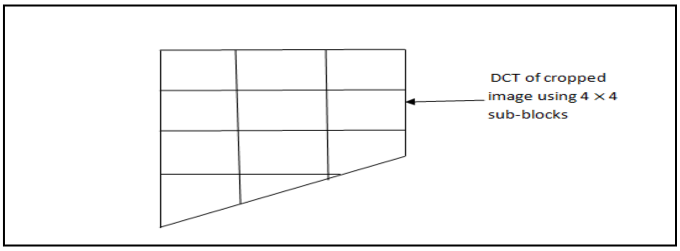


Figure 6. DCT of ROI obtained using 2 2 sub-blocks

Figure 7. DCT of ROI obtained using 4 4 sub-blocks

E. Results obtained using Proposed Method

We have applied the example of DCT coefficients exchange given in figure 14 to “Lena” image and to Region of Interest (ROI) and results obtained are given below:

Fig. 9 is input image from which we can extract any region for applying DCT coefficients exchange concept. After extracting ROI, resize it, so that it should be properly divided into blocks of 2 2 or 4 4. Fig. 10 shows the ROI extracted and resized ROI is shown by Fig. 11. DCT of ROI using 2 2 sub-blocks is shown in figure 12 and IDCT of same is shown in figure 13. Figure 14 shows the DCT of ROI using 4 4 sub-blocks. Direct conversion of ROI from 2 2 sub-blocks to 4 4 sub-blocks in DCT domain is shown in figure 15. Figure 16 shows the coefficient matrix of ROI obtained from 4 4 sub-blocks and it is same to matrix shown in figure 17 which is the coefficient matrix of ROI obtained from 2 2 to 4 4 sub-blocks in DCT domain itself. After comparing matrices, we have concluded that these two are same. Figure 18 shows the time requirements of both the methods in the form of t1 and t2 respectively. So, we have shown that time requirement of proposed method is 4 to 5 times less than that of existing methods i.e. using IDCT.

