Fast 2-D 8×8 discrete cosine transform algorithm for image coding

Article *in* Science in China Series F Information Sciences · February 2009

DOI: 10.1007/s11432-009-0038-4 · Source: DBLP

Xh Ji
Shandong University of Finance and Economics
13 PUBLICATIONS **50** CITATIONS

Caiming Zhang
Shandong University
212 PUBLICATIONS **1,073** CITATIONS

Jiaye Wang
Shandong University
28 PUBLICATIONS **290** CITATIONS

Science in China Series E: Technological Sciences

© 2007 SCIENCE IN CHINA PRESS

Springer

submit stencil

A Fast 2D 8*8 Discrete Cosine Transform Algorithm for Image Coding

Ji Xiuhua[1,2†] Zhang Caiming[1,2] Wang Jiaye[1] Wang Kai[3]

[1] School of Computer Science and Technology, Shandong Economics university, Jinan 250014,China;
[2] School of Computer Science and Technology, Shandong University, Jinan 250061,China
[2] Department of Computer Science, University of South Dakota, Vermillion, SD, 57069, U.S.A.

A new fast two-dimension 8×8 Discrete Cosine Transform (2D 8×8 DCT) algorithm based on the characteristics of the basic images of 2D DCT is presented in this paper. The new algorithm computes every DCT coefficient in turn more independently. Hence, the new fast algorithm is suitable for 2D DCT pruning algorithm of pruning away any number of high-frequency components of 2D DCT. The proposed pruning algorithm is more efficient than the existing pruning 2D DCT algorithms in terms of the number of arithmetic operations, especially the number of multiplications required in the computation.

Discrete Cosine Transform; Basic image; Quantization; Image coding;

1 Introduction

Discrete cosine transform (DCT) has been adopted as an essential part of many well-known image/video compression standards, such as JPEG, MPEG, ITU's H.264 and so on, due to its high energy compaction capability [1, 2]. It is important to implement DCT or inverse DCT (IDCT) efficiently in the real-time image transmission because of the large number of arithmetic operations. Especially, the research of 2-D 8×8 DCT fast algorithms is very significant as they are often used in image compression. There have been tremendous DCT fast algorithms in the literatures [3 to 22]. The most influential algorithms nowadays are listed in Table.1. The literatures [4 to 6] proposed 1-D DCT algorithms, which is applied to 2D 8×8 DCT in each of rows and columns in turn. Other algorithms in Ta-

Table.1 Numbers of arithmetic operations of the famous algorithms for 2D 8×8 DCT

algorithm	multiplication	addition	sum
Chen [6]	256(208)	416(464)	672(672)
Lee [5]	192	464	656
Loeffler [4]	176	464	640
S. C. Chan[19]	144	464	618
N. I. Cho[18]	96	466	562
Feig [3]	94	454	548
Feig Scaled DCT [3]	54	462	516

[†] Corresponding author (email:jixiuhua @sdie.edu.cn)

This work is supported by the National Key Basic Research 973 program of China (2006CB303102), the National Nature Science Foundation of China (No. 60673003, No.60533060).

ble.1 are for 2-D methods. The algorithm introduced by Feig and Winograd [3] is based on the sparse factorizations of the DCT matrix, which requires the least number of arithmetic operations. Generally speaking, the existing fast DCT algorithms focus mainly on reducing the number of operations for computing the whole 2D 8×8 DCT coefficients, but the side effect is that it is inconvenient to compute each DCT coefficient independently. Since DCT possesses a high energy compaction property, it becomes the most appropriate approach for lossy data compression. In many DCT-based applications, the most useful information is contained by the low-frequency DCT coefficients. Exploiting this characteristic, further speed-up is possible. One method is usually called ‘pruning’, where only a subset of the DCT coefficients are computed, and the subset contains only some low-frequency DCT components. Especially, in low bit-rate image coding, only a small proportion of the DCT coefficients located at the top-left corner are computed. There are several algorithms for pruning the 1-D DCT[23 to 25] and the pruning of the 2-D DCT is addressed in [26 to 28]. The pruning algorithm in [26] uses Chen’s fast DCT algorithm and it computes the 2-D DCT by using a pruned 1-D DCT. In [27], two pruning algorithms for Vector-Radix Fast Cosine Transform are presented. Both are based on an in-place approach of the direct two-dimensional fast cosine transform (2D FCT). The first pruning algorithm concerns the computation of $N_0 \times N_0$ out of N×N DCT points, where both N_0 and N are powers of 2. The second pruning algorithm is more general and concerns a recursive approach for the computation of any number of points of arbitrary shaped regions. The algorithm in [28] is based on the Feig and Winograd algorithm, but it is not a traditional pruning DCT algorithm because it ignores the permuting role of the matrix $\boldsymbol{P}_8$ (see Appendix C). In fact, Feig and Winograd exploited the inherent structures in the DCT deeply to reduce the computation cost of computing the whole 2D 8×8 DCT. However, it is inconvenient to compute each DCT coefficient independently with Feig’s algorithm. As a result, it is complicated and inefficient to get pruned DCT with Feig’s algorithm. The literatures [29] presents a fast 2D 8×8 IDCT algorithm for image decompression and the literature [30] introduces a fast 2D 8×8 DCT algorithm for low bit-rate image compression. The validity of the methods of [29] and [30] in image compression process is based on the sparseness of the DCT coefficients of actual image data. The disadvantage of [30] is that its algorithm needs more arithmetic operations to compute the full 2D DCT than the algorithms shown in Table 1.

In this paper, we propose a new fast DCT algorithm which computes each DCT coefficient one by one, which makes the algorithm suitable for parallel computing architectures. The computation cost of its scaled version is 126 multiplications and 452 additions. Based on the algorithm, a new pruning algorithm is introduced for the 2-D 8x8 DCT that can compute any number of low-frequency DCT components conveniently. Its speed of pruning 2-D 8x8 DCT is higher comparing with the existing pruning algorithms.

The paper is arranged as follows: in Section 2, the formula expressing DCT in basic images is introduced; in Sections 3 and 4 the new fast algorithm and its scaled version are described. The pruning algorithm is addressed in Section 5. We give some a simple discussion of the parallel efficiency of the algorithm in Section 6. The summary is given in Section 7.

2 Expressing DCT in Basic Image

The new DCT fast algorithm presented in this paper is based on the concept of basic images, which will be described in this section.

The 2D 8×8 DCT / IDCT can be expressed individually as:

$$F(u,v)=\frac{1}{4}C(u)C(v)\sum_{m=0}^{7}\sum_{n=0}^{7}(f(m,n)\cdot\cos\frac{(2m+1)\pi u}{16}\cdot\cos\frac{(2n+1)\pi v}{16})$$
$$f(m,n)=\frac{1}{4}\sum_{u=0}^{7}\sum_{v=0}^{7}(C(u)\cdot C(v)\cdot F(u,v)\cdot\cos\frac{(2m+1)\pi u}{16}\cdot\cos\frac{(2n+1)\pi v}{16}) \qquad (1)$$

where $m,n,u,v=0,1,...,7$; $f(m,n)$ is a pixel value of a source image $\boldsymbol{f}$ and $F(u,v)$ is a DCT coefficient;

$$C(s)=\begin{cases}1/\sqrt{2} & s=0\\ 1 & \text{others}\end{cases}$$

Formula (1) can also be expressed in a matrix form as:

$$\boldsymbol{F}=\boldsymbol{G}\cdot\boldsymbol{f}\cdot\boldsymbol{G}^T$$
$$\boldsymbol{f}=\boldsymbol{G}^T\cdot\boldsymbol{F}\cdot\boldsymbol{G}$$

where

$$\boldsymbol{F}=\begin{bmatrix}F(0,0) & F(0,1) & \cdots & F(0,7)\\ F(1,0) & F(1,1) & \cdots & F(1,7)\\ \vdots & \vdots & & \vdots\\ F(7,0) & F(7,1) & \cdots & F(7,7)\end{bmatrix}$$

$$\boldsymbol{f}=\begin{bmatrix}f(0,0) & f(0,1) & \cdots & f(0,7)\\ f(1,0) & f(1,1) & \cdots & f(1,7)\\ \vdots & \vdots & & \vdots\\ f(7,0) & f(7,1) & \cdots & f(7,7)\end{bmatrix}$$

$$\boldsymbol{G}=\frac{1}{2}\begin{bmatrix}\frac{1}{\sqrt{2}} & \frac{1}{\sqrt{2}} & \cdots & \frac{1}{\sqrt{2}}\\ \cos\frac{\pi}{16} & \cos\frac{3\pi}{16} & \cdots & \cos\frac{15\pi}{16}\\ \vdots & \vdots & & \vdots\\ \cos\frac{7\pi}{16} & \cos\frac{21\pi}{16} & \cdots & \cos\frac{105\pi}{16}\end{bmatrix}$$

The element of $\boldsymbol{G}$ can be expressed as $g(u,m)$ which is defined by

$$g(u,m)=\frac{1}{2}C(u)\cdot\cos\frac{(2m+1)u\cdot\pi}{2^4} \qquad (2)$$

Let $\boldsymbol{G}_u$ be the vector which locates in the uth row of $\boldsymbol{G}$, then

$$\boldsymbol{f}=\begin{bmatrix}\boldsymbol{G}_0{}^T & \boldsymbol{G}_1{}^T & \cdots & \boldsymbol{G}_7{}^T\end{bmatrix}\cdot\boldsymbol{F}\cdot\begin{bmatrix}\boldsymbol{G}_0\\ \boldsymbol{G}_1\\ \vdots\\ \boldsymbol{G}_7\end{bmatrix}$$

and

$$\boldsymbol{f}=\sum_{u=0}^{7}\sum_{v=0}^{7}(F(u,v)\times\boldsymbol{G}_u{}^T\times\boldsymbol{G}_v)=\sum_{u=0}^{7}\sum_{v=0}^{7}(F(u,v)\times\boldsymbol{T}_{uv}) \qquad (3)$$

where

$$\boldsymbol{T}_{uv}=\boldsymbol{G}_u{}^T\times\boldsymbol{G}_v\,.$$

$\boldsymbol{T}_{uv}$ is defined as the basic image of the DCT coefficient $F(u,v)$ ($u,v=0,1,...,7$). Formula (3)

indicates that the source image $\boldsymbol{f}$ is a linear combination of all the basic images with the DCT coefficients $F(u,v)$ ($u,v=0,1,...,7$) as weights.
For any two matrixes $\boldsymbol{A}$ and $\boldsymbol{B}$ with the same numbers of row n and column m,

$$\boldsymbol{A}=\begin{bmatrix} a_{0,0} & a_{0,1} & \cdots a_{0,m} \\ a_{1,0} & a_{1,1} & \cdots a_{1,m} \\ \vdots & \vdots & \ddots \\ a_{n,0} & a_{n,1} & \cdots a_{n,m} \end{bmatrix} \text{ and } \boldsymbol{B}=\begin{bmatrix} b_{0,0} & b_{0,1} & \cdots b_{0,m} \\ b_{1,0} & b_{1,1} & \cdots b_{1,m} \\ \vdots & \vdots & \ddots \\ b_{n,0} & b_{n,1} & \cdots b_{n,m} \end{bmatrix}$$

operator point product “⊙“ of $\boldsymbol{A}$ and $\boldsymbol{B}$ is defined as

$$\boldsymbol{A} \odot \boldsymbol{B} = \sum_{i=0}^{n}\sum_{j=0}^{m} a_{i,j}b_{i,j}$$

Let $t_{uv}(m,n)$ be the elements of a 8×8 matrix $\boldsymbol{T}_{uv}$. 2D 8×8 DCT can be expressed as:

$$F(u,v)=\sum_{m=0}^{7}\sum_{n=0}^{7}(f(m,n)\cdot t_{uv}(m,n)) = \boldsymbol{f} \odot \boldsymbol{T}_{uv} \qquad （4）$$

3 The New DCT Fast Algorithm

From Formula (4) the key point of improving the efficiency of DCT is to reduce the computing time of $\boldsymbol{f} \odot \boldsymbol{T}_{uv}$. There is a useful property of vectors $\boldsymbol{G}_u$ ($u=0,1,2,...,7$) is that the elements of $\boldsymbol{G}_u$ can be divided into several sets and the absolute values of the elements in each set are the same. According to the property of $\boldsymbol{G}_u$, $\boldsymbol{G}_u$ ($u=0,1,2,...,7$) can be classified into four groups. The relevant property also exists in the basic images $\boldsymbol{T}_{uv}$ ($u,v=0,1,2,...,7$), which can be divided into 16 groups. The basic thought of the new algorithm is that when we compute $F(u,v)$ by using Formula (4), the terms with the same absolute value coefficient can be combined and the operations of each combined term in Formula (4) can be reduced to one multiplication and several additions. As such, the arithmetic operations of 2D DCT are reduced.

