

FAST DCT ALGORITHM FOR VIDEO TRANSMISSION

Oleh
I Wayan Sutaya
Jurusan Teknik Elektronika, FTK, UNDIKSHA

ABSTRACT

This paper presents a fast DCT algorithm by using convolution method. Multipliers will be a crucial process in designing a fast DCT algorithm because they take a long time to complete their process. Reducing a number of multipliers is the one way to make a fast DCT algorithm. This paper shows how to reduce multipliers by using two steps: the first step is by grouping transformation matrix to be two groups in which this process uses the folding concept. The second step is by using fast convolution by inspection to minimize multipliers.

Keywords : *multiplier, fast DCT, fast convolution, folding.*

ABSTRAK

Paper ini membahas tentang sebuah algoritma *fast DCT* yang menggunakan metode konvolusi. *Multiplier* menjadi hal yang sangat penting dalam mendesain sebuah algoritma *fast DCT* karena ia memakan waktu banyak untuk menyelesaikan prosesnya. Pengurangan jumlah *multiplier* adalah salah satu cara dalam membuat sebuah algoritma *fast DCT*. Paper ini menunjukkan bagaimana cara mengurangi jumlah multiplier dengan menggunakan dua langkah yaitu: langkah pertaman dengan mengelompokkan matrik transformasi menjadi dua kelompok dimana proses ini menggunakan konsep melipat. Langkah kedua dengan menggunakan konvolusi cepat dengan cara inspeksi untuk meminimalisasi multiplier.

Kata-kata kunci : *multiplier, fast DCT, konvolusi cepat, melipat.*

1. INTRODUCTION

The Discrete Cosine Transform (DCT) is the core transform of many image processing applications for reduced bandwidth image and video transmission, including the JPEG image processing standard and high performance video coding standards such as MPEG and H.263.

Fast Dct Algorithm For Video Transmission.....(I Wayan Sutaya)

Because of the enormous popularity of the DCT, many researches had been published on fast DCT algorithms [Arai, 1988][Duhamel, 1990][Feig, 1992][Linzer, 1991], where the effort is devoted to reducing the number of arithmetic operations used. Scaled DCT algorithms rely on a post-pointwise scaling operation which removes some of the arithmetic operations from the main transform computation. In this paper, we discuss a new fast 1D-DCT algorithm by using fast convolution method. Several algorithms and architectures had previously been proposed to optimize DCT but rarely the algorithms which are created uses fast convolution method.

Fast Convolution is an implementation of convolution algorithm using fewer multiplication operations by algorithmic strength reduction that can be defined as Number of strong operations (such as multiplication operations) that is reduced at an expense of an increase in the number of weak operations (such as addition operations). These are best suited for implementation using either programmable or dedicated hardware.

2. THEORETICAL FRAMEWORK

2.1 Discrete Cosine Transform (DCT)

Discrete cosine transform (DCT) is a frequency transform used in still or moving video compression. We discuss the fast implementations of DCT based on algorithm-architecture transformations and the decimation-in-frequency approach. Denote the DCT of the data sequence $x(n)$, $n=0, 1, \dots, N-1$, by $X(k)$, $k=0, 1, \dots, N-1$. The equation for one-dimension N points DCT is as follows:

$$Z(u) = \alpha(u) \sum_{x=0}^{N-1} \left(f_{(x)} \cos \left[\frac{\pi(2x+1)u}{2N} \right] \right)$$

for $u=0,1,2 \dots N$ where $\alpha(u)$ is defined as

$$\alpha(u) = \begin{cases} \sqrt{\frac{1}{N}} & \text{for } u = 0 \\ \sqrt{\frac{2}{N}} & \text{for } u \neq 0 \end{cases}$$

It is clear that for $u=0$, we will have

$$Z(u = 0) = \sqrt{\frac{1}{N}} \sum_{x=0}^{N-1} (f(x))$$

Thus, the first DCT coefficient is the average value of the sample sequence. This value is called **DC coefficient** and all others values are called **AC coefficients**. The above equation can be treated as a matrix multiplication.

$$\mathbf{Z} = \mathbf{C}\mathbf{X}^t$$

where \mathbf{X} is a input matrix, \mathbf{C} is a transformation matrix, and \mathbf{Z} is a transformed matrix. For 8x8 points 1D-DCT, we will have the transformation matrix :

$$\mathbf{C} = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & c & f & g & -g & -f & -c & -b \\ d & e & -e & -d & -d & -e & e & d \\ c & -g & -b & -f & f & b & g & -c \\ a & -a & -a & a & a & -a & -a & a \\ f & -b & g & c & -c & -g & b & -f \\ e & -d & d & -e & -e & d & -d & e \\ g & -f & c & -b & b & -c & f & -g \end{bmatrix}$$

