Extraction of Region of Interest in Compressed Domain

Vaishali Choudhary[1], Preeti Voditel[2]

[1] Ramdeobaba College of Engineering and Management
Nagpur, Maharashtra, India

[2] Ramdeobaba College of Engineering and Management
Nagpur, Maharashtra, 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 namely image compression, image analysis, and image filtering and image enhancement.

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, Discrete Cosine Transform (DCT) compression mechanism is used as excellent energy compaction. The image processing tasks can be reduced by considering the approach described in this paper, which consists of finding relationship between coefficients of a block to all other sub blocks in DCT domain without decompressing it.

This paper proposes an algorithm in DCT domain which composed of a block from all its sub-blocks and sub-blocks from its block and produces similar result for both operations thus lowering the complexities of the algorithm.

Keywords: *compressed domain processing, Discrete Cosine Transform (DCT), spatial relationship, global features.*

1. Introduction

Over the years, image processing has been developed primarily in the pixel domain. The new research area is image processing in compressed domain ,as digital images are stored in the form of Discrete Cosine Transform (DCT) [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. Following are some reasons why we need transform domain operations.

1.1 Mathematical Convenience

Convolution In time domain	⟺	Multiplication In frequency domain

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.

1.2 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 and 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. The main reasons to transform an image are: first, the analysis for the image pattern can be accessible by transforming critical components which may be isolated. Second, the transformed image data is stored in compact form for efficient transmission.

1.3 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.

1.3.1 Properties of DCT

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

1.3.1.1 De-correlation

The removal of redundant neighboring pixels is the principle advantage of image transformation. This results in encoded uncorrelated transform coefficients. Clearly, after DCT operation the autocorrelation amplitude lowers during all lags. Hence, it can be inferred that DCT exhibits excellent de-correlation properties.

1.3.1.2 Energy Compaction

The efficiency of a transformation scheme is to bundle input data into as less coefficients as possible. The reconstructed image coefficients consisting of small amplitudes without visual distortions are discarded by the quantizer.DCT exhibits excellent energy compaction for highly correlated images.

2. 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.

Gupta et al. [2] proposed that the discrete cosine transform (DCT) is a technique for converting a signal into elementary frequency components. It is widely used in image compression. Compression using DCT and varying coefficients for compression were developed to show the resulting image and error image from the original images. Image Compression is studied using 2-D discrete Cosine Transform. The original image is transformed in 8-by-8 blocks and then inverse transformed in 8-by-8 blocks to create the reconstructed image. The inverse DCT performed using the subset of DCT coefficients. The error image (the difference between the original and reconstructed image) would be displayed. Error value for every image would be calculated over various values of DCT co-efficient as selected by the user and would be displayed in the end to detect the accuracy and compression in the resulting image.

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 latter. The proposed method has the lower computational complexity than the traditional algorithm 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. This algorithm which requires inverse transforms of two received coefficients blocks followed by a forward transform reduces multiplications and additions/subtractions.

Ephraim Feig et.al. [5] introduced several fast algorithms on multidimensional inputs of sizes which are powers of 2 for computing discrete cosine transforms (DCT's).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 used DCT coefficients, to detect oriented line features. 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 × 4 blocks. The whole image is used for the computation of the variance and means absolute values of each of these coefficients. The 32-component vector is used for the representation of the feature of the entire image, 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.

3. Need of Compressed Domain Operations

As this area of research is emerging, 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. To ensure optimized performances, a new problem of various DCT block sizes has to be used. This include 8 × 8 blocks used in JPEG, 4 × 4 blocks used in image indexing, and 16 × 16 macro-blocks in MPEG. The existing approach, for the inter-transfer of DCT coefficients from different blocks with various sizes, 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.

4. Possible Applications

4.1 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. It is desirable to have image processing tasks such as image indexing, feature extraction, and pattern classifications implemented directly in the DCT domain, in order to save memory and computational cost.
Extracting global features in compressed domain is useful for general image processing tasks which are widely used in image indexing and pyramid algorithms.
The computational complexity of the proposed algorithm is significantly lower than that of the existing methods, due to the fact that the corresponding coefficient matrix of the linear combination is sparse.

