AUTHOR GUIDELINES FOR ICIP-2004 PROCEEDINGS AUTHORS

Author(s) Name(s)

Author Affiliation(s)

ABSTRACT

Since the widespread use of the internet and internet-related applications storing, transmission and also processing of the images in the compressed domain remains one of the widely researched fields.The DCT domain for image compression is commonly used to save computational complexity and huge memory requirements. In this paper we have used quadtree decomposition and the spatial relationship of DCT coefficients between a block and its sub-blocks to achieve extraction, logo insertion for grayscale and color images. We have also shown that the proposed method saves significantly on computation and memory.

Discrete Cosine transform(DCT),Quadtree Decomposition, JPEG image compression standard, extraction of Region of Interest, Logo Insertion.

1. INTRODUCTION

Normally compressed images are decompressed for processing in the spatial domain and recompressed. The time, space and computational complexity of the above method is significant.

At present all digital images are stored and transferred in their compressed format in which discrete cosine transform (DCT) based compression remains one of the most important compression techniques due to the efforts from JPEG. In order to save the memory and computation cost, it is desirable to have image processing operations such as feature extraction, image indexing and pattern classification implemented directly in the DCT domain. In the compressed or DCT domain processing for optimised performance various DCT block sizes have to be used. These include 8 x 8 blocks used in JPEG, 4 x 4 blocks used in image indexing [**?**] and 16 x 16 macroblocks in MPEG. The conventional approach would involve decompression via IDCT to the pixel domain, redivision into the required block size and recompression. To overcome this inefficient approach the spatial relationship of DCT coefficients between a block and its sub-blocks has been analysed efficiently in [**?**].This relationship has been used in our paper. Work in this field dealing with the relationship of a block with its two adjacent sub-blocks is dealt in [**?**], [**?**]. Here we have used this spatial relationship of the DCT coefficients to decompose and recompose blocks of varying sizes. We have used a multilevel decomposition scheme for capturing internal structure of a block relating to a homogeneous property defined in the spatial domain. We have used quadtree decomposition for this purpose. Generally quadtree decomposition is an analysis technique that involves subdividing the image into blocks that are more homogeneous than the image itself. It thus obtains the image at various resolution levels. This results in some kind of image segmentation used for analysis and compression.

Thanks to XYZ agency for funding.

2. MULTILEVEL DECOMPOSITION AND RECOMPOSITION OF DCT BLOCKS

We discuss here briefly the decomposition of DCT blocks into smaller sub-blocks and their recomposition into larger 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 LxN rows and MxN columns in order to obtain a convenient representation of its sub-blocks. Correspondingly, this block can be divided into LxM sub-blocks, each of size NxN. For 1-D signals, the block B becomes a 1-D vector, which can be regarded as having MxN elements,$B = x_i, i = 0, 1, ..M \times N - 1$. this vector can be divided into M consecutive sections $T_p p = 0, 1...M - 1$, each having N elements. Let there be M blocks of$N$-point DCT of a sequence $\{x(n)\}, n = 0, 1, ..., M \times N - 1$.

Hence the DCT of the pth block can be expressed as follows:

$$C_p(k) = \sqrt{\tfrac{2}{N}}\alpha(k)\sum_{n=0}^{N-1} x(pN+n)cos(\tfrac{(2n+1)\pi k}{2N}), \quad 0 \le p \le M-1, 0 \le k \le N-1. \tag{1}$$

On the other hand, the $M \times N$ DCT of $x(n)$ is expressed as:

$$C(k) = \sqrt{\tfrac{2}{M\times N}}\alpha(k)\sum_{n=0}^{M\times N-1} x(n)cos(\tfrac{(2n+1)\pi k}{2\times M\times N}), \quad 0 \le k \le M \times N - 1 \tag{2}$$

In both of the above two equatons $\alpha(k)$ is given by Eq. (**??**).