3.1 The Property of the Basic Images

Computation of Formula (2) shows that

$$\boldsymbol{G}_0=[g(0,0), g(0,0), g(0,0), g(0,0), g(0,0), g(0,0), g(0,0), g(0,0)] \qquad (5)$$

$$\boldsymbol{G}_4=[g(4,0), -g(4,0), -g(4,0), g(4,0), g(4,0), -g(4,0), -g(4,0), g(4,0)] \qquad (6)$$

Similarly, $\boldsymbol{G}_2$ and $\boldsymbol{G}_6$ can be expressed by the following general form,

$$\boldsymbol{G}_u=[g(u,0), g(u,1), -g(u,1), -g(u,0), -g(u,0), -g(u,1), g(u,1), g(u,0)] \quad u\text{=2,6} \qquad (7)$$

and $\boldsymbol{G}_1, \boldsymbol{G}_3, \boldsymbol{G}_5$ and $\boldsymbol{G}_7$ can also be expressed in a general form.

$$\boldsymbol{G}_u=[g(u,0), g(u,1), g(u,2), g(u,3), -g(u,3), -g(u,2), -g(u,1), -g(u,0)] \text{，} u\text{=1,3,5,7} \quad (8)$$

We define h_u as the classifying parameter in order to discuss the following parts conveniently. The variable u can be resolved as follows:

$$u = 2^{h_u} \times L_u$$

where h_u is zero or a positive integer and L_u is zero or a positive odd integer. If $u=1,3,5,7$, $h_u=0$ and L_u = 1,3,5,7 ; if $u=2,6$, $h_u=1$ and L_u = 1,3 ; if $u=4$, $h_u=2$ and L_u = 1 ; if $u=0$, L_u = 0 and $h_u=\lambda$ (λ is an arbitrary value). According to the discussion above, the vectors $\boldsymbol{G}_u$ ($u = 0,1,...,7$) can be classified into four groups which correspond to $h_u = 0,1,2,\lambda$ respectively and each group can be expressed by a general form and there is only 2^{2-h_u} different absolute values in $\boldsymbol{G}_u$ (here, we can let $\lambda = 2$ because there is one absolute value in $\boldsymbol{G}_0$). Since the vectors $\boldsymbol{G}_u$ and $\boldsymbol{G}_v$ ($u,v = 0,1,...,7$) can be classified into four groups respectively, we can also classify the basic images $\boldsymbol{T}_{uv}$ ($u,v = 0,1,2,...,7$) into 16 groups, see Appendix A. In each group, the basic images can be expressed by the same general form. There is some symmetries in the vectors $\boldsymbol{G}_u$ and $\boldsymbol{G}_v$, and we can also find the relevant symmetries in the basic images $\boldsymbol{T}_{uv} = \boldsymbol{G}_u{}^T \times \boldsymbol{G}_v$, that is, in $\boldsymbol{T}_{uv}$, the symmetry of each column is the same as $\boldsymbol{G}_u$ and the symmetry of each row is the same as $\boldsymbol{G}_v$. Fig.1 illustrates the general form of $\boldsymbol{T}_{12}$, $\boldsymbol{T}_{16}$, $\boldsymbol{T}_{32}$, $\boldsymbol{T}_{36}$, $\boldsymbol{T}_{52}$, $\boldsymbol{T}_{56}$, $\boldsymbol{T}_{72}$ and $\boldsymbol{T}_{76}$. Every one of them can be divided into 8 sub-matrixes $\boldsymbol{L}_1 \sim \boldsymbol{L}_8$ (ref. Fig.1(a)(b)), and the values of their elements are different permutation of $|A|,|B|,|C|,...|H|$. The relation between $\boldsymbol{L}_1$ and $\boldsymbol{L}_2$ in $[\boldsymbol{L}_1 \vdots \boldsymbol{L}_2]$ is an odd symmetry which takes "$\vdots$" as the symmetry axis, and the relation between $\boldsymbol{L}_1$ and $\boldsymbol{L}_5$ in $\begin{bmatrix} \boldsymbol{L}_1 \\ \hline \boldsymbol{L}_5 \end{bmatrix}$ is also an odd symmetry which takes "---" as the symmetry axis. Similarly, the relation between $[\boldsymbol{L}_1 \ \boldsymbol{L}_2]$ and $[\boldsymbol{L}_3 \ \boldsymbol{L}_4]$ in $[\boldsymbol{L}_1 \ \boldsymbol{L}_2 \vdots \boldsymbol{L}_3 \ \boldsymbol{L}_4]$ is an even symmetry, etc. $\boldsymbol{T}_{uv}$ can be therefore obtained by extending $\boldsymbol{L}_1$ according to symmetry. The other groups of basic images also have the similar symmetries.

$$\left[\begin{array}{cc|cc|cc|cc}
\mathrm{A} & E & -E & -A & -A & -E & E & A \\
B & F & -F & -B & -B & -F & F & B \\
C & G & -G & -C & -C & -G & G & C \\
D & H & -H & -D & -D & -H & H & D \\
\hline
-D & -H & H & D & D & H & -H & -D \\
-C & -G & G & C & C & G & -G & -C \\
-B & -F & F & B & B & F & -F & -B \\
-A & -E & \mathrm{E} & A & A & \mathrm{E} & -E & -A
\end{array}\right]$$

(a)

$$\left[\begin{array}{c|c|c|c}
\boldsymbol{L}_1 & \boldsymbol{L}_2 & \boldsymbol{L}_3 & \boldsymbol{L}_4 \\
\hline
\boldsymbol{L}_5 & \boldsymbol{L}_6 & \boldsymbol{L}_7 & \boldsymbol{L}_8
\end{array}\right]$$

(b)

Fig.1 the coefficients in $\boldsymbol{T}_{12}$,$\boldsymbol{T}_{16}$,$\boldsymbol{T}_{32}$, $\boldsymbol{T}_{36}$, $\boldsymbol{T}_{52}$,$\boldsymbol{T}_{56}$,$\boldsymbol{T}_{72}$ and $\boldsymbol{T}_{76}$ can be grouped into 8 blocks, the elements in each block have the same absolute value $|A|,|B|,....|H|$. (a) the general form of the 14th group; (b) $\boldsymbol{L}_1 \sim \boldsymbol{L}_8$ represent 8 sub-matrix in (a).

Definition 1. A group of data is called the **smallest unit** $\overline{\boldsymbol{T}}_{uv}$ of $\boldsymbol{T}_{uv}$, if it is a smallest data set, and by extending which $\boldsymbol{T}_{uv}$ can be obtained.

In Appendix.A, the sub-matrices in the dashed rectangles of Fig.A1~A20 are the smallest units. Obviously, according to the above analyses, we can get the properties of 2D 8×8 DCT basic images as follows.

Property 1. If two basic images $\boldsymbol{T}_{uv}$ have the same (h_u, h_v), they have the same symmetry, and all the 64 basic images of 2D 8×8 DCT can be classified as 16 groups according to their symmetries (ref Appendix.A).

Property 2. The size of the smallest unit $\overline{\boldsymbol{T}}_{uv}$ is $2^{2-h_u} \times 2^{2-h_v}$ (here let $\lambda = 2$).

Now, we take $\boldsymbol{T}_{16}$ as an example to explain how to reduce the operations of Formula (4) by exploiting the symmetries in basic images. From the expressions of $\boldsymbol{G}_1, \boldsymbol{G}_6$ and $\boldsymbol{T}_{16} = \boldsymbol{G}_1{}^T \times \boldsymbol{G}_6$, it follows that

$$\overline{\boldsymbol{T}}_{16} = \begin{bmatrix} A & E \\ B & F \\ C & G \\ D & H \end{bmatrix} = \begin{bmatrix} \frac{1}{4}\cos(\frac{\pi}{8}) \\ \frac{1}{4}\cos(\frac{3\pi}{8}) \\ \frac{1}{4}\cos(\frac{5\pi}{8}) \\ \frac{1}{4}\cos(\frac{7\pi}{8}) \end{bmatrix} \times \begin{bmatrix} \frac{1}{2}\cos(\frac{3\pi}{8}) & -\frac{1}{2}\cos(\frac{\pi}{8}) \end{bmatrix}$$

If the terms with the same coefficients in (4) are combined, the formula of computing $F(1,6)$ can be computed by the following simple formula.

$$F(1,6) = \sum_{m=0}^{7}\sum_{n=0}^{7}(f(m,n)\cdot t_{16}(m,n)) = \boldsymbol{f} \odot \boldsymbol{T}_{16} = \overline{\boldsymbol{f}}_{16} \odot \overline{\boldsymbol{T}}_{16}$$

where the matrix $\overline{\boldsymbol{f}}_{16}$ is

$$\overline{\boldsymbol{f}}_{16} = \begin{bmatrix} f_A & f_E \\ f_B & f_F \\ f_C & f_G \\ f_D & f_H \end{bmatrix},$$

with

$$f_A = \begin{bmatrix} 1 & 0 & 0 & -1 & -1 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ -1 & 0 & 0 & 1 & 1 & 0 & 0 & -1 \end{bmatrix} \odot \begin{bmatrix} f(0,0) & f(0,1) & \cdots & f(0,7) \\ f(1,0) & f(1,1) & \cdots & f(1,7) \\ \vdots & \vdots & & \vdots \\ f(7,0) & f(7,1) & \cdots & f(7,7) \end{bmatrix}$$

$$= f(0,0) - f(0,3) - f(0,4) + f(0,7) - f(7,0) + f(7,3) + f(7,4) - f(7,7)$$

The first matrix in right side can be obtained by simply replacing 1 for the element A, and 0 for the other elements of the matrix in Fig.1(a). Similarly we can get $f_B, f_C, \ldots\ldots f_H$.

In general, we obtain

$$F(u,v) = \sum_{m=0}^{7}\sum_{n=0}^{7}(f(m,n)\cdot t_{uv}(m,n)) = \boldsymbol{f} \odot \boldsymbol{T}_{uv}$$
$$= \overline{\boldsymbol{f}}_{uv} \odot \overline{\boldsymbol{T}}_{uv} \qquad (9)$$

3.2 Fast Calculation of the Matrix $\overline{\boldsymbol{f}}_{uv}$

From Formula (5)~(8), $\boldsymbol{G}_u$ $(u = 0,1,2,\ldots,7)$ can be classified into four groups and all the vectors in each group have the same symmetry. In accordance, there are four kinds of symmetries in all the rows or columns of basic images, that is, there are four cases of

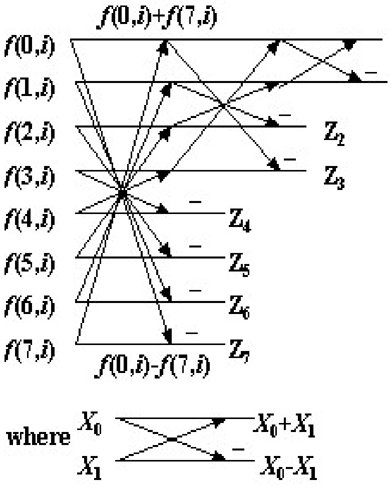


Fig. 2 Flow of merging each row or column

merging the elements of each row or column in the process of computing $\overline{f}_{uv}\ (u,v=0,1,2,...,7)$. Let f_i be the ith column vector of f, that is

$$f_i=[f(0,i),f(1,i),f(2,i),f(3,i),f(4,i),f(5,i),f(6,i),f(7,i)]^T$$

The four merging cases are presented as follows:

① $\overline{f_i^{\,0}}=[\sum_{j=0}^{7}f(j,i)]$ when $h_u=\lambda$;

② $\overline{f_i^{\,1}}=[\sum_{j=0}^{7}(-1)^j f(j,i)]$ when $h_u=2$;

③ $\overline{f_i^{\,2}}=\begin{bmatrix}f(0,i)+f(7,i)-f(3,i)-f(4,i)\\ f(1,i)+f(6,i)-f(2,i)-f(5,i)\end{bmatrix}$ when $h_u=1$;

④ $\overline{f_i^{\,3}}=\begin{bmatrix}f(0,i)-f(7,i)\\ f(1,i)-f(6,i)\\ f(2,i)-f(5,i)\\ f(3,i)-f(4,i)\end{bmatrix}$ when $h_u=0$.

In order to reduce arithmetic operations, we propose that f_i is processed in-place to obtain $\overline{f_i^{\,0}},\overline{f_i^{\,1}},\overline{f_i^{\,2}}$ *and* $\overline{f_i^{\,3}}$ fast according to the flow of Fig.2 which needs 14 additions. The process consists of three steps.

Step 1: $(k,i)+(7-k,i)\rightarrow(k,i)$ and
$(k,i)-(7-k,i)\rightarrow(7-k,i)$ (k=0,1,2,3)

Step 2: $(k,i)+(3-k,i)\rightarrow(k,i)$ and
$(k,i)-(3-k,i)\rightarrow(3-k,i)$ (k=0,1)

Step 3: $(0,i)+(1,i)\rightarrow(0,i)$
$(0,i)-(1,i)\rightarrow(1,i)$

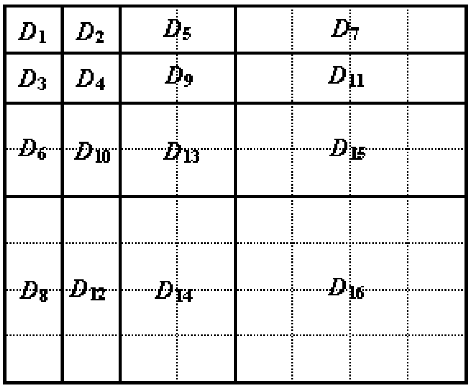


Fig.3 Distribution of $\bar{f}_{uv}$ in matrix ***D***

where "$(j,i)+(k,i)\rightarrow(m,i)$" means that the (j,i) element is added to the (k,i) element and the sum is placed in the (m,i) element of the matrix f. After the above three steps, matrix $\overline{f_i'}$ is obtained as follows, which consists of $\overline{f_i^{\,0}},\overline{f_i^{\,1}},\overline{f_i^{\,2}}$ *and* $\overline{f_i^{\,3}}$.

$$\overline{f_i'}=\begin{bmatrix}z_0\\ z_1\\ z_2\\ z_3\\ z_4\\ z_5\\ z_6\\ z_7\end{bmatrix}=\begin{bmatrix}\sum_{j=0}^{7}f(j,i)\\ \sum_{j=0}^{7}(-1)^j f(j,i)\\ f(1,i)+f(6,i)-f(2,i)-f(5,i)\\ f_0(0,i)+f(7,i)-f(3,i)-f(4,i)\\ f(3,i)-f(4,i)\\ f(2,i)-f(5,i)\\ f(1,i)-f(6,i)\\ f(0,i)-f(7,i)\end{bmatrix}=\begin{bmatrix}\overline{f_i^{\,0}}^{\,\mathrm{c}}\\ \overline{f_i^{\,1}}^{\,\mathrm{c}}\\ \overline{f_i^{\,2}}^{\,\mathrm{c}}\\ \overline{f_i^{\,3}}^{\,\mathrm{c}}\end{bmatrix}$$

where the matrix $\boldsymbol{X}^{\mathrm{C}}$ is obtained from the matrix $\boldsymbol{X}$. That is, its element $X^{\mathrm{C}}(j)=X(N\text{-}1\text{-}j)$, where

N is the row number of $\boldsymbol{X}$.