4.2 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. To provide additional information or to point out specifics, it is often used. We can add a caption and identifying figure number to charts, smart art and pictures.
On the internet many videos are uploaded every day. Detecting video copies from a video sample is the basis of "Video Copy Detection". 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.

5. ROI Extraction Steps

5.1 Boundary Detection

It can be done by:

- Impoly
- Imrect
- Roipoly
- Getline functions.

Among these, we have used roipoly and getline functions.

5.1.1 Roipoly

Specify polygonal region of interest (ROI). It is used 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. An interactive polygon tool, associated with the image displayed in the current figure, called the target image, is created by BW = roipoly. When we move the pointer over the image in the figure, the pointer changes to cross hairs provided that the polygon tool is active. We can specify the region by selecting vertices of the polygon, using the mouse. You can move or resize the polygon using the mouse. After sizing and positioning the polygon, the mask is created by double-clicking or by right-clicking inside the region and selecting create mask from the context menu. The function 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.

5.1.2 Getline

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

5.2 ROI Extraction in Spatial Domain

Fig. 1 Input image

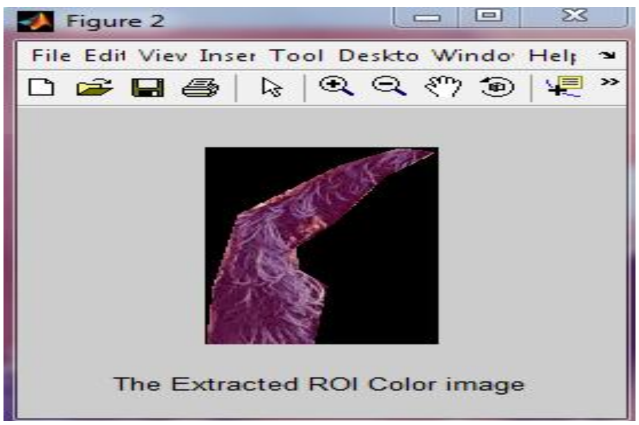

Fig. 2 ROI extracted in spatial domain.

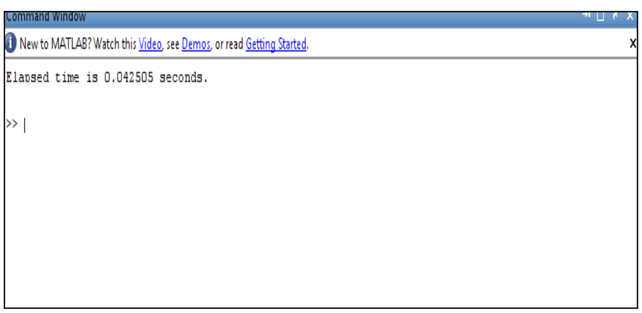

Fig. 3 Time required to extract ROI in spatial domain.

5.3 Mathematical Derivation for Mapping of 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 $L \times N$ rows and $M \times N$ columns, in order to obtain a convenient representation of its sub-blocks. Correspondingly, this block B can be divided into $L \times M$ sub-blocks represented as B_{lm} 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 B_{lm} are represented as C_B and C_{lm}(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 C_{lm}(u, v), i.e., the relationship between the DCT coefficients of the block B and that of its sub-blocks B_{lm} .

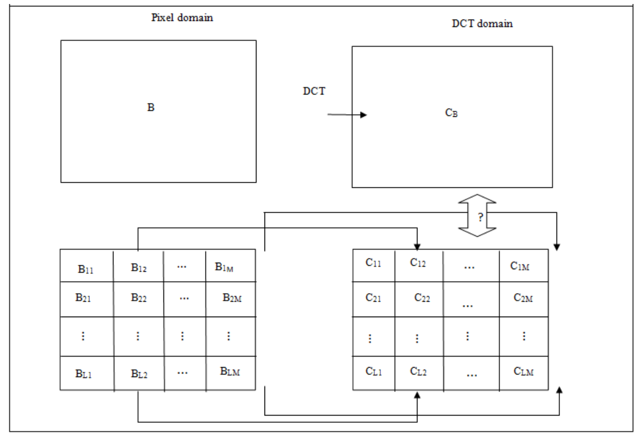

Fig. 4 Schematic illustration of problem to be solved

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,

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{2}}\ for\ u = 0 \\ 1 \quad otherwise \end{cases}$$