Fig. 9 Lena image

Figure 10. ROI extracted from

Figure 11. ROI resized

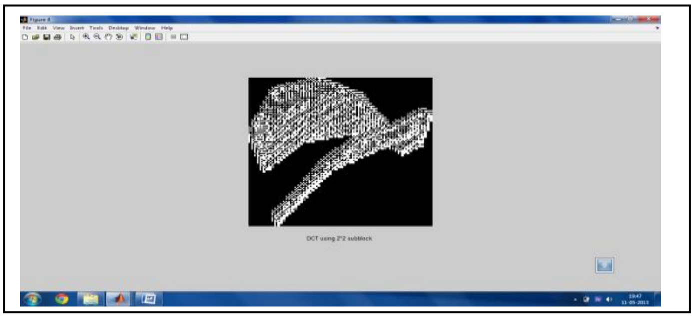


Fig. 12 DCT of ROI using 2 2 sub-blocks

Figure 13. IDCT of ROI using 2 2 sub-blocks

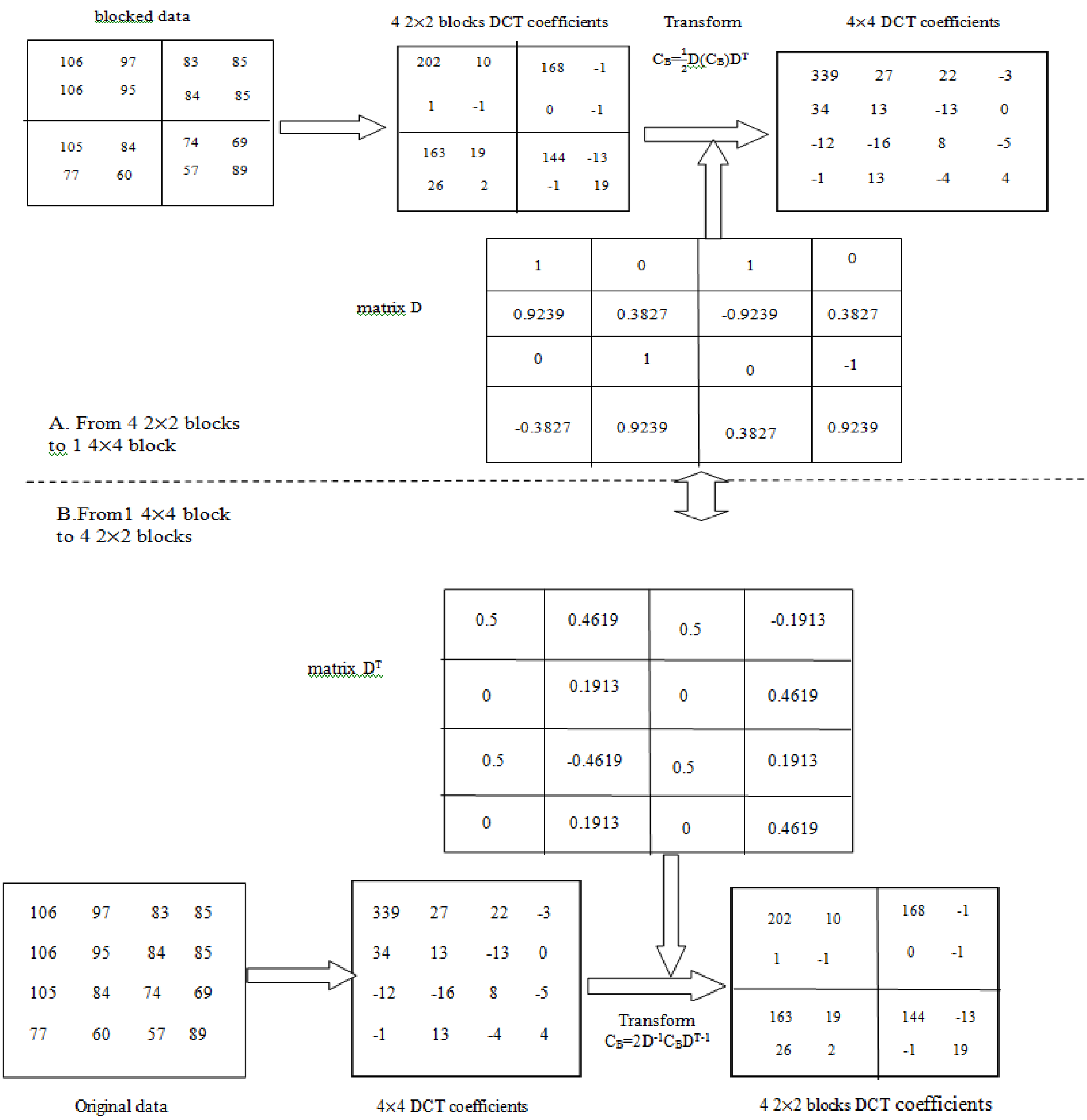


Fig.ure 14. The example for DCT coefficients exchange between 4 2 2 blocks and 1 4 4 block

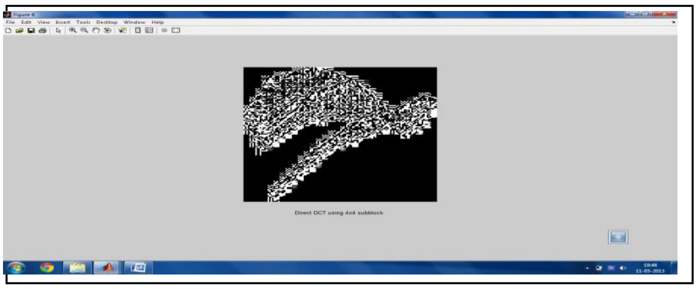


Fig ure 15. DCT of ROI using 4 4 sub-blocks

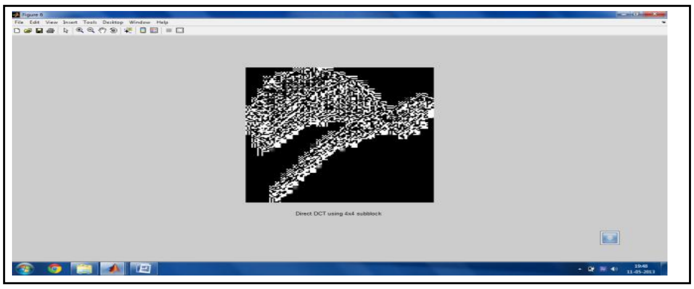
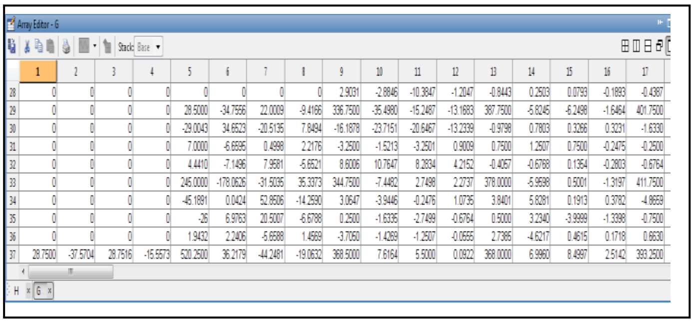