In each group of $\boldsymbol{T}_{uv}$ ($u,v=0,1,2,...,7$), the corresponding matrices $\overline{\boldsymbol{f}}_{uv}$ are the same, therefore, it only needs to compute 16 different $\overline{\boldsymbol{f}}_{uv}$. If all the rows and columns of $\boldsymbol{f}$ are processed in turn according to the flow of Fig.2, the matrix $\boldsymbol{D}$, consisting of 16 sub-matrices $\boldsymbol{D}_i$ $(i=1,2,...,16)$, is obtained. Fig. 3 shows the distribution of $\boldsymbol{D}_i$ $(i=1,2,...,16)$ in the matrix $\boldsymbol{D}$. The corresponding relationship between $\boldsymbol{D}_i$ $(i=1,2,...,16)$ and $\overline{\boldsymbol{f}}_{uv}$ ($u,v=0,1,...,7$) are shown as follows:

$$\begin{aligned}
&\boldsymbol{D}_1^C=\overline{\boldsymbol{f}}_{00},\quad \boldsymbol{D}_2^C=\overline{\boldsymbol{f}}_{04},\ \boldsymbol{D}_3^C=\overline{\boldsymbol{f}}_{40},\ \boldsymbol{D}_4^C=\overline{\boldsymbol{f}}_{44},\\
&\boldsymbol{D}_5^C=\overline{\boldsymbol{f}}_{02}=\overline{\boldsymbol{f}}_{06},\quad \boldsymbol{D}_6^C=\overline{\boldsymbol{f}}_{20}=\overline{\boldsymbol{f}}_{60},\quad \boldsymbol{D}_7^C=\overline{\boldsymbol{f}}_{01}=\overline{\boldsymbol{f}}_{03}=\overline{\boldsymbol{f}}_{05}=\overline{\boldsymbol{f}}_{07},\\
&\boldsymbol{D}_8^C=\overline{\boldsymbol{f}}_{10}=\overline{\boldsymbol{f}}_{30}=\overline{\boldsymbol{f}}_{50}=\overline{\boldsymbol{f}}_{70},\quad \boldsymbol{D}_9^C=\overline{\boldsymbol{f}}_{42}=\overline{\boldsymbol{f}}_{46},\qquad \boldsymbol{D}_{10}^C=\overline{\boldsymbol{f}}_{24}=\overline{\boldsymbol{f}}_{64},\\
&\boldsymbol{D}_{11}^C=\overline{\boldsymbol{f}}_{41}=\overline{\boldsymbol{f}}_{43}=\overline{\boldsymbol{f}}_{45}=\overline{\boldsymbol{f}}_{47},\ \boldsymbol{D}_{12}^C=\overline{\boldsymbol{f}}_{14}=\overline{\boldsymbol{f}}_{34}=\overline{\boldsymbol{f}}_{54}=\overline{\boldsymbol{f}}_{74},\\
&\boldsymbol{D}_{13}^C=\overline{\boldsymbol{f}}_{22}=\overline{\boldsymbol{f}}_{66}=\overline{\boldsymbol{f}}_{26}=\overline{\boldsymbol{f}}_{62},\ \boldsymbol{D}_{14}^C=\overline{\boldsymbol{f}}_{12}=\overline{\boldsymbol{f}}_{32}=\overline{\boldsymbol{f}}_{52}=\overline{\boldsymbol{f}}_{72}=\overline{\boldsymbol{f}}_{16}=\overline{\boldsymbol{f}}_{36}=\overline{\boldsymbol{f}}_{56}=\overline{\boldsymbol{f}}_{76},\qquad (10)\\
&\boldsymbol{D}_{15}^C=\overline{\boldsymbol{f}}_{21}=\overline{\boldsymbol{f}}_{23}=\overline{\boldsymbol{f}}_{25}=\overline{\boldsymbol{f}}_{27}=\overline{\boldsymbol{f}}_{61}=\overline{\boldsymbol{f}}_{63}=\overline{\boldsymbol{f}}_{65}=\overline{\boldsymbol{f}}_{67},\\
&\boldsymbol{D}_{16}^C=\overline{\boldsymbol{f}}_{11}=\overline{\boldsymbol{f}}_{33}=\overline{\boldsymbol{f}}_{55}=\overline{\boldsymbol{f}}_{77}=\overline{\boldsymbol{f}}_{15}=\overline{\boldsymbol{f}}_{31}=\overline{\boldsymbol{f}}_{57}=\overline{\boldsymbol{f}}_{73}\\
&\qquad\quad=\overline{\boldsymbol{f}}_{17}=\overline{\boldsymbol{f}}_{35}=\overline{\boldsymbol{f}}_{53}=\overline{\boldsymbol{f}}_{71}=\overline{\boldsymbol{f}}_{13}=\overline{\boldsymbol{f}}_{37}=\overline{\boldsymbol{f}}_{51}=\overline{\boldsymbol{f}}_{75}
\end{aligned}$$

where the matrix $\boldsymbol{D}_i^{\mathrm{C}}$ is obtained from the matrix $\boldsymbol{D}_i$. That is, its element D_i^{C} (j, k) $=D_i$ $(N\text{-}1\text{-}j, M\text{-}1\text{-}k)$, where M or N is the row or column number of $\boldsymbol{D}_i$**.** The whole merging process requires 14×16=224 additions to obtain the matrix $\boldsymbol{D}$.

3.3 Fast Computing of $F(u,v)=\overline{\boldsymbol{f}}_{uv}\odot\overline{\boldsymbol{T}}_{uv}$

In order to decrease multiplication operations for computing $F(u,v)=\overline{\boldsymbol{f}}_{uv}\odot\overline{\boldsymbol{T}}_{uv}$, the element $t_{uv}(m,n)$ in $\overline{\boldsymbol{T}}_{uv}$ is decomposed as follows.

$$\begin{aligned}
t_{uv}(m,n) &= \frac{1}{4}C(u)C(v)\cos\frac{(2m+1)u\pi}{16}\cos\frac{(2n+1)v\pi}{16}\\
&=\frac{1}{8}C(u)C(v)\left[\cos\frac{\pi(2m+1)u+\pi(2n+1)v}{16}+\cos\frac{\pi(2m+1)u-\pi(2n+1)v}{16}\right] \qquad (11)\\
&=\frac{1}{8}C(u)C(v)\left[\cos\frac{\pi(\alpha+\beta)}{2^{4-h}}+\cos\frac{\pi(\alpha-\beta)}{2^{4-h}}\right]
\end{aligned}$$

where $\alpha=(2m+1)\cdot 2^{h_u-h}\cdot L_u$, $\beta=(2n+1)\cdot 2^{h_v-h}\cdot L_v$, $h=\min(h_u,h_v)$.

Now let us discuss the values of $\cos\frac{(\alpha+\beta)\cdot\pi}{2^{4-h}}$ and $\cos\frac{(\alpha-\beta)\cdot\pi}{2^{4-h}}$ in Formula (11).

(1) For $\cos(\frac{k\pi}{2^{4-h}})$ (k is an integer, $k=0,1,...,2^{3-h}$), its absolute value can only be one of $(2^{3-h}+1)$ different values. In the case of $k=2^{3-h}$, it equals to zero and no operation is needed when it is multiplied by any value.

(2) Assuming $h=h_u$, then $(2^{h_u-h}\cdot L_u)=L_u$. Both $(\alpha+\beta)$ and $(\alpha-\beta)$ are even integers if $(2^{h_v-h}\cdot L_v)$ is an odd integer. Both $(\alpha+\beta)$ and $(\alpha-\beta)$ are odd integers if $(2^{h_v-h}\cdot L_v)$ is

an even integer. Whether ($\alpha+\beta$) and ($\alpha-\beta$) are even or odd is independent on the value of *m* and *n*, and it depends only on u and v. The results are similar assuming $h=h_v$.

According to the above discussion, for given u and v, when the values of *m* and *n* are taken from 0 to 7, $\left|\cos\frac{(\alpha+\beta)\cdot\pi}{2^{4-h}}\right|$ and $\left|\cos\frac{(\alpha-\beta)\cdot\pi}{2^{4-h}}\right|$ can only have $\left(2^{3-h}\right)/2=2^{2-h}$ values except zero. 2^{2-h} multiplication operations are needed to compute $\overline{\boldsymbol{f}}_{uv}\odot\overline{\boldsymbol{T}}_{\boldsymbol{uv}}$.

As for the first 12 groups of basic images, one of their *u* and *v* must be 0 or 4, and each $\overline{\boldsymbol{T}}_{uv}$ of them consists of $2^{2-h_u}\times 2^{2-h_v}=2^{2-h}$ elements according to Section 3.1.1. Since the number of multiplication operations for each DCT coefficient is also 2^{2-h} by decomposing $\overline{\boldsymbol{T}}_{uv}$, the first 12 groups of basic images need not be decomposed, and we compute the DCT coefficients in the first 12 groups directly by using $F(u,v)=\overline{\boldsymbol{f}}_{uv}\odot\overline{\boldsymbol{T}}_{\boldsymbol{uv}}$. As for the 13th to 16th groups of basic images, in order to reduce multiplication operations, each $\overline{\boldsymbol{T}}_{uv}$ requires decomposing according to Formula (11) because the size of $\overline{\boldsymbol{T}}_{uv}$ in these groups is more than 2^{2-h}.

3.3.1 Decomposing $\overline{\boldsymbol{T}}_{uv}$ in the 13th to 16th Groups of Basic Images

Each group of the basic images can be further classified according to their decomposing expressions. For example, the 14th group of basic images consists of 8 basic images which can be classified into two sub-groups: one includes $\overline{\boldsymbol{T}}_{12}$, $\overline{\boldsymbol{T}}_{36}$, $\overline{\boldsymbol{T}}_{56}$ and $\overline{\boldsymbol{T}}_{72}$ and another includes $\overline{\boldsymbol{T}}_{32}$, $\overline{\boldsymbol{T}}_{16}$, $\overline{\boldsymbol{T}}_{52}$ and $\overline{\boldsymbol{T}}_{76}$. As an example, we discuss the first sub-group to explain why they belong to the same sub-group.

$\overline{\boldsymbol{T}}_{12}$ can be decomposed as follows:

$$\overline{\boldsymbol{T}}_{12}=\frac{\gamma(1)}{8}\begin{bmatrix}1&0\\1&0\\0&1\\0&1\end{bmatrix}+\frac{\gamma(3)}{8}\begin{bmatrix}1&0\\0&1\\1&0\\0&-1\end{bmatrix}+\frac{\gamma(5)}{8}\begin{bmatrix}0&1\\1&0\\0&-1\\1&0\end{bmatrix}+\frac{\gamma(7)}{8}\begin{bmatrix}0&1\\0&-1\\1&0\\-1&0\end{bmatrix}$$

$$=\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_3+\frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_3+\frac{\gamma(5)}{8}\cdot\boldsymbol{R}_3+\frac{\gamma(7)}{8}\cdot\boldsymbol{S}_3 \tag{12}$$

where $\gamma(k)=\cos(\pi k/16)$.

From (9) and (12), it follows

$$F(1,2)=\overline{\boldsymbol{f}}_{12}\odot\overline{\boldsymbol{T}}_{12}\ =\ \boldsymbol{D}_{14}^{C}\odot\left(\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_3+\frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_3+\frac{\gamma(5)}{8}\cdot\boldsymbol{R}_3+\frac{\gamma(7)}{8}\cdot\boldsymbol{S}_3\right)$$

$$=\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_3\odot\boldsymbol{D}_{14}^{C}+\frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_3\odot\boldsymbol{D}_{14}^{C}+\frac{\gamma(5)}{8}\cdot\boldsymbol{R}_3\odot\boldsymbol{D}_{14}^{C}+\frac{\gamma(7)}{8}\cdot\boldsymbol{S}_3\odot\boldsymbol{D}_{14}^{C}$$

After $\boldsymbol{P}_3\odot\boldsymbol{D}_{14}^{C}$, $\boldsymbol{Q}_3\odot\boldsymbol{D}_{14}^{C}$, $\boldsymbol{R}_3\odot\boldsymbol{D}_{14}^{C}$ and $\boldsymbol{S}_3\odot\boldsymbol{D}_{14}^{C}$ being computed by addition operators, then four multiplications and 3 additions are needed to compute $F(1,2)$.