Jiang and Feng showed that a block DCT transformation as expressed by Eq. (1) is nothing but the orthonormal expansion of the sequence $\{x(n)\}$ with a set of $M \times N$ basis vectors, each of which is derived from the basis vectors of N-point DCT. Hence there exists an invertible linear transformation from M blocks of N-point DCT transform to the usual MN-point DCT transform. In other words, for a sequence of N-point DCT blocks $\{C_i^{(N)}\}, i = 0, 1, ...M - 1$, the corresponding composite DCT $C^{(M.N)}$ ($M.N$-point DCT), there exists a matrix $A_{(M,N)}$ of size $MN \times MN$ such that

$$C^{(M.N)} = A_{(M,N)}.[C_0^{(N)} C_1^{(N)} \dots C_{M-1}^{(N)}]^T, \quad (3)$$

where the super-script ^{T} denotes the transpose of a matrix.

The analysis in the 1-dimension can be easily extended to 2-dimensions. In this case, for an $L \times M$ adjacent DCT blocks, a composite DCT block can be formed by the following equation:

$$C^{(LN \times MN)} = A_{(L,N)} \cdot \begin{bmatrix} C_{0,0}^{(N\times N)} & C_{0,1}^{(N\times N)} & \dots & C_{0,M-1}^{(N\times N)} \\ C_{1,0}^{(N\times N)} & C_{1,1}^{(N\times N)} & \dots & C_{1,M-1}^{(N\times N)} \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ C_{L-1,0}^{(N\times N)} & C_{L-1,1}^{(N\times N)} & \dots & C_{L-1,M-1}^{(N\times N)} \end{bmatrix} \cdot A_{(M,N)}^T \quad (4)$$

Similarly, for decomposing a DCT block $C^{(L.N \times M.N)}$ to $L \times M$ DCT blocks of size $N \times N$ each, following expression is used:

$$\begin{bmatrix} C_{0,0}^{(N\times N)} & C_{0,1}^{(N\times N)} & \dots & C_{0,M-1}^{(N\times N)} \\ C_{1,0}^{(N\times N)} & C_{1,1}^{(N\times N)} & \dots & C_{1,M-1}^{(N\times N)} \\ . & . & \dots & . \\ . & . & \dots & . \\ . & . & \dots & . \\ C_{L-1,0}^{(N\times N)} & C_{L-1,1}^{(N\times N)} & \dots & C_{L-1,M-1}^{(N\times N)} \end{bmatrix} = A_{(L,N)}^{-1} \cdot C^{(L.N \times M.N)} \cdot A_{(M,N)}^{-1\,T} \quad (5)$$

Typical examples of the conversion matrices are given in Eqs. (8) and (9). The matrix $A_{(2,4)}$ is used to decompose a DCT block of 8×8 to four blocks of 4×4 each. Similarly it can be used to recompose four 4×4 DCT blocks to a single block of 8×8.

$$\begin{bmatrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{bmatrix} = inv(A_{(2,4)}) \times C_{(8\times8)} \times inv(A_{(2,4)})^T \quad (6)$$

$$[C_{8\times8}] = A_{(2,4)} \times \begin{matrix} C_{11} & C_{12} \\ C_{21} & C_{22} \end{matrix} \times A_{(2,4)}^T \quad (7)$$

It may be noted that the conversion matrices and their inverses are sparse. Hence, lesser number of multiplications and additions of two numbers are needed than those required in the usual matrix multiplications.

3. EXTRACTION OF REGION OF INTEREST

JPEG transmission standard is in the form of 8×8 blocks of DCT. For extraction of the region of interest in the pixel domain inverse discrete cosine transform of the blocks covering the area of interest and also other partially overlapping blocks will have to be performed. If extraction of the region of interest has to be performed in the DCT domain itself we will save the above effort of the IDCT calculation. In this case, we find out the DCT blocks which lie in the region of interest and also those which partially lie within the region of interest. Each DCT block os checked for homogeneity i.e only those blocks whic partially lie in the region of interest are further decomposed applying multilevel decomposition to block sizes of 4×4 and if further needed to sizes of 2×2. It has been observed that the required accuracy is obtained till block sizes of 2×2. Thus for blocks on the edges of the region of interest, not contained wholly in the region of interest, multilevel decomposition similar to quadtree decomposition is applied. This multilevel decomposition is as per (6) Each decomposed 8×8 block is recomposed to 8×8 block following the same quadtree partitioning in the reverse order starting in the bottom up order.