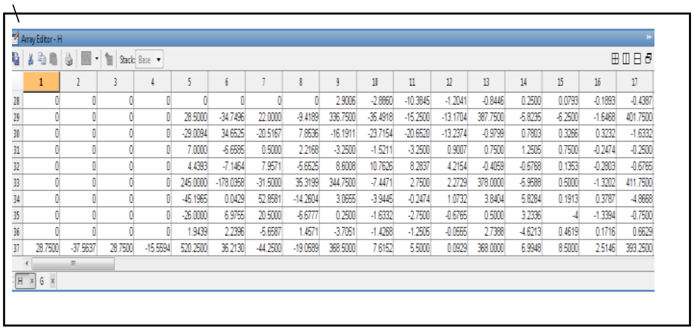
Figure 16. Conversion of ROI from 2 2 sub-blocks to 4 4 sub-blocks in DCT domain



Array Editor - G

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
28	0	0	0	0	0	0	0	0	2.9031	-2.8846	-10.3847	-1.2047	-0.8443	0.2503	0.0793	-0.1893	-0.4387
29	0	0	0	0	28.5000	-34.7556	22.0009	-9.4166	336.7500	-35.4980	-15.2487	-13.1683	387.7500	-5.8245	-6.2498	-1.6464	401.7500
30	0	0	0	0	-29.0043	34.6523	-20.5135	7.8494	-16.1878	-23.7151	-20.6467	-13.2339	-0.9798	0.7803	0.3266	0.3231	-1.6330
31	0	0	0	0	7.0000	-6.6595	0.4998	2.2176	-3.2500	-1.5213	-3.2501	0.9009	0.7500	1.2507	0.7500	-0.2475	-0.2500
32	0	0	0	0	4.4410	-7.1496	7.9581	-5.6521	8.6006	10.7647	8.2834	4.2152	-0.4067	-0.6768	0.1354	-0.2803	-0.6764
33	0	0	0	0	245.0000	-178.0626	-31.5035	35.3373	344.7500	-7.4482	2.7498	2.2737	378.0000	-5.9598	0.5001	-1.3197	411.7500
34	0	0	0	0	-45.1891	0.0424	52.8506	-14.2590	3.0647	-3.9446	-0.2476	1.0735	3.8401	5.8281	0.1913	0.3782	-4.8659
35	0	0	0	0	-26	6.9763	20.5007	-6.6788	0.2500	-1.6336	-2.7499	-0.6764	0.5000	3.2340	-3.9999	-1.3398	-0.7500
36	0	0	0	0	1.9432	2.2406	-5.6588	1.4569	-3.7060	-1.4269	-1.2507	-0.0555	2.7385	-4.6217	0.4615	0.1718	0.6630
37	28.7500	-37.5704	28.7516	-15.5573	520.2500	36.2179	-44.2481	-19.0632	368.5000	7.6164	5.5000	0.0922	368.0000	6.9960	8.4997	2.5142	393.2500

Figure 17. Coefficient matrix of ROI obtained from 4 4 sub-blocks



Array Editor - H

	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
28	0	0	0	0	0	0	0	0	2.9006	-2.8860	-10.3845	-1.2041	-0.8446	0.2500	0.0793	-0.1893	-0.4387
29	0	0	0	0	28.5000	-34.7496	22.0000	-9.4189	336.7500	-35.4918	-15.2500	-13.1704	387.7500	-5.8235	-6.2500	-1.6468	401.7500
30	0	0	0	0	-29.0094	34.6525	-20.5167	7.8536	-16.1911	-23.7154	-20.6520	-13.2374	-0.9799	0.7803	0.3266	0.3232	-1.6332
31	0	0	0	0	7.0000	-6.6585	0.5000	2.2168	-3.2500	-1.5211	-3.2500	0.9007	0.7500	1.2505	0.7500	-0.2474	-0.2500
32	0	0	0	0	4.4393	-7.1464	7.9571	-5.6525	8.6008	10.7626	8.2837	4.2154	-0.4059	-0.6768	0.1353	-0.2803	-0.6765
33	0	0	0	0	245.0000	-178.0358	-31.5000	35.3199	344.7500	-7.4471	2.7500	2.2729	378.0000	-5.9588	0.5000	-1.3202	411.7500
34	0	0	0	0	-45.1965	0.0429	52.8581	-14.2604	3.0655	-3.9445	-0.2474	1.0732	3.8404	5.8284	0.1913	0.3787	-4.8668
35	0	0	0	0	-26.0000	6.9755	20.5000	-6.6777	0.2500	-1.6332	-2.7500	-0.6765	0.5000	3.2336	-4	-1.3394	-0.7500
36	0	0	0	0	1.9439	2.2396	-5.6587	1.4571	-3.7061	-1.4268	-1.2506	-0.0555	2.7388	-4.6213	0.4619	0.1716	0.6629
37	28.7500	-37.5637	28.7500	-15.5594	520.2500	36.2130	-44.2500	-19.0689	368.5000	7.6152	5.5000	0.0929	368.0000	6.9948	8.5000	2.5146	393.2500