Similarly, we obtain

$$\overline{\boldsymbol{T}}_{36}=\frac{\gamma(3)}{8}\cdot\boldsymbol{P}_3-\frac{\gamma(7)}{8}\cdot\boldsymbol{Q}_3-\frac{\gamma(1)}{8}\cdot\boldsymbol{R}_3-\frac{\gamma(5)}{8}\cdot\boldsymbol{S}_3$$

$$\overline{T}_{56} = -\frac{\gamma(5)}{8}\cdot \boldsymbol{P}_3 + \frac{\gamma(1)}{8}\cdot \boldsymbol{Q}_3 - \frac{\gamma(7)}{8}\cdot \boldsymbol{R}_3 - \frac{\gamma(3)}{8}\cdot \boldsymbol{S}_3$$
$$\overline{T}_{72} = -\frac{\gamma(7)}{8}\cdot \boldsymbol{P}_3 + \frac{\gamma(5)}{8}\cdot \boldsymbol{Q}_3 - \frac{\gamma(3)}{8}\cdot \boldsymbol{R}_3 - \frac{\gamma(1)}{8}\cdot \boldsymbol{S}_3$$

and

$$F(3,6) = \frac{\gamma(3)}{8}\cdot \boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(7)}{8}\cdot \boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(1)}{8}\cdot \boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(5)}{8}\cdot \boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C$$
$$F(5,6) = \frac{\gamma(5)}{8}\cdot \boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(1)}{8}\cdot \boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(7)}{8}\cdot \boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(3)}{8}\cdot \boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C$$
$$F(7,2) = -\frac{\gamma(7)}{8}\cdot \boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(5)}{8}\cdot \boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(3)}{8}\cdot \boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C - \frac{\gamma(1)}{8}\cdot \boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C$$

The above formulas show that $\boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C, \boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C, \boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C$ and $\boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C$ are repeatedly used to compute $F(1,2)$, $F(3,6)$, $F(5,6)$ and $F(7,2)$. Similarly all $\overline{\boldsymbol{T}}_{uv}$ in the 13~16th groups of basic images can be decomposed in the same manner (See Appendix.B) and each group can be further classified into two or four sub-groups respectively (See Table 2). We can restore the results of $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}, \boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ in the process of computing the low frequency coefficients and then use them directly for the computation of the higher frequency coefficients in the same subgroups.

Table.2 Sub-groups in the 13~16th groups of basic images

<table>
<tr><th>Group NO.</th><th colspan="4">Sub-groups</th></tr>
<tr><td>The 13th group</td><td colspan="2">$\overline{f}_{22}$, $\overline{f}_{66}$</td><td colspan="2">$\overline{f}_{26}$, $\overline{f}_{62}$</td></tr>
<tr><td>The 14th group</td><td colspan="2">$\overline{f}_{12}, \overline{f}_{72}, \overline{f}_{36}, \overline{f}_{56}$</td><td colspan="2">$\overline{f}_{32}, \overline{f}_{52}, \overline{f}_{16}, \overline{f}_{76}$</td></tr>
<tr><td>The 15th group</td><td colspan="2">$\overline{f}_{21}, \overline{f}_{27}, \overline{f}_{63}, \overline{f}_{65}$</td><td colspan="2">$\overline{f}_{23}, \overline{f}_{25}, \overline{f}_{61}, \overline{f}_{67}$</td></tr>
<tr><td>The 16th group</td><td>$\overline{f}_{11}, \overline{f}_{33}$
$\overline{f}_{55}, \overline{f}_{77}$</td><td>$\overline{f}_{13}, \overline{f}_{37}$
$\overline{f}_{51}, \overline{f}_{75}$</td><td>$\overline{f}_{17}, \overline{f}_{35}$
$\overline{f}_{53}, \overline{f}_{71}$</td><td>$\overline{f}_{15}, \overline{f}_{31}$
$\overline{f}_{57}, \overline{f}_{73}$</td></tr>
</table>

3.3.2 Computing $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$

From Formula (10), we know that all the matrices $\overline{\boldsymbol{f}}_{uv}$ in the kth group equal to $\boldsymbol{D}_k^C$. In this section, according to the decomposing expressions of $\overline{\boldsymbol{T}}_{uv}$ in Appendix.B, we present our method to compute $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ as follows.

(1) Computing $\boldsymbol{P}_i \odot \boldsymbol{D}_{13}^C$ and $\boldsymbol{Q}_i \odot \boldsymbol{D}_{13}^C$ of the 13th group

$b1 = D(3, 3) - D(2, 2)$; $b2 = D(3, 2) + D(2, 3)$

$\boldsymbol{P}_1 \odot \boldsymbol{D}_{13}^C = D(2, 2) + D(3, 3)$; $\boldsymbol{Q}_1 \odot \boldsymbol{D}_{13}^C = b1 + b2$

$\boldsymbol{P}_2 \odot \boldsymbol{D}_{13}^C = D(3, 2) - D(2, 3)$; $\boldsymbol{Q}_2 \odot \boldsymbol{D}_{13}^C = b1 - b2$

(2) Computing $\boldsymbol{P}_i \odot \boldsymbol{D}_{14}^C$, $\boldsymbol{Q}_i \odot \boldsymbol{D}_{14}^C, \boldsymbol{R}_i \odot \boldsymbol{D}_{14}^C$ and $\boldsymbol{S}_i \odot \boldsymbol{D}_{14}^C$ of the 14th group

$n1 = D(4, 2) + D(7, 3)$; $n2 = D(6, 3) + D(5, 2)$; $n3 = D(7, 3) - D(4, 2)$
$n4 = D(5, 3) + D(6, 2)$; $n5 = D(7, 2) + D(4, 3)$; $n6 = D(6, 3) - D(5, 2)$
$n7 = D(7, 2) - D(4, 3)$; $n8 = D(5, 3) - D(6, 2)$

$\boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C = n1 + n2$; $\boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C = n3 + n4$; $\boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C = n5 + n6$

$\boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C = n7 + n8$; $\boldsymbol{P}_4 \odot \boldsymbol{D}_{14}^C = n6 - n5$; $\boldsymbol{Q}_4 \odot \boldsymbol{D}_{14}^C = n1 - n2$

$\boldsymbol{R}_4 \odot \boldsymbol{D}_{14}^C = n8 - n7$; $\boldsymbol{S}_4 \odot \boldsymbol{D}_{14}^C = n3 - n4$

(3) Computing $\boldsymbol{P}_i \odot \boldsymbol{D}_{15}^C, \boldsymbol{Q}_i \odot \boldsymbol{D}_{15}^C, \boldsymbol{R}_i \odot \boldsymbol{D}_{15}^C$ and $\boldsymbol{S}_i \odot \boldsymbol{D}_{15}^C$ of the 15th group

$m1 = D(2, 4) + D(3, 7)$; $m2 = D(3, 6) + D(2, 5)$; $m3 = D(3, 7) - D(2, 4)$
$m4 = D(3, 5) + D(2, 6)$; $m5 = D(2, 7) + D(3, 4)$; $m6 = D(3, 6) - D(2, 5)$
$m7 = D(2, 7) - D(3, 4)$; $m8 = D(3, 5) - D(2, 6)$;

$\boldsymbol{P}_5 \odot \boldsymbol{D}_{15}^C = m1 + m2$; $\boldsymbol{Q}_5 \odot \boldsymbol{D}_{15}^C = m3 + m4$; $\boldsymbol{R}_5 \odot \boldsymbol{D}_{15}^C = m5 + m6$

$\boldsymbol{S}_5 \odot \boldsymbol{D}_{15}^C = m7 + m8$; $\boldsymbol{P}_6 \odot \boldsymbol{D}_{15}^C = m6 - m5$; $\boldsymbol{Q}_6 \odot \boldsymbol{D}_{15}^C = m1 - m2$

$\boldsymbol{R}_6 \odot \boldsymbol{D}_{15}^C = m8 - m7$; $\boldsymbol{S}_6 \odot \boldsymbol{D}_{15}^C = m3 - m4$

(4) Computing $\boldsymbol{P}_i \odot \boldsymbol{D}_{16}^C$, $\boldsymbol{Q}_i \odot \boldsymbol{D}_{16}^C, \boldsymbol{R}_i \odot \boldsymbol{D}_{16}^C$ and $\boldsymbol{S}_i \odot \boldsymbol{D}_{16}^C$ of the 16th group

$a1 = D(4, 4) + D(7, 7)$; $a2 = D(5, 5) + D(6, 6)$; $a3 = D(7, 4) + D(4, 7)$
$a4 = D(7, 5) + D(5, 7)$; $a5 = D(6, 4) + D(4, 6)$; $a6 = D(6, 6) - D(5, 5)$
$a7 = D(6, 7) + D(7, 6)$; $a8 = D(4, 5) + D(5, 4)$; $a9 = D(7, 7) - D(4, 4)$
$a10 = D(6, 5) + D(5, 6)$; $aa1 = a3 + a6$; $aa2 = a4 - a5$; $aa3 = a7 - a8$;
$aa4 = a4 + a5$; $aa5 = a7 + a8$; $aa6 = a9 + a10$;

$\boldsymbol{P}_7 \odot \boldsymbol{D}_{16}^C = a1 + a2$; $\boldsymbol{Q}_7 \odot \boldsymbol{D}_{16}^C = aa1 + aa2$; $\boldsymbol{R}_7 \odot \boldsymbol{D}_{16}^C = aa3 + aa4$;

$\boldsymbol{S}_7 \odot \boldsymbol{D}_{16}^C = aa5 + aa6$

$a11 = D(6, 7) - D(5, 4)$; $a12 = D(7, 5) + D(4, 6)$; $a13 = D(4, 7) - D(7, 4)$
$a14 = D(5, 6) - D(6, 5)$; $a15 = D(6, 7) - D(7, 6)$; $a16 = D(5, 4) - D(4, 5)$
$a17 = D(5, 7) + D(4, 6)$; $a18 = D(7, 5) + D(6, 4)$; $aa7 = a1 - a2$; $aa8 = a13 + a14$;
$aa9 = a6 - a3$; $aa10 = a15 + a16$; $aa11 = a9 - a10$; $aa12 = a17 - a18$;

$\boldsymbol{P}_8 \odot \boldsymbol{D}_{16}^C = a11 - a12$; $\boldsymbol{Q}_8 \odot \boldsymbol{D}_{16}^C = aa7 + aa8$; $\boldsymbol{R}_8 \odot \boldsymbol{D}_{16}^C = aa9 + aa10$

$\boldsymbol{S}_8 \odot \boldsymbol{D}_{16}^C = aa11 + aa12$

$a19 = D(7, 6) - D(4, 5)$; $a20 = D(6, 4) + D(5, 7)$

$\boldsymbol{P}_9 \odot \boldsymbol{D}_{16}^C = a20 - a19$; $\boldsymbol{Q}_9 \odot \boldsymbol{D}_{16}^C = aa11 - aa12$; $\boldsymbol{R}_9 \odot \boldsymbol{D}_{16}^C = aa7 - aa8$

$\boldsymbol{S}_9 \odot \boldsymbol{D}_{16}^C = aa9 - aa10$

$aa13 = a15 - a16$; $aa14 = a12 - a20$

$\boldsymbol{P}_{10} \odot \boldsymbol{D}_{16}^C = a14 - a13$; $\boldsymbol{Q}_{10} \odot \boldsymbol{D}_{16}^C = aa6 - aa5$; $\boldsymbol{R}_{10} \odot \boldsymbol{D}_{16}^C = aa13 + aa14$

$\boldsymbol{S}_{10} \odot \boldsymbol{D}_{16}^C = aa1 - aa2$

3.3.3 The Fast Method of Computing DCT Coefficients in the 13th and 16th Groups

After $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}, \boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ **and** $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ are obtained, we can compute the DCT coefficients in the 13th ~16th groups. By analyzing the decomposed expressions in Appendix.B, a fast method to compute the DCT coefficients in the 13th and 16th groups can be summarized as follows:

(1) DCT coefficients in the 13th group

According to Appendix.B, we know that

$F(2,2) = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_1 \odot \boldsymbol{D}_{13}^C + \frac{\gamma(4)}{8} \cdot \boldsymbol{Q}_1 \odot \boldsymbol{D}_{13}^C$ and

$F(6,6) = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_1 \odot \boldsymbol{D}_{13}^C - \frac{\gamma(4)}{8} \cdot \boldsymbol{Q}_1 \odot \boldsymbol{D}_{13}^C$.

Let $pa1 = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_1 \odot \boldsymbol{D}_{13}^C$ and $pa2 = \frac{\gamma(4)}{8} \cdot \boldsymbol{Q}_1 \odot \boldsymbol{D}_{13}^C$**,** then

$F(2,2) = pa1 + pa2$, $F(6,6) = pa1 - pa2$

Similarly, let $pa3 = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_2 \odot \boldsymbol{D}_{13}^C$ and $pa4 = \frac{\gamma(4)}{8} \cdot \boldsymbol{Q}_2 \odot \boldsymbol{D}_{13}^C$, then

$F(2,6) = pa3 - pa4$, $F(6,2) = -pa3 - pa4$

Now, 4 multiplications and 4 additions are needed for computing $F(2,2)$, $F(2,6)$, $F(6,2)$ and $F(6,6)$.