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

$\cos\left(\frac{(2i+1)u\pi}{2LN}\right)\cos\left(\frac{(2j+1)v\pi}{2MN}\right)$ as Φ_{uv}(i, j). In fact, it is the product of 2 1-D basis functions

Φ_u^1 (i) =cos $\left(\frac{(2i+1)u\pi}{2LN}\right)$ And Φ_v^2(j)=cos $\left(\frac{(2j+1)v\pi}{2MN}\right)$

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

$$\Psi_{uv}(i,j) = \begin{cases} \cos\left(\frac{2(i mod N)(u mod N)\Pi}{2N}\right)\cos\left(\frac{2(j mod N)(v mod N)\Pi}{2N}\right) & \text{When}[i/N]=[u/N] \text{ and } [j/N]=[v/N] \\ 0 & \text{Otherwise} \end{cases} \quad (3)$$

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

$$\Psi_u^1(i) = \begin{cases} \cos\left(\frac{2(i mod N)(u mod N)\Pi}{2N}\right) & \text{When } [i/N] = [u/N] \\ 0 & \text{Otherwise} \end{cases}$$

(i, u = 0,1 ,…., LN-1) (4)

And

$$\Psi_v^2(j) = \begin{cases} \cos\left(\frac{2(j mod N)(v mod N)\pi}{2N}\right) & \text{When } [j/N] = [v/N] \\ 0 & \text{Otherwise} \end{cases}$$

(j, v= 0,1,…., LM-1) (5)

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

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

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

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

where β_k^u are the parameters with respect to $\Phi_u^1(i)$ and $\Psi_k^1(i)$, which can be uniquely determined by substituting the value of i (i=0,l,. .. , *LN-1)*. Similarly for Φ_v^2 (j),

$$\Phi_v^2(j) = \sum_{l=0}^{MN-1} \gamma_l^v \Psi_l^2(j).\ (v= 0, 1,\dots, MN-1) \quad (8)$$

Denote matrices $\{\beta_k^u\}$ and $\{\gamma_l^v\}$ as $\mathbf{D}_{LN \times LN}$ and $\mathbf{F}_{MN \times MN}$. 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).

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

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

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 C_{ij} represents the set of DCT coefficients for sub-block B_{ij}, 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:

$$C_B = \frac{1}{M}\, D(C_{lm})\, D^T \quad (12)$$

Finally, Since there exists inverse matrix for matrices D or F, the block matrix { C_{ij} }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.

We have applied an example of DCT coefficients exchange given in figure 11 to Region of Interest (ROI) and results obtained are given below:

5.3.1 DCT Coefficients Exchange Applied to ROI

Fig. 5 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. 6 shows the ROI extracted and resized ROI is shown by Fig. 7. G matrix is obtained by applying 2×2 to 4×4 conversion i.e. obtain DCT of ROI using 2×2 sub-blocks, extract 4×4 block from each row and multiply it with the D matrix and its transpose to obtain coefficients, which will be same as that of by 4×4 sub-block. After concatenating blocks first horizontally and then vertically we will get DCT of ROI in form of matrix G and DCT of ROI in form of matrix H is obtained from 4×4 block size. Figures 8 and 9 show the coefficients obtained from the conversion of 2×2 block size to 4×4 and directly from 4×4 blocks respectively, which are approximately same.

$$C_B(u,v) = \sqrt{\frac{4}{LN*MN}}\sum_{i=0}^{LN-1}\sum_{j=0}^{MN-1} x(i,j)\Psi_{uv}^2(i,j)$$