Figure 1. The transformation matrix

Where

$$a = \sqrt{\frac{1}{8}}, b = \sqrt{\frac{1}{4}} \cos\left(\frac{\pi}{16}\right), c = \sqrt{\frac{1}{4}} \cos\left(\frac{3\pi}{16}\right), d = \sqrt{\frac{1}{4}} \cos\left(\frac{\pi}{8}\right),$$

$$e = \sqrt{\frac{1}{4}} \cos\left(\frac{3\pi}{8}\right), f = \sqrt{\frac{1}{4}} \cos\left(\frac{5\pi}{16}\right), g = \sqrt{\frac{1}{4}} \cos\left(\frac{7\pi}{16}\right)$$

2.2. Fast Convolution

Reducing the multiplication complexity in complex number multiplication by assuming $(a+jb)(c+dj)=e+jf$, it can be expressed using the matrix form, which requires 4 multiplications and 2 additions:

$$\begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} c & -d \\ d & c \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix}$$

However, the number of multiplications can be reduced to 3 at the expense of 3 extra additions by using:

$$\begin{cases} ac - bd = a(c-d) + d(a-b) \\ ad + bc = b(c+d) + d(a-b) \end{cases}$$

Rewrite it into matrix form, its coefficient matrix can be decomposed as the product of a 2×3 (**C**), a 3×3 (**H**) and a 3×2 (**D**) matrix:

$$s = \begin{bmatrix} e \\ f \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 1 \end{bmatrix} \cdot \begin{bmatrix} c-d & 0 & 0 \\ 0 & c+d & 0 \\ 0 & 0 & d \end{bmatrix} \cdot \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 1 & -1 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \end{bmatrix} = C \cdot H \cdot D \cdot x$$

Where **C** is a post-addition matrix (requires 2 additions), **D** is a pre-addition matrix (requires 1 addition), and **H** is a diagonal matrix (requires 2 additions to get its diagonal elements) So, the arithmetic complexity is reduced to 3 multiplications and 3 additions (not including the additions in **H** matrix).

3. FAST ALGORITHM DCT PROCESS

The DCT process are showed in the Figure 2, where the output is a 8 x 1 matrix, the transformer is a 8 x 8 matrix and the transformed is a 8 x1 matrix.

$$\begin{bmatrix} y_1 \\ y_2 \\ y_3 \\ y_4 \\ y_5 \\ y_6 \\ y_7 \\ y_8 \end{bmatrix} = \begin{bmatrix} a & a & a & a & a & a & a & a \\ b & c & f & g & -g & -f & -c & -b \\ d & e & -e & -d & -d & -e & e & d \\ c & -g & -b & -f & f & b & g & -c \\ a & -a & -a & a & a & -a & -a & a \\ f & -b & g & c & -c & -g & b & -f \\ e & -d & d & -e & -e & d & -d & e \\ g & -f & c & -b & b & -c & f & -g \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \\ x_5 \\ x_6 \\ x_7 \\ x_8 \end{bmatrix}$$

Figure 2. The process of DCT

The process of DCT in Figure 2, can be folded into two groups as shown in Figure 3 and Figure 4.

$$\begin{bmatrix} y_1 \\ y_3 \\ y_5 \\ y_7 \end{bmatrix} = \begin{bmatrix} a & a & a & a \\ d & e & -e & -d \\ a & -a & -a & a \\ e & -d & d & -e \end{bmatrix} \begin{bmatrix} x_1 + x_8 \\ x_2 + x_7 \\ x_3 + x_6 \\ x_4 + x_5 \end{bmatrix}$$

Figure 3. The first group

$$\begin{bmatrix} y_2 \\ y_4 \\ y_6 \\ y_8 \end{bmatrix} = \begin{bmatrix} b & c & f & g \\ c & -g & -b & -f \\ f & -b & g & c \\ g & -f & c & -b \end{bmatrix} \begin{bmatrix} x_1 - x_8 \\ x_2 - x_7 \\ x_3 - x_6 \\ x_4 - x_5 \end{bmatrix}$$

Figure 4. The second group

Extracting the first group we will get the equation as follows

$$y_1 = a [(x_1 + x_8) + (x_2 + x_7) + (x_3 + x_6) + (x_4 + x_5)]$$

$$y_5 = a [(x_1 + x_8) - (x_2 + x_7) - (x_3 + x_6) + (x_4 + x_5)]$$

$$\begin{bmatrix} y_3 \\ y_7 \end{bmatrix} = \begin{bmatrix} d & e \\ e & -d \end{bmatrix} \begin{bmatrix} (x_1 + x_8) - (x_4 + x_5) \\ (x_2 + x_7) - (x_3 + x_6) \end{bmatrix}$$

$$y_3 = d[(x_1 + x_8) - (x_4 + x_5)] + e[(x_2 + x_7) - (x_3 + x_6)]$$

$$y_7 = e[(x_1 + x_8) - (x_4 + x_5)] - d[(x_2 + x_7) - (x_3 + x_6)]$$

By inspection we get equations from the second group.