(2) DCT coefficients in the 16th group

As for the ($F(1,1), F(3,3), F(5,5), F(7,7)$) subgroup, let

$$pb1 = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_7 \odot \boldsymbol{D}_{16}^C, \quad pb2 = \frac{\gamma(6)}{8} \cdot \boldsymbol{Q}_7 \odot \boldsymbol{D}_{16}^C, \quad pb3 = \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_7 \odot \boldsymbol{D}_{16}^C,$$

$$pb4 = \frac{\gamma(2)}{8} \cdot \boldsymbol{S}_7 \odot \boldsymbol{D}_{16}^C, \quad pb5 = \frac{\gamma(2)}{8} \cdot \boldsymbol{Q}_7 \odot \boldsymbol{D}_{16}^C, \quad pb6 = \frac{\gamma(6)}{8} \cdot \boldsymbol{S}_7 \odot \boldsymbol{D}_{16}^C,$$

$ppb1 = pb1 + pb3$, $ppb2 = pb2 + pb4$, $ppb3 = pb1 - pb3$, $ppb4 = pb6 - pb5$, then

$F(1,1) = ppb1 + ppb2$, $F(3,3) = ppb3 + ppb4$,

$F(5,5) = ppb3 - ppb4$, $F(7,7) = ppb1 - ppb2$.

By the similar methods, we can compute the other subgroups in the 16th group and each subgroup only needs 6 multiplications and 8 additions.

As for the DCT coefficients in the 14th and 15th groups, we can compute them directly by using the formula as follows:

$$F(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \overline{\boldsymbol{T}}_{uv}$$

$$= \frac{\gamma(k_1)}{8} \cdot \boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv} - \frac{\gamma(k_2)}{8} \cdot \boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv} - \frac{\gamma(k_3)}{8} \cdot \boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv} - \frac{\gamma(k_4)}{8} \cdot \boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$$

3.4 Analyzing the Operation Number of the New Algorithm

The arithmetic operations for computing all $F(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \overline{\boldsymbol{T}}_{uv}$ ($u, v = 0,1,2,...,7$) are analyzed as follows：

(1) Multiplication Operations

It is known from Section 3.3 that to compute a DCT coefficient $F(u,v)$ requires 2^{2-h} multiplications where $h = \min(h_u, h_v)$, then

① The 4 DCT coefficients in the 1th~4th groups requires 4×1=4 multiplications.

② The 8 coefficients in the 5th, 6th, 9th and 10th groups requires 8×2=16 multiplications.

③ There are 4 DCT coefficients in the 13th group. After they are decomposed (See Appendix.B), they only requires 4 multiplications.

④ The 32 coefficients in the 7th, 8th, 11th, 12th, 14th and 15th groups need 32×4=128 multiplications.

⑤ The 4 subgroups in the 16th group require 4×6=24 multiplications.

Therefore, 4+16+4+128+24=176 multiplications are required to compute all the DCT coefficients.

In fact, only 166 multiplications are needed for computing all the DCT coefficients because of $\gamma(0) = \cos(\pi \times 0/16) = 1$, in addition, 10 shift operations are needed for the multiplier 1/8.

(2) Addition Operations

① 224 additions are required to compute matrix $\boldsymbol{D}$.

②As for the DCT coefficients in the 1th~12th groups，they are computed directly with Formula (9), then the 4 DCT coefficients in the 1^{th}~4^{th} groups do not require any additions;

the 8 coefficients in the 5^{th},6^{th} ,9^{th} and 10^{th} groups require 8×1=8 additions;

the 16 coefficients in the 7^{th},8^{th} ,11^{th} and 12^{th} groups require 16×3=48 additions.

③As for the DCT coefficients in the 13^{th}~16^{th} groups their basic images need to be decomposed during the process of computation. To compute all $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ requires 88 additions according to Section 3.3.2. Moreover, for the last results, each coefficient in the 13^{th} group requires 1 addition and each coefficient in the 14^{th} and 15^{th} groups requires 3 additions. All the coefficients in the 16^{th} group requires 4×8=32 additions.

Hence, total of 224+8+48+88+(4×1+16×3+32)=452 additions are required to compute all the DCT coefficients.

In summary, total of 166 multiplications and 452 additions are required to compute all the DCT coefficients.

4 The Scaled Algorithm

In order to reduce the multiplication number of the proposed algorithm further in image compression, we rewrite the expression of $F(1,2)$:

$$\begin{aligned} F(1,2) &= \overline{\boldsymbol{f}}_{12} \odot \overline{\boldsymbol{T}}_{12} \\ &= \frac{\gamma(1)}{8} \cdot [\boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(3)}{\gamma(1)} \cdot \boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(5)}{\gamma(1)} \cdot \boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(7)}{\gamma(1)} \cdot \boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C] \\ &= S(1,2) \times F_s(1,2) \end{aligned}$$

where $S(1,2) = \frac{\gamma(1)}{8}$ and

$$F_s(1,2) = \boldsymbol{P}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(3)}{\gamma(1)} \cdot \boldsymbol{Q}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(5)}{\gamma(1)} \cdot \boldsymbol{R}_3 \odot \boldsymbol{D}_{14}^C + \frac{\gamma(7)}{\gamma(1)} \cdot \boldsymbol{S}_3 \odot \boldsymbol{D}_{14}^C$$

It is noted that three multiplications are required to compute $F_s(1,2)$. The number of multiplication is reduced by one in comparison with the computation of $F(1,2)$. Of course, $S(1,2)$ can also be $\frac{\gamma(3)}{8}, \frac{\gamma(5)}{8}$ or $\frac{\gamma(7)}{8}$. We can do the same job for the other DCT coefficients as follows:

$$F(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \overline{\boldsymbol{T}}_{uv} = F_s(u,v) \times S(u,v)$$

$$F_s(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \frac{\overline{\boldsymbol{T}}_{uv}}{S(u,v)} \tag{13}$$

In image compression, we can compute $F_s(u,v)$ instead of $F(u,v)$ and the scaling factor $S(u,v)$ can be merged by a quantization process. As a result, 126 multiplications and 452 additions are needed for computing all the scaled DCT coefficients. Comparing with the algorithms shown in Table 1, the new scaled algorithm outperforms all except Feig's algorithm in terms of multiplications and additions. The new fast algorithm is suitable for pruning because each DCT coefficient is computed more independently in the new algorithm.

5 Pruning 2-D DCT and Comparative Results

The standard DCT algorithms inherently assume that the length of the input and output sequences are equal. However in many applications such as image compression, the most important information about a signal is contained by the low-frequency DCT components. Therefore, only the low-frequency components need to be computed in these applications. A method where only a subset of the output is computed to increase the computational efficiency is commonly referred to as "pruning".

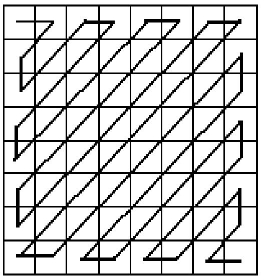


Fig.4 Pruning scheme of the zig-zag

5.1 Pruning Algorithm

With the above-mentioned algorithm in Sections 3 and 4, we compute each DCT coefficient in turn according to Fig.4. Computing each DCT coefficient requires the following two steps:

(1) To obtain $\overline{\boldsymbol{f}}_{uv}$

For the computational efficiency only some low frequency DCT coefficients are computed, it is not necessary to compute all the matrices $\boldsymbol{D}_i$ (i=1, 2,…, 16) at one time, and each $\boldsymbol{D}_i$ is computed when it is needed for a DCT coefficient. Every computed $\boldsymbol{D}_i$ is stored for computing higher frequency DCT coefficients in the same group.

(2) To compute $F_s(u,v)$

If $F_s(u,v)$ is in the 1th~12th groups, it is computed directly with the formula (13), for example:

$$F_s(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \frac{\overline{\boldsymbol{T}}_{uv}}{S(u,v)} \qquad \text{(here } (h_u, h_v) = (0, 2)\text{)}$$

$$= \overline{f}_{uv}(0,0) + \overline{f}_{uv}(1,0) \cdot \frac{\overline{T}_{uv}(1,0)}{\overline{T}_{uv}(0,0)} + \overline{f}_{uv}(2,0) \cdot \frac{\overline{T}_{uv}(2,0)}{\overline{T}_{uv}(0,0)} + \overline{f}_{uv}(3,0) \cdot \frac{\overline{T}_{uv}(3,0)}{\overline{T}_{uv}(0,0)}$$

If $F_s(u,v)$ is in the 13th~16th groups, it is computed according to the following steps:

① To compute $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ by using the method of section 3.3.2. If the four values are computed for the first time, they should be stored for the other coefficients in the same subgroup.

② If $F_s(u,v)$ is in the 14th or 15th group, let $S(u,v) = \frac{\gamma(k_1)}{8}$, then

$$F_s(u,v) = \boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv} - \frac{\gamma(k_2)}{\gamma(k_1)} \cdot \boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv} - \frac{\gamma(k_3)}{\gamma(k_1)} \cdot \boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv} - \frac{\gamma(k_4)}{\gamma(k_1)} \cdot \boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv};$$

If $F_s(u,v)$ is in the 13th or 16th group, let $S(u,v) = \frac{\gamma(0)}{8}$, then

$$F_s(u,v) = \boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv} - \gamma(k_2) \cdot \boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv} - \gamma(k_3) \cdot \boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv} - \gamma(k_4) \cdot \boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv} \qquad (14)$$

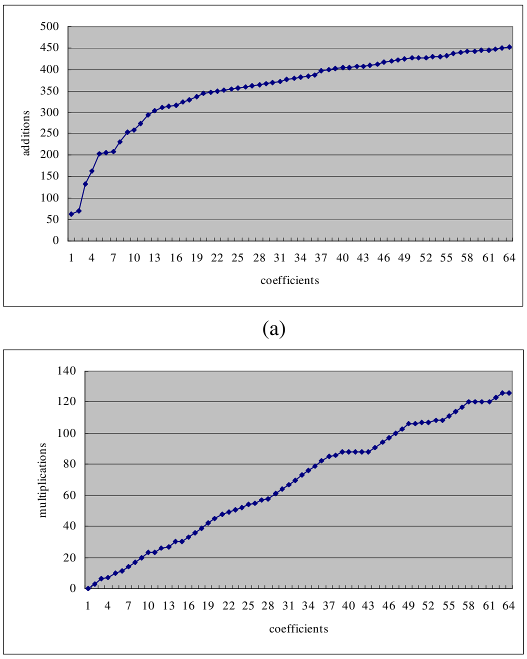
We compute Formula (14) by using the method of Section 3.3.3. According to Fig.4, we can compute any number of low-frequency DCT components conveniently. The computational complexity of computing the pruned version of the DCT depends on the number of desired DCT coefficients. Fig. 5 shows the evolution of the computational complexity according to the number of desired DCT coefficients.

5.2 Comparative Results

In this section, we will compare the performance of the proposed pruning algorithm to the algorithms described in [26 and 27]. The pruning algorithms in [26 and 27] differ from each other. The algorithm in [27] computes a set of coefficients included in a top-left triangle and it corresponds to a zig-zag scanning where all coefficients in each diagonal are computed. In [27], the pruning algorithm computes $N_0 \times N_0$ block coefficients in a top-left rectangle (N_0 is a power of 2) out of N×N.

Table 3 shows the number of required multiplications and additions of the proposed pruning algorithm compared with the DCT pruning algorithm of [26]. With respect to the pruning algorithm in [27], Table 4 lists the number of operations comparing to the proposed pruning algorithm.

As we can observe from Table 3 and Table 4, the proposed algorithm outperforms the existing algorithms in terms of arithmetic operations. On average, the proposed algorithm reduces 8.7% of multiplication and addition operations; if we only consider the number of multiplications, the algorithm reduces 58.7% of multiplications comparing with [26]. In comparison with [27], it achieves 14% reduction in the number of multiplication and addition operations, and 51.4% reduction in the number of multiplications operations.



(a)

(b)

Fig.5 Number of operations according to the desired number of zig-zag output coefficients (a) multiplications, (b) additions.

6 The Efficiency of Parallel Computation

In this section, we will explain why the new algorithm is very suitable for parallel computing architecture. From Sections 3 and 4, we can see that the algorithm needs two steps to compute the whole 2D 8*8 DCT.

1) Fast calculation of the matrix $\boldsymbol{D}$, which consists of all the matrices $\overline{\boldsymbol{f}}_{uv}$ (u, v=0, 1…, 7);

2) Computing all the DCT coefficients by using $F(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \overline{\boldsymbol{T}}_{\boldsymbol{uv}}$ (u, v=0, 1…, 7).

The parallel efficiency for each step will be discussed as follows.

6.1 Parallel computation of the matrix $\boldsymbol{D}$

In this step, first, the rows of $\boldsymbol{f}$ are accessed according to the flow of Fig.2 to get $\boldsymbol{Z}_i$, and then we repeat this process for the columns of $\boldsymbol{f}$, finally to produce the matrix $\boldsymbol{D}$ consisting of all the matrices $\overline{\boldsymbol{f}}_{uv}$ (u, v=0,1…,7). We can easily see that the computations of $\boldsymbol{Z}_i$ (i = 0,1…,7) are independent to each another, and therefore, can be done simultaneously. So the maximum degree of parallelism in this step is 64 for the 2D 8*8 DCT.