(9)

$$= \sqrt{\frac{4}{LN*MN}}\sum_{i=0}^{LN-1}\sum_{j=0}^{MN-1}\left(x(i,j)(\beta_0^u \quad \beta_1^u \quad \cdots \quad \beta_{LN-1}^u)\begin{pmatrix}\Psi_0^1(i)\\ \Psi_1^1(i)\\ \vdots \\ \Psi_{LN-1}^1(i)\end{pmatrix}(\Psi_0^2(j) \quad \Psi_1^2(j) \quad \cdots \quad \Psi_{MN-1}^2(j))\begin{pmatrix}\gamma_0^v\\ \gamma_1^v\\ \vdots \\ \gamma_{LN-1}^v\end{pmatrix}\right)$$

$$= \sqrt{\frac{4}{LN*MN}}(\beta_0^u \quad \beta_1^u \quad \cdots \quad \beta_{LN-1}^u)\sum_{i=0}^{LN-1}\sum_{j=0}^{MN-1}\left(x(i,j)\begin{pmatrix}\Psi_0^1(i)\Psi_0^2(j) & \Psi_0^1(i)\Psi_1^2(j) & \cdots & \Psi_0^1(i)\Psi_{MN-1}^2(j)\\ \Psi_1^1(i)\Psi_0^2(j) & \Psi_1^1(i)\Psi_1^2(j) & \cdots & \Psi_1^1(i)\Psi_{MN-1}^2(j)\\ \vdots & \vdots & & \vdots \\ \Psi_{LN-1}^1(i)\Psi_0^2(j) & \Psi_{LN-1}^1(i)\Psi_1^2(j) & \cdots & \Psi_{LN-1}^1(i)\Psi_{MN-1}^2(j)\end{pmatrix}\right)\begin{pmatrix}\gamma_0^v\\ \gamma_1^v\\ \vdots \\ \gamma_{LN-1}^v\end{pmatrix}$$

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

$$C_B(u,v) = \sqrt{\frac{1}{L*M}}\begin{pmatrix}\beta_0^u & \beta_1^u & \cdots & \beta_{LN-1}^u\end{pmatrix}\begin{pmatrix} C_{0,0}(0,0) & \cdots & C_{0,0}(0,N-1) & C_{0,M-1}(0,0) & \cdots & C_{0,M-1}(0,N-1) \\ \vdots & \cdots & & \vdots \;\cdots\; \vdots & \cdots & \vdots \\ C_{0,0}(0,0) & \cdots & C_{0,0}(N-1,N-1) & C_{0,M-1}(0,0) & \cdots & C_{0,M-1}(N-1,N-1) \\ C_{L-1,0}(0,0) & \cdots & C_{L-1,0}(0,N-1) & C_{L-1,M-1}(0,0) & \cdots & C_{L-1,M-1}(0,N-1) \\ \vdots & \cdots & \vdots & \vdots & & \vdots \\ C_{L-1,0}(0,0) & \cdots & C_{L-1,0}(N-1,N-1) & C_{L-1,M-1}(0,0) & \cdots & C_{L-1,M-1}(N-1,N-1) \end{pmatrix}\begin{pmatrix}\gamma_0^v \\ \gamma_1^v \\ \vdots \\ \gamma_{LN-1}^v\end{pmatrix} \qquad (10)$$