3.1. Algorithm *Extraction*

Input: Binary image of region of interest, 8×8 DCT of original image
Output:The extracted image.
Begin

(a) If an 8×8 block is totally contained in ROI output it as it is.

(b) If an 8×8 block is totally outside the ROI, set DCT values to zero.

(c) For a mixed 8×8 block, perform quadtree decomposition. Retain the coefficients if the leaf node is totally contained in ROI else discard it.

(d) Recompose the decomposed nodes to 8×8 blocks following the same quadtree partitioning.

End *Extraction*

3.2. Analysis of the Extraction Algorithm

3.2.1. *Worst case Analysis*

The worst case for one block is when the DCT for its leaves of all possible granularities need to be found. Remember

$$A_{(2,4)} = \begin{bmatrix} 0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 \\ 0.6407 & 0.294 & -0.0528 & 0.0162 & -0.6407 & 0.294 & 0.0528 & 0.0162 \\ 0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 & 0 \\ -0.225 & 0.5594 & 0.3629 & -0.0690 & 0.225 & 0.5594 & -0.3629 & -0.069 \\ 0 & 0 & 0.7071 & 0 & 0 & 0 & 0.7071 & 0 \\ 0.1503 & -0.2492 & 0.5432 & 0.3468 & -0.1503 & -0.2492 & -0.5432 & 0.3468 \\ 0 & 0 & 0 & 0.7071 & 0 & 0 & 0 & -0.7071 \\ -0.1274 & 0.1964 & -0.2654 & 0.6122 & 0.1274 & 0.1964 & 0.2654 & 0.6122 \end{bmatrix} \quad (8)$$

$$A_{(2,2)} = \begin{bmatrix} 0.7071 & 0 & 0.7071 & 0 \\ 0.6533 & 0.2706 & -0.6533 & 0.2706 \\ 0 & 0.7071 & 0 & -0.7071 \\ -0.2706 & 0.6533 & 0.2706 & 0.6533 \end{bmatrix} \quad (9)$$

that as per Jiang and Feng, the number of addition operations required for decomposing the DCT of an Nby N block to get the DCTs for N/2 by N/2 sub-blocks is $2N + N^2$ and the number of multiplication operations is $\frac{2N+N^2}{2}$.

The overall complexity thus becomes $O(JF(8/4)+JF(4/2)+JF(2/1))$. $JF(2n/n)$ means the time complexity for decomposition of the DCT of an $2n\times 2n$ block to get the DCTs of its 4 $n \times n$ sub-blocks. This amounts to $(80 + 24 + 8 = 112)$ additions and $(40 + 12 + 4 = 56)$ multiplications per $8x8$ block that intersects the region of interest.

Note that the corresponding number of 8x8, 4x4 and 2x2 blocks don't actually matter because Jiang Feng's algorithm gives the DCTs for all the kxk sub-blocks of an image in one shot.

The next step is to find out for the worst case, the number of such $8x8$ blocks on which these worst-case computations must be made. This can happen if each $8x8$ block on the boundary of the rectangle encapsulating the ROI has to undergo the above-mentioned number of operations. The number of blocks on the boundary is $\lceil \frac{perimeter}{8} \rceil = \lceil \frac{2m+2n}{8} \rceil = 2 \times \lceil \frac{n}{8} \rceil + 2 \times \lceil \frac{m}{8} \rceil$.

Let a rectangular block of size mxn just encapsulate the ROI. This means that the bottom-most, top-most, left-most and right-most pixel of the image are (x, km), $(x, km+m)$, (y, kn) and $(y, kn + n)$ respectively.