6.2 Parallel computation of all the DCT coefficients $F(u,v)$ (u, v=0, 1…, 7)

In this step, the computations of the DCT coefficients are divided into many groups. The first 12 groups have 28 DCT coefficients, which are computed by $F(u,v) = \overline{\boldsymbol{f}}_{uv} \odot \overline{\boldsymbol{T}}_{\boldsymbol{uv}}$. They can be done simultaneously. To compute the DCT coefficients in the 13th ~ 16th groups, we need to compute the values of $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ first. In parallel implementation, the values of $\boldsymbol{P}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{Q}_i \odot \overline{\boldsymbol{f}}_{uv}$, $\boldsymbol{R}_i \odot \overline{\boldsymbol{f}}_{uv}$ and $\boldsymbol{S}_i \odot \overline{\boldsymbol{f}}_{uv}$ can be pre-computed, and then broadcasted to the 13th ~ 16th groups. Thus, the 16 coefficients in the 14th and 15th groups, and the 6 subgroups in the 13th and 16th groups can be computed at the same time. So the maximum degree of parallelism in this step is 50 (=28+16+6).

Based on the above discussion, we can easily see that this algorithm can be easily implemented on any parallel computer or a multi-core CPU machine with a high degree of parallelism.

Table 3 COMPARATIVE RESULTS BETWEEN [26] AND THE PROPOSED ONE

Number of low frequency DCT coefficients	Multiplications / additions	
	[26]	Proposed pruning algorithm
1	0 / 72	0 / 63
3	36 / 143	6 / 133
6	58 / 176	11 / 206
10	82 / 229	23 / 258
15	99 / 250	30 / 313
21	117 / 289	48 / 347
28	136 / 312	58 / 364
36	156 / 353	82 / 388

Table 4 COMPARATIVE RESULTS BETWEEN [27] AND THE PROPOSED ONE

Number of low frequency DCT coefficients	Multiplications / additions	
	[27]	Proposed pruning algorithm
2×2	48 / 188	9 / 172
4×4	96 / 312	38 / 314
8×8	144 / 464	126 / 452

7 Conclusions

In the application of image/video coding, statistic shows that only 6~7 coefficients of a 8×8 DCT block are nonzero (7~16 coefficients for special cases), especially the chromatic difference U/V signals have much less nonzero coefficients. Generally, DCT concentrates the image energy to the top-left corner, and most zeros lie at the bottom-right corner. Especially, in low bit-rate image coding, only a small proportion of the DCT coefficients located at the top-left corner are computed. The existing well-known fast DCT algorithms focus mainly on making the operation number of the whole 2D 8×8 DCT coefficients as low as possible, but each DCT coefficient cannot be independently computed conveniently. If all the DCT coefficients are computed, the new fast algorithm is not better than some existing algorithms, such as the algorithm introduced by Feig and Winograd[3]. Feig and Winograd exploited the inherent structures in the DCT deeply to reduce the computation cost of computing the whole 2D 8×8 DCT. However, it is inconvenient to compute each DCT coefficient independently with Feig's algorithm. As a result, it is complicated and not efficient to get pruned DCT with Feig's algorithm. In the new algorithm, each DCT coefficient is computed more independently and quickly by using the formula $F(u,v) = \overline{\boldsymbol{f}} \odot \overline{\boldsymbol{T}}_{uv}$. As such, the new algorithm is suitable for pruning and parallel computing. In order to decrease arithmetic operations greatly, we can compute only the nonzero low-frequency DCT coefficients for coding and transmission, and the speed of DCT will be further increased with the new algorithm. The efficiency and rationality of the new algorithm are illustrated in the above analysis.

1 Pennebaker W B and Mitchell J L. JPEG Still Image Data Compression Standard. New York: Van Nostrand Reinhold, 1993

2 Mitchell J L, Pennebaker W B, Fogg C E, et al. MPEG Video Compression Standard. New York: Chapman and Hall, 1997

3 Feig E and Winograd S. Fast algorithms for the discrete cosine transform. IEEE Trans. on Signal Processing, 1992, 40(9): 2174–2193

4 Leoffler C, Ligtenberg A, Moschytz G S. Practical fast 1D DCT algorithms with 11 mutiplications. In Proc. IEEE ICASSP, 1989, 2: 988–991

5 Lee B G. A new algorithm to compute the discrete cosine transform. IEEE Trans. on Acoustic, Speech, Signal Processing, 1984, 32(12): 1243–1245

6 Chen W H, Smith C H, and Fralick S C. A fast computational algorithm for the discrete cosine transform. IEEE Trans. on Communications, 1977, 25(9): 1004–1009

7 Ahmed N, Natarajan T, Rao K R. Discrete Cosine Transform", IEEE. Trans. on Computer, 1974,23(1): 90–93

8 Vetterli M, Nussbaumer H. Simple FFT and DCT Algorithms with Reduced Number of Operations. Signal Processing, 1984, 6: 267–278

9 Suehiro N, Hateri M. Fast Algorithms for the DFT and other sinusoidal transforms. IEEE Trans. on Acoustic, Signal, and Speech Processing, 1986, 34(6): 642– 644

10 Arai Y, Agui T, Nakajima M. A Fast DCT-SQ Scheme for Images. Trans. of the IEICE, 1988,71(11): 1095–1097

11 Hou H S. A fast recursive algorithm for computing the discrete cosine transform. IEEE Trans. on Acoustic, Speech, Signal Processing, 1987, 35(10): 1445–1461

12 Duhamel P, Guillemot C. Polynomial transform computation of 2-D DCT. In Proc. ICASSP, 1990: 1515–1518

13 Kamanagar F A, Rao K R. Fast Algorithms for the 2-D Discrete Cosine Transform. IEEE Trans. on Computers, 1982, 31(9): 899–906

14 Huang Y M, Wu J L. A Refined Fast 2-D Discrete Cosine Transform Algorithm. IEEE Trans. on Signal Processing, 1999, 47(3): 904–907

15 Skodras A N. Direct transform to computation. IEEE Signal Processing Lett., 1999, 6: 202–204

16 Liang J. Fast Multiplierless Approximations of the DCT With the Lifting Scheme. IEEE Trans. on Signal Processing , 2001, 49(12): 3032–3044

17 Cho N I, Lee S U. A fast 4x4 algorithm for the recursive 2-D DCT. IEEE Trans. on Signal Processing, 1992, 40(9): 2166–2173

18 Cho N I, Lee S U. Fast algorithm and implementation of 2-D discrete cosine transform. IEEE Trans. on Circuits Syst., 1991, 38: 297–305

19 Chan S C, Ho K L. A new two-dimensional fast cosine transform. IEEE Trans. on Signal Processing, 1991, 39: 481–485

20 Ma C. A fast recursive two-dimensional cosine transform. Proc.SPIEInt. Soc. Opt. Eng., 1988: 541–548

21 Vetterli M. Fast 2-D discrete cosine transform. In Proc. ICASSP, 1985: 1538–1541

22 Duhamel P, Guillemot C. Polynomial transform computation of 2-D DCT. In Proc. ICASSP,1990: 1515–1518

23 Skodras A N. Fast discrete cosine transform pruning. IEEE Trans. on Signal Processing, 1994, 42(7): 1833–1837

24 Wang Z. Pruning the fast discrete cosine Transform. IEEE Trans. on Communication, 1991, 39(5): 640–643

25 Huang Y, Wu J, Chang C. A generalized output pruning algorithm for matrix-vector multiplication and its application to computing pruning discrete cosine transform. IEEE Trans. on Signal Processing, 2000, 48(2): 561–563

26 Walmsley N P, Skodras A N, Curtis K M. A fast picture compression technique. IEEE Trans. on Consumer Elect., 1994, 40(1): 11–19

27 Christopoulos C A, Skodras A N. Pruning the two-dimensional fast cosine transform. Proceedings of the VII European Signal Processing Conference (EUSIPCO), Edinburgh, Scotland, UK, 1994: 596–599

28 Silva A, Navarro A. Fast 8x8 DCT Pruning Algorithm. IEEE International Conference on Image Processing, 2005, 2: 317–20

29 Ji X H, Zhang C M. A Fast IDCT Algorithm for Image Decompression. Chinese journal of Computer, 2005, 28(12): 2079–2088

30 Ji X H. Fast DCT Algorithm for Low Bit-Rate Image compression. Chinese Journal of Computer-aided design & Computer graphics, 2004, 16(10): 1355–1359

适于图像压缩的二维 8*8 离散余弦正变换快速算法

纪秀花[1,2] 张彩明[1,2] 汪嘉业[1] 王铠[1]

[1] 山东经济学院计算机科学与技术学院 济南 250014

[2] 山东大学计算机科学与技术学院 济南 250061

?

摘要：基于对二维离散余弦变换（DCT）的基本图像特点的研究，提出一种二维8*8离散余弦正变换的快速算法。新算法快速独立地依次计算每一个DCT系数，所以该算法很适合做裁掉（Pruning）任意数目DCT高频成分的二维DCT Pruning算法。就运算复杂性而言，本文提出的Pruning算法比已有的其它二维DCT Pruning算法更有效，尤其在所需的乘法运算量方面。

关键词：离散余弦正变换（DCT）；基本图像；量化处理；图像编码

Appendix.A

According to h_u and h_v, the 64 basic images of 2D 8×8 DCT are classified into 16 groups and the basic images in each group have the similar symmetry (See Fig.A1~A20). Fig.A1~A20 are explained as follows:

(1) The elements (*A*,*B*,*C* and *D*) of the matrices in Fig.A1~A20 are used to explain the symmetries. *A*,*B*,*C* and *D* in different matrix of Fig.A1~A20 do not mean they have the same values.

(2) The sub-matrices in the dashed rectangles of Fig.A1~A20 are the smallest units.

(3) It is known by observing that the basic images in the 13th and 16th groups can be classified into several kinds of symmetries.

$$\begin{bmatrix} \boxed{A} & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \end{bmatrix}$$

Fig.A1 Symmetry of the 1st group $\boldsymbol{T}_{00}$

$$\begin{bmatrix} \boxed{A} & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \end{bmatrix}$$

Fig.A2 Symmetry of the 2nd group $\boldsymbol{T}_{04}$

$$\begin{bmatrix} \boxed{A} & A & A & A & A & A & A & A \\ -A & -A & -A & -A & -A & -A & -A & -A \\ -A & -A & -A & -A & -A & -A & -A & -A \\ A & A & A & A & A & A & A & A \\ A & A & A & A & A & A & A & A \\ -A & -A & -A & -A & -A & -A & -A & -A \\ -A & -A & -A & -A & -A & -A & -A & -A \\ A & A & A & A & A & A & A & A \end{bmatrix}$$

Fig.A3 Symmetry of the 3nd group $\boldsymbol{T}_{40}$

$$\begin{bmatrix} \boxed{A} & -A & -A & A & A & -A & -A & A \\ -A & A & A & -A & -A & A & A & -A \\ -A & A & A & -A & -A & A & A & -A \\ A & -A & -A & A & A & -A & -A & A \\ A & -A & -A & A & A & -A & -A & A \\ -A & A & A & -A & -A & A & A & -A \\ -A & A & A & -A & -A & A & A & -A \\ A & -A & -A & A & A & -A & -A & A \end{bmatrix}$$

Fig.A4 Symmetry of the 4th group $\boldsymbol{T}_{44}$

$$\begin{bmatrix} \boxed{A} & \boxed{B} & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \end{bmatrix}$$

Fig.A5 Symmetry of the 5th group ($\boldsymbol{T}_{02}$, $\boldsymbol{T}_{06}$)

$$\begin{bmatrix} \boxed{A} & A & A & A & A & A & A & A \\ \boxed{B} & B & B & B & B & B & B & B \\ -B & -B & -B & -B & -B & -B & -B & -B \\ -A & -A & -A & -A & -A & -A & -A & -A \\ -A & -A & -A & -A & -A & -A & -A & -A \\ -B & -B & -B & -B & -B & -B & -B & -B \\ B & B & B & B & B & B & B & B \\ A & A & A & A & A & A & A & A \end{bmatrix}$$

Fig.A6 Symmetry of the 6th group $\boldsymbol{T}_{20}$ and $\boldsymbol{T}_{60}$

$$\begin{bmatrix} \boxed{A} & \boxed{B} & \boxed{C} & \boxed{D} & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \end{bmatrix}$$

Fig.A7 Symmetry of the 7th group ($\boldsymbol{T}_{01}$, $\boldsymbol{T}_{03}$, $\boldsymbol{T}_{05}$, $\boldsymbol{T}_{07}$)

$$\begin{bmatrix} \boxed{A} & A & A & A & A & A & A & A \\ \boxed{B} & B & B & B & B & B & B & B \\ \boxed{C} & C & C & C & C & C & C & C \\ \boxed{D} & D & D & D & D & D & D & D \\ -D & -D & -D & -D & -D & -D & -D & -D \\ -C & -C & -C & -C & -C & -C & -C & -C \\ -B & -B & -B & -B & -B & -B & -B & -B \\ -A & -A & -A & -A & -A & -A & -A & -A \end{bmatrix}$$