Fig. 5 Lena image

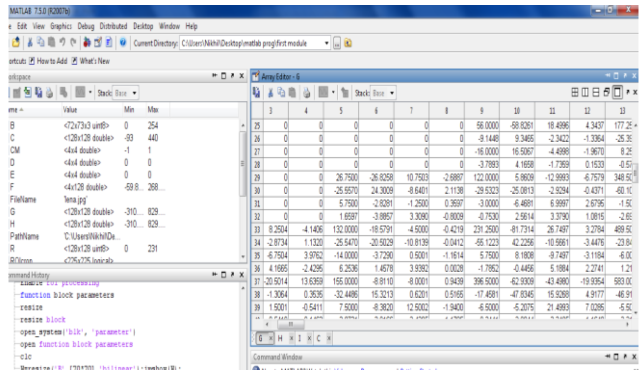

Fig. 8 G matrix obtained from 2×2 block to 4×4 block size

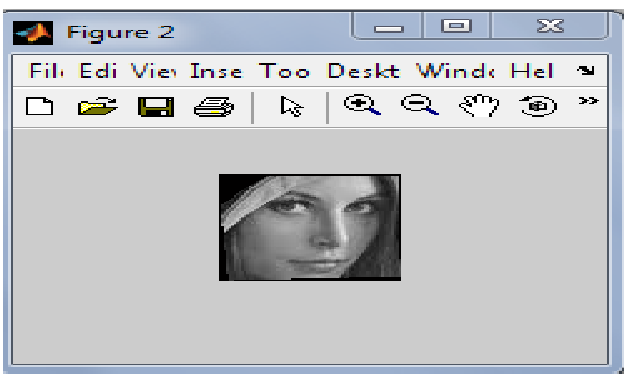

Fig. 6 ROI extracted from fig. 5

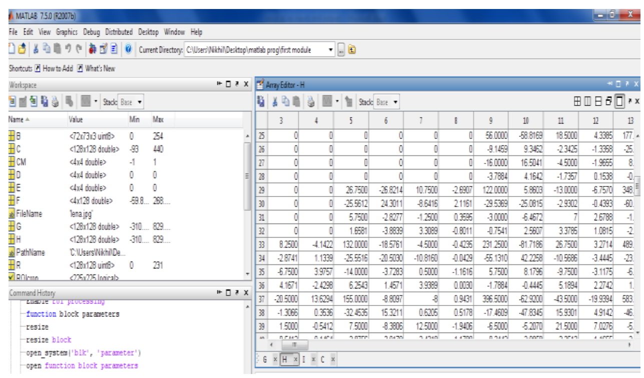

Fig. 9 H matrix obtained from 4×4 block size

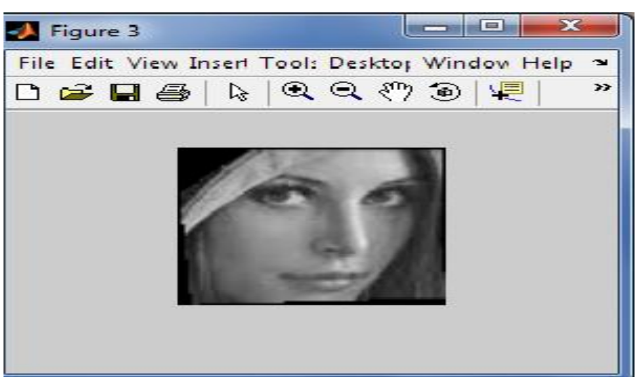

Fig. 7 ROI resized

5.4 Our Application of Block to Sub-block Theory

Figure 10 shows how the relationship of DCT coefficients between a block and its sub-blocks is used. 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 is possible if we use getline function in addition to roipoly function. Among impoly, imrect and roipoly functions, roipoly is used because with help of this function we can choose any region of our interest. For example, if we want to select hair in “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 are using roipoly function to detect the boundary. Consider a situation as shown in figure 10 where an image is already divided into number of blocks having dimension 8×8 and then we have selected a region of an image as shown . Now, according to the figure, the problem is that some of the blocks are not completely inside the boundary of selected region so what we will do according to existing methods (spatial domain), we will apply IDCT to that particular block for obtaining coefficients in pixel domain and will again divide that block into 4× 4 block (if we want) to apply the DCT again and produce the DCT coefficients of 4×4 block each. 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 will be as follows:

Instead of going back to pixel domain with the help of IDCT, we can obtain coefficients of any block sizes in DCT domain only. This will be possible with the help of (12) and by applying the DCT coefficients exchange example as shown in figure 11. After extracting ROI in spatial domain and in DCT domain we will compare the time taken to extract those in spatial and in DCT domain respectively. In this way, we may expect lower computational complexity, lower computing cost and also expecting that time required in DCT domain will be less than that of those in pixel domain.