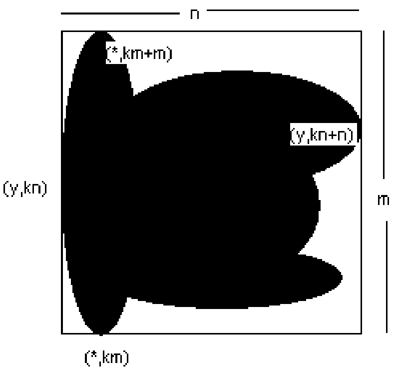

Fig. 1. Worst Case Analysis

Thus in the worst case, the time required for the extraction algorithm to extract the region of interest would be :

$$[2\times\lceil \tfrac{n}{8} \rceil+2\times\lceil \tfrac{m}{8} \rceil]\times[JF(4\times 4)+JF(2\times 2)+JF(1\times 1)]$$

3.2.2. *Best case Analysis*

Best case for the analysis is when the ROI is such that it is exactly a rectangle of size $n \times m$ where n and m are both multiples of 8. Hence - no quadtree decomposition needs to be performed on the region of interest. The corresponding time required will be the constant required to simply output the DCTs of the encapsulated blocks as it is. Thus, time required will be $O(1)$.

4. LOGO INSERTION

In logo insertion the point of placement of the logo is marked and is considered the midpoint of the logo. Logo insertion in the pixel domain will involve computing the inverse DCT of the blocks in the region of placement of the logo. IDCT of blocks completely covered by the logo and those partially covered will have to be computed. For logo insertion in the compressed domain ....(to discuss)

Algorithm *LogoInsertion* Input: Binary image of Logo after placement on image
Output:The original image with the Logo inserted.
Begin

(a) Form an image (called as logo image) of same size with logo placed in the desired location. the rest of the pixels are set to zero.

(b) Partition the logo image in 8×8 blocks.

(c) If all the pixels of a block belong to the logo, perform 8×8 DCT of that block and replace the corresponding DCT block.

(d) If a ZERO block of the logo image is totally contained in the DCT block of the given image, output the DCT block of the given image.

(e) For a mixed block, perform multi-level decomposition following a quad-tree. Retain the DCT coefficients from either the given image or from the logo image according to the homogeneous property of its leaf-node.

(f) Recompose the multilevel-decomposed DCT blocks to 8×8 dct blocks.

End *LogoInsertion*

5. ADDING TRANSPARENCY TO THE LOGO

The above method of logo insertion replaces the blocks of the original image directly by the blocks of the logo. But transparency can be introduced in the logo by introducing a factor α depending upon which some coefficients of the original image and some from the logo will be taken. This factor α varies between 0 and 1.

6. ANALYSIS

For composition from 4 blocks of 4×4 to a single block of 8×8 via the inverse DCT in the pixel domain would require 128 multiplications and 120 additions for 1-D and 256 multiplications and 240 additions for 2-D.

For decomposition for a block of 8×8 to 4 blocks of 4×4 via the inverse DCT in the pixel domain would require 128 multiplications and 96 additions.

For recomposing 4 $N \times N$ DCT blocks into $2N \times 2N$ DCT blocks one requires $2.N^2$ multiplications and $4.N.(N-1)$ additions. For recomposing 4 $N \times N$ DCT blocks into $2N \times 2N$ DCT blocks one requires $2.N^2$ multiplications and $4.N.(N-1)$ additions. For decomposing a $N \times N$ DCT block into 4 $N/2 \times N/2$ DCT block one requires $(2.N + N^2)/2$ multiplications and $2.N + N^2$ additions.

7. EXTENSION TO THE PRESENT WORK

The present work of extraction of region of interest, logo insertion and adding transparency to the logo have been extended to color images.

OTher possible applications of the present theory and work can be in the area of text insertion in the DCT domain, all these applications for MPEG-2 video and various pixel domain applications such as cropping, enhancement, resizing, filtering, edge detection, dilation, erosion etc. to be converted to the DCT domain.