Fig.A8 Symmetry of the 8th group ($\boldsymbol{T}_{10}$, $\boldsymbol{T}_{30}$, $\boldsymbol{T}_{50}$, $\boldsymbol{T}_{70}$)

$$\begin{bmatrix} \boxed{A} & \boxed{B} & -B & -A & -A & -B & B & A \\ -A & -B & B & A & A & B & -B & -A \\ -A & -B & B & A & A & B & -B & -A \\ A & B & -B & -A & -A & -B & B & A \\ A & B & -B & -A & -A & -B & B & A \\ -A & -B & B & A & A & B & -B & -A \\ -A & -B & B & A & A & B & -B & -A \\ A & B & -B & -A & -A & -B & B & A \end{bmatrix}$$

Fig.A9 Symmetry of the 9th group ($\boldsymbol{T}_{42}$, $\boldsymbol{T}_{46}$)

$$\begin{bmatrix} A & -A & -A & A & A & -A & -A & A \\ B & -B & -B & B & B & -B & -B & B \\ -B & B & B & -B & -B & B & B & -B \\ -A & A & A & -A & -A & A & A & -A \\ -A & A & A & -A & -A & A & A & -A \\ -B & B & B & -B & -B & B & B & -B \\ B & -B & -B & B & B & -B & -B & B \\ A & -A & -A & A & A & -A & -A & A \end{bmatrix}$$

Fig.A10 Symmetry of the 10th group ($\boldsymbol{T}_{24}$ 、 $\boldsymbol{T}_{64}$)

$$\begin{bmatrix} A & B & C & D & -D & -C & -B & -A \\ -A & -B & -C & -D & D & C & B & A \\ -A & -B & -C & -D & D & C & B & A \\ A & B & C & D & -D & -C & -B & -A \\ A & B & C & D & -D & -C & -B & -A \\ -A & -B & -C & -D & D & C & B & A \\ -A & -B & -C & -D & D & C & B & A \\ A & B & C & D & -D & -C & -B & -A \end{bmatrix}$$

Fig.A11 Symmetry of the 11th group ($\boldsymbol{T}_{41}$, $\boldsymbol{T}_{43}$, $\boldsymbol{T}_{45}$, $\boldsymbol{T}_{47}$)

$$\begin{bmatrix} A & -A & -A & A & A & -A & -A & A \\ B & -B & -B & B & B & -B & -B & B \\ C & -C & -C & C & C & -C & -C & C \\ D & -D & -D & D & D & -D & -D & D \\ -D & D & D & -D & -D & D & D & -D \\ -C & C & C & -C & -C & C & C & -C \\ -B & B & B & -B & -B & B & B & -B \\ -A & A & A & -A & -A & A & A & -A \end{bmatrix}$$

Fig.A12 Symmetry of the 12th group ($\boldsymbol{T}_{14}$, $\boldsymbol{T}_{34}$, $\boldsymbol{T}_{54}$, $\boldsymbol{T}_{74}$)

$$\begin{bmatrix} A & B & -B & -A & -A & -B & B & A \\ B & C & -C & -B & -B & -C & C & B \\ -B & -C & C & B & B & C & -C & -B \\ -A & -B & B & A & A & B & -B & -A \\ -A & -B & B & A & A & B & -B & -A \\ -B & -C & C & B & B & C & -C & -B \\ B & C & -C & -B & -B & -C & C & B \\ A & B & -B & -A & -A & -B & B & A \end{bmatrix}$$

Fig.A13 Symmetry of the 13th group ($\boldsymbol{T}_{22}$, $\boldsymbol{T}_{66}$)

$$\begin{bmatrix} B & A & -A & -B & -B & -A & A & B \\ C & -B & B & -C & -C & B & -B & C \\ -C & B & -B & C & C & -B & B & -C \\ -B & -A & A & B & B & A & -A & -B \\ -B & -A & A & B & B & A & -A & -B \\ -C & B & -B & C & C & -B & B & -C \\ C & -B & B & -C & -C & B & -B & C \\ B & A & -A & -B & -B & -A & A & B \end{bmatrix}$$

Fig.A14 Symmetry of the 13th group ($\boldsymbol{T}_{26}$, $\boldsymbol{T}_{62}$)

$$\begin{bmatrix} A & E & -E & -A & -A & -E & E & A \\ B & F & -F & -B & -B & -F & F & B \\ C & G & -G & -C & -C & -G & G & C \\ D & H & -H & -D & -D & -H & H & D \\ -D & -H & H & D & D & H & -H & -D \\ -C & -G & G & C & C & G & -G & -C \\ -B & -F & F & B & B & F & -F & -B \\ -A & -E & E & A & A & E & -E & -A \end{bmatrix}$$

Fig.A15 Symmetry of the 14th group ($\boldsymbol{T}_{12}$, $\boldsymbol{T}_{32}$, $\boldsymbol{T}_{52}$, $\boldsymbol{T}_{72}$, $\boldsymbol{T}_{16}$, $\boldsymbol{T}_{36}$, $\boldsymbol{T}_{56}$, $\boldsymbol{T}_{76}$)

$$\begin{bmatrix} A & B & C & D & -D & -C & -B & -A \\ E & F & G & H & -H & -G & -F & -E \\ -E & -F & -G & -H & H & G & F & E \\ -A & -B & -C & -D & D & C & B & A \\ -A & -B & -C & -D & D & C & B & A \\ -E & -F & -G & -H & H & G & F & E \\ E & F & G & H & -H & -G & -F & -E \\ A & B & C & D & -D & -C & -B & -A \end{bmatrix}$$

Fig.A16 Symmetry of the 15th group ($\boldsymbol{T}_{21}$, $\boldsymbol{T}_{23}$, $\boldsymbol{T}_{25}$, $\boldsymbol{T}_{27}$, $\boldsymbol{T}_{61}$, $\boldsymbol{T}_{63}$, $\boldsymbol{T}_{65}$, $\boldsymbol{T}_{67}$)

$$\begin{bmatrix} A & B & C & D & -D & -C & -B & -A \\ B & E & F & G & -G & -F & -E & -B \\ C & F & H & I & -I & -H & -F & -C \\ D & G & I & J & -J & -I & -G & -D \\ -D & -G & -I & -J & J & I & G & D \\ -C & -F & -H & -I & I & H & F & C \\ -B & -E & -F & -G & G & F & E & B \\ -A & -B & -C & -D & D & C & B & A \end{bmatrix}$$

Fig.A17 Symmetry of the 16th group ($\boldsymbol{T}_{11}$, $\boldsymbol{T}_{33}$, $\boldsymbol{T}_{55}$, $\boldsymbol{T}_{77}$)

$$\begin{bmatrix} A & B & C & D & -D & -C & -B & -A \\ E & -D & J & F & -F & -J & D & -E \\ G & -A & H & E & -E & -H & A & -G \\ H & -C & I & J & -J & -I & C & -H \\ -H & C & -I & -J & J & I & -C & H \\ -G & A & -H & -E & E & H & -A & G \\ -E & D & -J & -F & F & J & -D & E \\ -A & -B & -C & -D & D & C & B & A \end{bmatrix}$$

Fig.A18 Symmetry of the 16th group ($\boldsymbol{T}_{15}$, $\boldsymbol{T}_{31}$, $\boldsymbol{T}_{57}$, $\boldsymbol{T}_{73}$)

Appendix.B

The basic images in the 13th~16th groups can be decomposed as follows ($\gamma(k)=\cos(\pi k/16)$ ）

$$\bar{\boldsymbol{T}}_{22}=\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_1+\frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_1,\quad \bar{\boldsymbol{T}}_{66}=\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_1-\frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_1$$

$$\bar{\boldsymbol{T}}_{26}=\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_2+\frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_2,\quad \bar{\boldsymbol{T}}_{62}=-\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_2+\frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_2$$

where

$$\boldsymbol{P}_1=\begin{bmatrix}1&0\\0&1\end{bmatrix},\ \boldsymbol{Q}_1=\begin{bmatrix}1&1\\1&-1\end{bmatrix},\boldsymbol{P}_2=\begin{bmatrix}0&-1\\1&0\end{bmatrix},\boldsymbol{Q}_2=\begin{bmatrix}1&-1\\-1&-1\end{bmatrix}$$

$$\bar{\boldsymbol{T}}_{12}=\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_3+\frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_3+\frac{\gamma(5)}{8}\cdot\boldsymbol{R}_3+\frac{\gamma(7)}{8}\cdot\boldsymbol{S}_3$$

$$\bar{\boldsymbol{T}}_{36}=\frac{\gamma(3)}{8}\cdot\boldsymbol{P}_3-\frac{\gamma(7)}{8}\cdot\boldsymbol{Q}_3-\frac{\gamma(1)}{8}\cdot\boldsymbol{R}_3-\frac{\gamma(5)}{8}\cdot\boldsymbol{S}_3$$

$$\bar{\boldsymbol{T}}_{56}=-\frac{\gamma(5)}{8}\cdot\boldsymbol{P}_3+\frac{\gamma(1)}{8}\cdot\boldsymbol{Q}_3-\frac{\gamma(7)}{8}\cdot\boldsymbol{R}_3-\frac{\gamma(3)}{8}\cdot\boldsymbol{S}_3$$

$$\bar{\boldsymbol{T}}_{72}=-\frac{\gamma(7)}{8}\cdot\boldsymbol{P}_3+\frac{\gamma(5)}{8}\cdot\boldsymbol{Q}_3-\frac{\gamma(3)}{8}\cdot\boldsymbol{R}_3+\frac{\gamma(1)}{8}\cdot\boldsymbol{S}_3$$

where

$$\boldsymbol{P_3}=\begin{bmatrix}1&0\\1&0\\0&1\\0&1\end{bmatrix},\ \boldsymbol{Q_3}=\begin{bmatrix}1&0\\0&1\\1&0\\0&-1\end{bmatrix},\ \boldsymbol{R_3}=\begin{bmatrix}0&1\\1&0\\0&-1\\1&0\end{bmatrix},\ \boldsymbol{S}_3=\begin{bmatrix}0&1\\0&-1\\1&0\\-1&0\end{bmatrix}$$

$$\bar{\boldsymbol{T}}_{32}=\frac{\gamma(7)}{8}\cdot\boldsymbol{P}_4+\frac{\gamma(5)}{8}\cdot\boldsymbol{Q}_4-\frac{\gamma(3)}{8}\cdot\boldsymbol{R}_4+\frac{\gamma(1)}{8}\cdot\boldsymbol{S}_4$$

$$\bar{\boldsymbol{T}}_{16}=\frac{\gamma(3)}{8}\cdot\boldsymbol{P}_4+\frac{\gamma(7)}{8}\cdot\boldsymbol{Q}_4+\frac{\gamma(1)}{8}\cdot\boldsymbol{R}_4+\frac{\gamma(5)}{8}\cdot\boldsymbol{S}_4$$

$$\bar{\boldsymbol{T}}_{52}=-\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_4+\frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_4+\frac{\gamma(5)}{8}\cdot\boldsymbol{R}_4+\frac{\gamma(7)}{8}\cdot\boldsymbol{S}_4$$

$$\bar{\boldsymbol{T}}_{76}=\frac{\gamma(5)}{8}\cdot\boldsymbol{P}_4+\frac{\gamma(1)}{8}\cdot\boldsymbol{Q}_4-\frac{\gamma(7)}{8}\cdot\boldsymbol{R}_4-\frac{\gamma(3)}{8}\cdot\boldsymbol{S}_4$$

where

$$\boldsymbol{P}_4=\begin{bmatrix}0&-1\\1&0\\0&-1\\-1&0\end{bmatrix},\boldsymbol{Q}_4=\begin{bmatrix}1&0\\-1&0\\0&-1\\0&1\end{bmatrix},\boldsymbol{R}_4=\begin{bmatrix}0&-1\\0&-1\\1&0\\1&0\end{bmatrix},\boldsymbol{S}_4=\begin{bmatrix}1&0\\0&-1\\-1&0\\0&-1\end{bmatrix}$$

$$\bar{\boldsymbol{T}}_{21}=\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_5+\frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_5+\frac{\gamma(5)}{8}\cdot\boldsymbol{R}_5+\frac{\gamma(7)}{8}\cdot\boldsymbol{S}_5$$

$$\bar{\boldsymbol{T}}_{63}=\frac{\gamma(3)}{8}\cdot\boldsymbol{P}_5-\frac{\gamma(7)}{8}\cdot\boldsymbol{Q}_5-\frac{\gamma(1)}{8}\cdot\boldsymbol{R}_5-\frac{\gamma(5)}{8}\cdot\boldsymbol{S}_5$$

$$\bar{\boldsymbol{T}}_{65}=-\frac{\gamma(5)}{8}\cdot\boldsymbol{P}_5+\frac{\gamma(1)}{8}\cdot\boldsymbol{Q}_5-\frac{\gamma(7)}{8}\cdot\boldsymbol{R}_5-\frac{\gamma(3)}{8}\cdot\boldsymbol{S}_5$$

$$\overline{\boldsymbol{T}}_{27} = -\frac{\gamma(7)}{8}\cdot\boldsymbol{P}_5 + \frac{\gamma(5)}{8}\cdot\boldsymbol{Q}_5 - \frac{\gamma(3)}{8}\cdot\boldsymbol{R}_5 + \frac{\gamma(1)}{8}\cdot\boldsymbol{S}_5$$

where

$$\boldsymbol{P}_5 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 \end{bmatrix}, \boldsymbol{Q}_5 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}, \boldsymbol{R}_5 = \begin{bmatrix} 0 & 1 & 0 & 1 \\ 1 & 0 & -1 & 0 \end{bmatrix}, \boldsymbol{S}_5 = \begin{bmatrix} 0 & 0 & 1 & -1 \\ 1 & -1 & 0 & 0 \end{bmatrix}$$