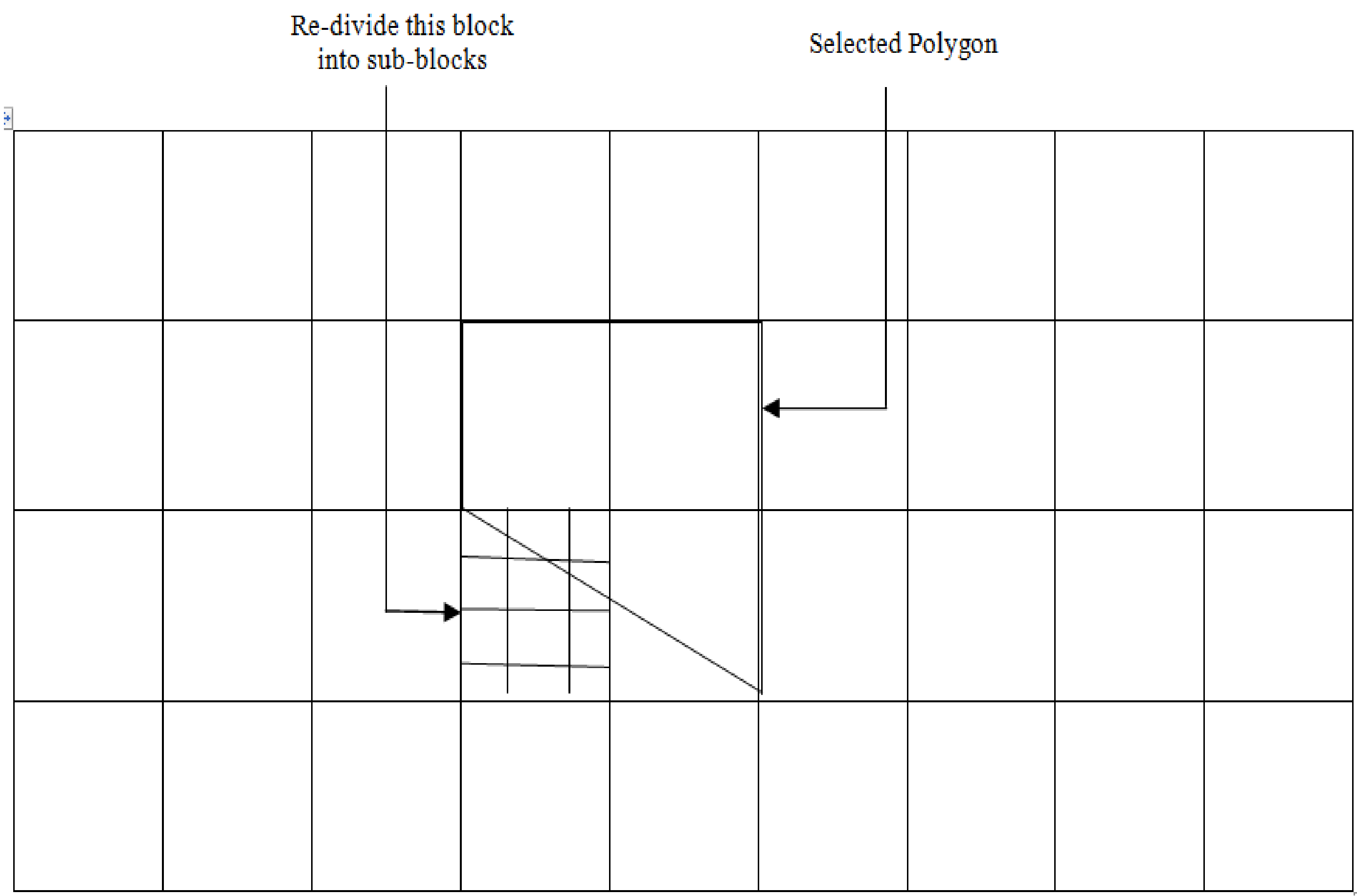

Fig. 10 Application of block to sub-block theory

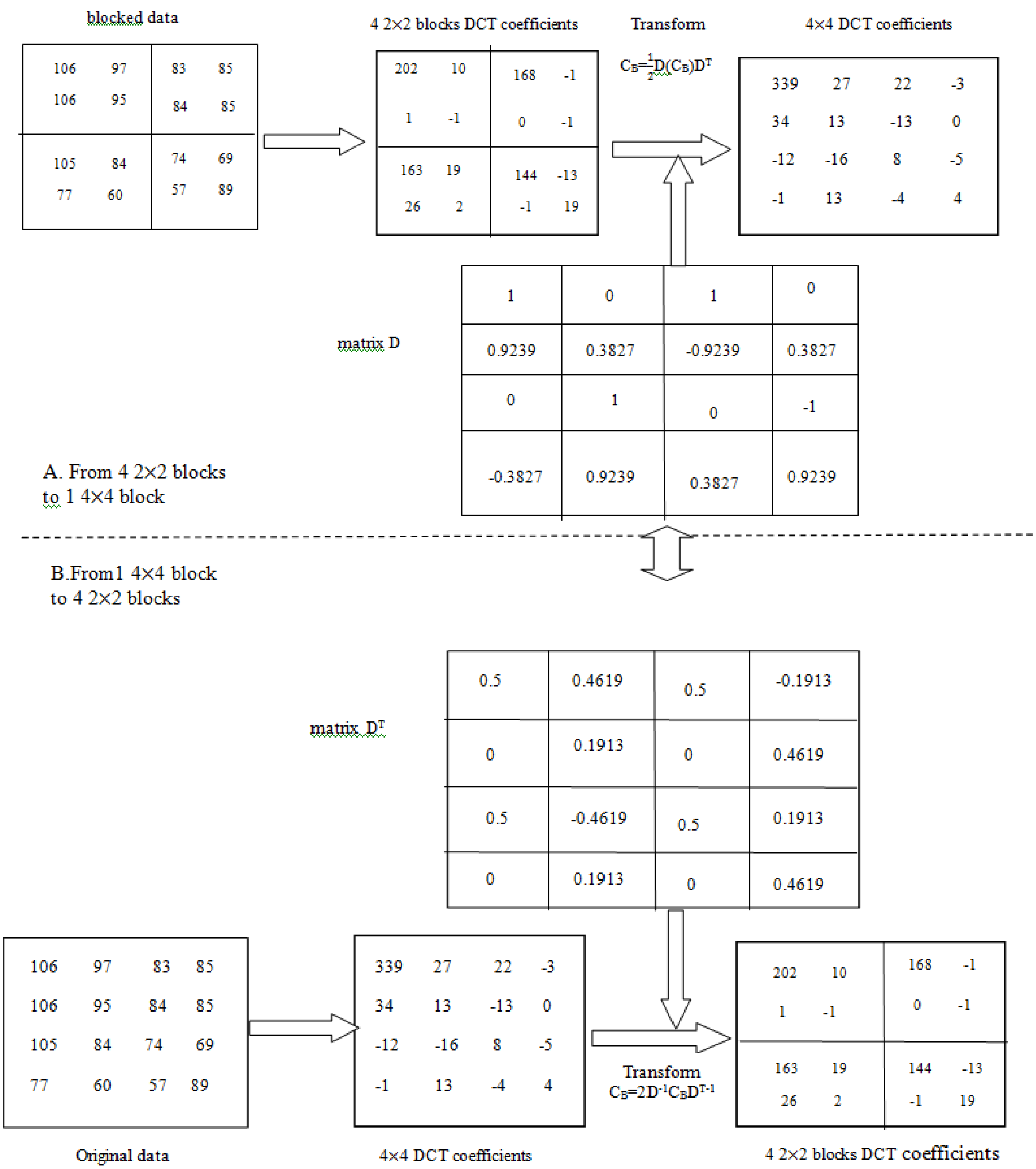

Fig. 11 The example for DCT coefficients exchange between 4 2×2 blocks and 1 4×4 block

6. 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.

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

• It 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.