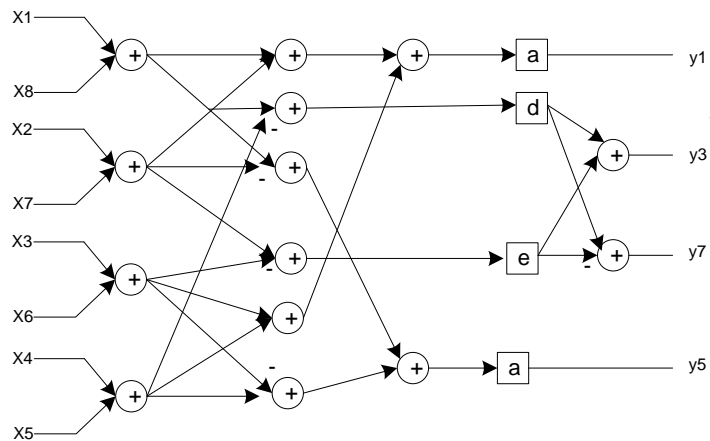
$$y_2 = b(x_1 - x_8) + c(x_2 - x_7) + f(x_3 - x_6) + g(x_4 - x_5)$$

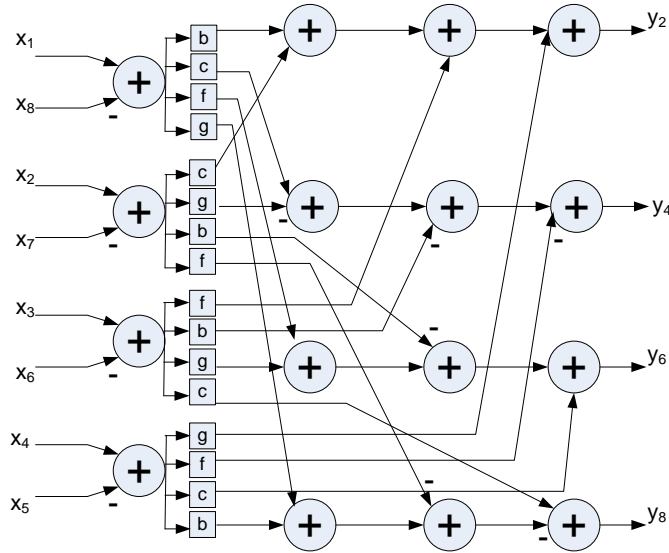
$$y_4 = b(x_1 - x_8) - g(x_2 - x_7) - b(x_3 - x_6) - f(x_4 - x_5)$$

$$y_6 = f(x_1 - x_8) - b(x_2 - x_7) + g(x_3 - x_6) + c(x_4 - x_5)$$

$$y_8 = g(x_1 - x_8) - f(x_2 - x_7) + c(x_3 - x_6) - b(x_4 - x_5)$$

The architecture as follows





Simulation on MATLAB results an output as follows



Figure 5. Matlab simulation

4. CONCLUSIONS

This work shows a fast DCT algorithm by using convolution by inspection method. Using direct DCT process need 64 multipliers but using this method only needs 22 multipliers. This method will be more optimal if the convolution that used in this algorithm use winograd convolution or cook-tom convolution.

REFERENCES

- E. Feig and S. Winograd, "Fast Algorithms for the discrete Cosine ASSP-40, no. 9, pp. 2166-2173, Sep, 1992.
- PeiZong Lee and Fang-Yu Huang, "Restructured Recursive DCT and DST Algorithms," IEEE Trans.
- K. H. Cheng, C. S. Huang, and C. P. Lin, "The design and implementation of DCT/IDCT chip with novel architecture," in *Proc. IEEE Int. Symp. Circuits Syst.*, Geneva, Switzerland, May 28–31, 2000, pp.IV-741–IV-744.
- Y. J. Chen, S. Oraintara, and T. Nguyen, "Video compression using integer DCT," in *Proc. Int. Conf. Image Processing*, vol. 2, 2000, pp.844–847.