$$\overline{\boldsymbol{T}}_{23} = \frac{\gamma(7)}{8}\cdot\boldsymbol{P}_6 + \frac{\gamma(5)}{8}\cdot\boldsymbol{Q}_6 - \frac{\gamma(3)}{8}\cdot\boldsymbol{R}_6 + \frac{\gamma(1)}{8}\cdot\boldsymbol{S}_6$$

$$\overline{\boldsymbol{T}}_{61} = \frac{\gamma(3)}{8}\cdot\boldsymbol{P}_6 + \frac{\gamma(7)}{8}\cdot\boldsymbol{Q}_6 + \frac{\gamma(1)}{8}\cdot\boldsymbol{R}_6 + \frac{\gamma(5)}{8}\cdot\boldsymbol{S}_6$$

$$\overline{\boldsymbol{T}}_{25} = -\frac{\gamma(1)}{8}\cdot\boldsymbol{P}_6 + \frac{\gamma(3)}{8}\cdot\boldsymbol{Q}_6 + \frac{\gamma(5)}{8}\cdot\boldsymbol{R}_6 + \frac{\gamma(7)}{8}\cdot\boldsymbol{S}_6$$

$$\overline{\boldsymbol{T}}_{67} = -\frac{\gamma(5)}{8}\cdot\boldsymbol{P}_6 + \frac{\gamma(1)}{8}\cdot\boldsymbol{Q}_6 - \frac{\gamma(7)}{8}\cdot\boldsymbol{R}_6 - \frac{\gamma(3)}{8}\cdot\boldsymbol{S}_6$$

where

$$\boldsymbol{A}_6 = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 0 & -1 & 0 \end{bmatrix}, \boldsymbol{B}_6 = \begin{bmatrix} 1 & -1 & 0 & 0 \\ 0 & 0 & -1 & 1 \end{bmatrix}, \boldsymbol{C}_6 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ -1 & -1 & 0 & 0 \end{bmatrix}, \boldsymbol{D}_6 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & -1 & 0 & -1 \end{bmatrix}$$

$$\overline{\boldsymbol{T}}_{11} = \frac{\gamma(0)}{8}\cdot\boldsymbol{P}_7 + \frac{\gamma(6)}{8}\cdot\boldsymbol{Q}_7 + \frac{\gamma(4)}{8}\cdot\boldsymbol{R}_7 + \frac{\gamma(2)}{8}\cdot\boldsymbol{S}_7$$

$$\overline{\boldsymbol{T}}_{33} = \frac{\gamma(0)}{8}\cdot\boldsymbol{P}_7 - \frac{\gamma(2)}{8}\cdot\boldsymbol{Q}_7 - \frac{\gamma(4)}{8}\cdot\boldsymbol{R}_7 + \frac{\gamma(6)}{8}\cdot\boldsymbol{S}_7$$

$$\overline{\boldsymbol{T}}_{55} = \frac{\gamma(0)}{8}\cdot\boldsymbol{P}_7 + \frac{\gamma(2)}{8}\cdot\boldsymbol{Q}_7 - \frac{\gamma(4)}{8}\cdot\boldsymbol{R}_7 - \frac{\gamma(6)}{8}\cdot\boldsymbol{S}_7$$

$$\overline{\boldsymbol{T}}_{77} = \frac{\gamma(0)}{8}\cdot\boldsymbol{P}_7 - \frac{\gamma(6)}{8}\cdot\boldsymbol{Q}_7 + \frac{\gamma(4)}{8}\cdot\boldsymbol{R}_7 - \frac{\gamma(2)}{8}\cdot\boldsymbol{S}_7$$

where

$$\boldsymbol{P}_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{Q}_7 = \begin{bmatrix} 0 & 0 & 1 & 1 \\ 0 & 1 & 0 & -1 \\ 1 & 0 & -1 & 0 \\ 1 & -1 & 0 & 0 \end{bmatrix}, \boldsymbol{R}_7 = \begin{bmatrix} 0 & 1 & 1 & 0 \\ 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & -1 \\ 0 & 1 & -1 & 0 \end{bmatrix}, \boldsymbol{S}_7 = \begin{bmatrix} 1 & 1 & 0 & 0 \\ 1 & 0 & 1 & 0 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & -1 \end{bmatrix}$$

$$\overline{\boldsymbol{T}}_{13} = \frac{\gamma(0)}{8}\cdot\boldsymbol{P}_8 + \frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_8 + \frac{\gamma(6)}{8}\cdot\boldsymbol{R}_8 + \frac{\gamma(2)}{8}\cdot\boldsymbol{S}_8$$

$$\overline{\boldsymbol{T}}_{51} = -\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_8 + \frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_8 - \frac{\gamma(2)}{8}\cdot\boldsymbol{R}_8 + \frac{\gamma(6)}{8}\cdot\boldsymbol{S}_8$$

$$\overline{\boldsymbol{T}}_{37} = -\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_8 + \frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_8 + \frac{\gamma(2)}{8}\cdot\boldsymbol{R}_8 - \frac{\gamma(6)}{8}\cdot\boldsymbol{S}_8$$

$$\overline{\boldsymbol{T}}_{75} = -\frac{\gamma(0)}{8}\cdot\boldsymbol{P}_8 - \frac{\gamma(4)}{8}\cdot\boldsymbol{Q}_8 + \frac{\gamma(6)}{8}\cdot\boldsymbol{R}_8 + \frac{\gamma(2)}{8}\cdot\boldsymbol{S}_8$$

where

$$\boldsymbol{P}_8 = \begin{bmatrix} 0 & 0 & -1 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ 0 & -1 & 0 & 0 \end{bmatrix}, \boldsymbol{Q}_8 = \begin{bmatrix} 1 & 0 & 0 & -1 \\ 0 & -1 & -1 & 0 \\ 0 & 1 & -1 & 0 \\ 1 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{R}_8 = \begin{bmatrix} 0 & -1 & 0 & -1 \\ 1 & 1 & 0 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & 0 & -1 & 0 \end{bmatrix}, \boldsymbol{S}_8 = \begin{bmatrix} 1 & 0 & -1 & 0 \\ 0 & 0 & -1 & -1 \\ 1 & -1 & 0 & 0 \\ 0 & 1 & 0 & -1 \end{bmatrix}$$

$$\overline{\boldsymbol{T}}_{31} = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_9 + \frac{\gamma(2)}{8} \cdot \boldsymbol{Q}_9 + \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_9 + \frac{\gamma(6)}{8} \cdot \boldsymbol{S}_9$$

$$\overline{\boldsymbol{T}}_{15} = -\frac{\gamma(0)}{8} \cdot \boldsymbol{P}_9 + \frac{\gamma(6)}{8} \cdot \boldsymbol{Q}_9 + \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_9 - \frac{\gamma(2)}{8} \cdot \boldsymbol{S}_9$$

$$\overline{\boldsymbol{T}}_{57} = -\frac{\gamma(0)}{8} \cdot \boldsymbol{P}_9 + \frac{\gamma(2)}{8} \cdot \boldsymbol{Q}_9 - \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_9 + \frac{\gamma(6)}{8} \cdot \boldsymbol{S}_9$$

$$\overline{\boldsymbol{T}}_{73} = -\frac{\gamma(0)}{8} \cdot \boldsymbol{P}_9 - \frac{\gamma(6)}{8} \cdot \boldsymbol{Q}_9 + \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_9 + \frac{\gamma(2)}{8} \cdot \boldsymbol{S}_9$$

where

$$\boldsymbol{P}_9 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -1 \\ -1 & 0 & 0 & 0 \\ 0 & 0 & -1 & 0 \end{bmatrix}, \boldsymbol{Q}_9 = \begin{bmatrix} 1 & 0 & 1 & 0 \\ 0 & 0 & -1 & 1 \\ -1 & -1 & 0 & 0 \\ 0 & -1 & 0 & -1 \end{bmatrix}, \boldsymbol{R}_9 = \begin{bmatrix} 1 & 0 & 0 & 1 \\ 0 & -1 & 1 & 0 \\ 0 & -1 & -1 & 0 \\ -1 & 0 & 0 & 1 \end{bmatrix}, \boldsymbol{S}_9 = \begin{bmatrix} 0 & 1 & 0 & -1 \\ -1 & 1 & 0 & 0 \\ 0 & 0 & -1 & -1 \\ -1 & 0 & 1 & 0 \end{bmatrix}$$

$$\overline{\boldsymbol{T}}_{35} = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_{10} + \frac{\gamma(2)}{8} \cdot \boldsymbol{Q}_{10} + \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_{10} + \frac{\gamma(6)}{8} \cdot \boldsymbol{S}_{10}$$

$$\overline{\boldsymbol{T}}_{17} = -\frac{\gamma(0)}{8} \cdot \boldsymbol{P}_{10} + \frac{\gamma(6)}{8} \cdot \boldsymbol{Q}_{10} + \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_{10} - \frac{\gamma(2)}{8} \cdot \boldsymbol{S}_{10}$$

$$\overline{\boldsymbol{T}}_{53} = -\frac{\gamma(0)}{8} \cdot \boldsymbol{P}_{10} + \frac{\gamma(2)}{8} \cdot \boldsymbol{Q}_{10} - \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_{10} + \frac{\gamma(6)}{8} \cdot \boldsymbol{S}_{10}$$

$$\overline{\boldsymbol{T}}_{71} = \frac{\gamma(0)}{8} \cdot \boldsymbol{P}_{10} + \frac{\gamma(6)}{8} \cdot \boldsymbol{Q}_{10} - \frac{\gamma(4)}{8} \cdot \boldsymbol{R}_{10} - \frac{\gamma(2)}{8} \cdot \boldsymbol{S}_{10}$$

where

$$\boldsymbol{P}_{10} = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 0 & 0 & -1 & 0 \\ 0 & 1 & 0 & 0 \\ -1 & 0 & 0 & 0 \end{bmatrix}, \boldsymbol{Q}_{10} = \begin{bmatrix} 1 & -1 & 0 & 0 \\ -1 & 0 & 1 & 0 \\ 0 & 1 & 0 & -1 \\ 0 & 0 & -1 & -1 \end{bmatrix}, \boldsymbol{R}_{10} = \begin{bmatrix} 0 & -1 & 1 & 0 \\ 1 & 0 & 0 & -1 \\ -1 & 0 & 0 & -1 \\ 0 & 1 & 1 & 0 \end{bmatrix}, \boldsymbol{S}_{10} = \begin{bmatrix} 0 & 0 & -1 & 1 \\ 0 & 1 & 0 & 1 \\ -1 & 0 & -1 & 0 \\ 1 & 1 & 0 & 0 \end{bmatrix}$$

?

Appendix.C

The pruning algorithm of [28] is not a correct pruning 2D DCT algorithm, because it ignores the permuting role of the matrix $\boldsymbol{P}_8$.
In [28], we know that

$$\boldsymbol{Y} = (\boldsymbol{R}_{81} \otimes \boldsymbol{R}_{81})(\boldsymbol{M}_8 \otimes \boldsymbol{M}_8)(\boldsymbol{R}_{82} \otimes \boldsymbol{R}_{82}) \cdot \boldsymbol{X}$$

$$\boldsymbol{F} = (\boldsymbol{P}_8 \otimes \boldsymbol{P}_8)(\boldsymbol{D}_8 \otimes \boldsymbol{D}_8) \cdot \boldsymbol{Y} \ ,$$

where $\boldsymbol{P}_8 = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & -1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & -1 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & -1 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{bmatrix}$ and $\boldsymbol{D}_8$ is a diagonal matrix.

$\boldsymbol{Y}$ is the scaled 2D DCT coefficient matrix and $\boldsymbol{F}$ is the 2D DCT coefficient matrix. Due to the process of $\boldsymbol{P_8}$, the element $Y(u,v)$ of $\boldsymbol{Y}$ is not corresponding to the element $F(u,v)$ of $\boldsymbol{F}$. For example:
According to the equation (7) in [28], $Y(1,0)$ is the second element of the matrix $\boldsymbol{R}_{81}\boldsymbol{M}_8\boldsymbol{R}_{82}(\boldsymbol{X}_0 + \boldsymbol{X}_1 + \boldsymbol{X}_2 + \boldsymbol{X}_3 + \boldsymbol{X}_4 + \boldsymbol{X}_5 + \boldsymbol{X}_6 + \boldsymbol{X}_7)$. $Y(1,0)$ is got as follows:

$$Y(1,0) = \sum_{i=0}^{7}\left[x(1,i) + x(2,i) + x(5,i) + x(6,i)\right] - \sum_{i=0}^{7}\left[x(0,i) + x(3,i) + x(4,i) + x(7,i)\right].$$

The value of $Y(1,0)$ is not corresponding to the element $F(1,0)$ and it is corresponding to the element $F(4,0)$ due to the process of $\boldsymbol{P_8}$. Moreover, The value of $Y(0,1)$ is not corresponding to the element $F(0,1)$ of the DCT coefficient matrix and it is corresponding to the element $F(0,4)$,etc.

The equation (5) not only is a scaling factor, but also plays a role of permuting the matrix $\boldsymbol{Y}$. So, the pruning scheme of the matrix $\boldsymbol{Y}$ using the zig-zag scanning pattern is not equivalent to the same pruning scheme of the DCT coefficient matrix $\boldsymbol{F}$ in [28]. So the pruning algorithm of [28] is not a correct pruning 2D DCT algorithm.