Figure 18. Coefficient matrix of ROI obtained from 2 2 to 4 4 sub-blocks

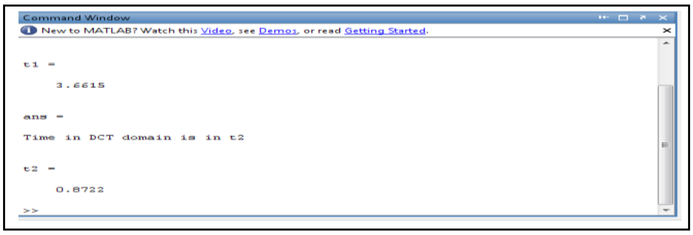


Fig.ure 19. Time requirements of both methods

VI . CONCLUSIONS

In this paper, we have proposed a concept of general spatial relationship between the DCT of a block and that of its sub-blocks and described an analytical expression of this relationship. The results reveal that there exists a concise and linear relationship between the DCT of a block and that of its sub-blocks. This is represented by (12) for 2-D signals. The significance of the work lies in the fact that a substantial savings in computing cost can be achieved in comparison with those in the pixel domain, which is especially useful when image processing is carried out directly in the DCT domain. In comparison with the work recently reported work, the major differences of our work can be highlighted as follows.

• Our approach works out a general spatial relationship directly in the DCT domain, yet in others it is essentially along the existing technique to design the inverse transform, decomposition/composition, and forward transform into a pipelining structure.

• While our approach is designed to optimize the general algorithm complexity and computing cost, and optimizes hardware/VLSI implementation.

• While our approach is characterized by clear and direct relationships between the two sets of DCT coefficients for one block and its decomposed sub-blocks, in other papers it is represented by a pipelining architecture, which can be regarded as a filtering process in the transform domain.

• In our approach, the direct spatial relationship in the DCT domain is characterized by the sparse matrix A, which contributes to significant savings in computing cost in comparison with the work reported in [1]. However, existing methods has a higher computing cost than [1] in terms of software implementations

References

[1] Guocan Feng and Jianmin Jiang, "The spatial relationship of DCT coefficients between a block and its sub-blocks," IEEE Trans. signal processing, vol.50, no.5, may 2002.

[2] Gupta, Maneesha; Garg, Amit Kumar; Kaushik, Abhishek, "Review: Image Compression Algorithm," IJCSI & Technolo; Vol. 1 Issue 10, p649, Nov. 2011.

[3] Athanassios N. Skodras, "Direct Transform to Transform Computation," IEEE Signal Processing, vol. 6, no. 8, August 1999.

[4] W. Kou and T. Fjallbrant, "A direct computation of DCT coefficients for a signal block taken from two adjacent blocks," IEEE *Trans. Signal* Processing, vol. 39, pp. 1692– 1695, July 1991.

[5] E. Feig and S. Winograd, "Fast algorithms for the discrete cosine transform," *IEEE Trans. Signal Processing*, vol. 40, pp. 2174–2193, Sept. 1992.

[6] I. Ng, T. Tan, and J. Kitter, "On local linear transform and Gabor filter representation of texture," in *Proc. 11th Int. Conf. Pattern Recognit.*, 1992, pp. 627–631.

[7] J. R. Smith and S. F. Chang, "Transform feature for texture classification and discrimination in large image database," in *Proc IEEE Int. Conf. Image Process*, vol. 3, 1994, pp. 407– 411.

[8] A. A. Abdel-Malek and J. E. Hershey, "Feature cueing in the discrete cosine domain," *J. Electron. Imag.*, vol. 3, pp. 71-80, Jan. 1994.

[9] B. Shen and I. K. Sethi, "Direct feature extraction from Compressed images," *Proc. SPIE, Storage Retrieval Image Video Databases IV*, vol. 267, 1996.

[10] R. Reeve, K. Kubik, and W. Osberger, "Texture characterization of compressed aerial images using DCT coefficients," *Proc. SPIE, Storage Retrieval Image Video Databases V*, vol. 3022, pp. 398 407, Feb. 1997.

[11] J. R. Hernandez, M. Amado, and F. P. Gonzalez, "DCT domain watermarking techniques for still images: Detector performance analysis and a new structure," *IEEE Trans. Image Processing*, vol. 9, pp. 55–68, Jan. 2000.

[12] M. K. Mandal, F. Idris, and S. Panchanatha, "Image and video indexing in the compressed domain: A critical review," *Image Vis. Comput. J.*, to be published.

[13] Guocan Feng and Jianmin Jiang, "Image spatial transformation in DCT domain," IEEE, 2001.