Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X

IJCSMC, Vol. 3, Issue. 3, March 2014, pg.873 – 878

RESEARCH ARTICLE

Analysis of Integer Transformation and Quantization Blocks using H.264 Standard and the Conventional DCT Techniques

Priyanka P James[1], Chirappanath B Albert[2], Inbanila.K[3]

Christ University, Bangalore, 60
priyankajames1989@gmail.com
albertbazil@gmail.com
inbanila.k@christuniversity.in

Abstract:-

H.264 standard, transformation is a technique of converting the image samples into elementary frequency components. Integer Transformation helps in removing redundant data from an image and involves only real components and quantization reduces the precision of transform coefficients. H.264 is a lossy compression format because of Integer Transformation and Quantization. This paper deals with the understanding and the analysis in the reduction of complexity of integer transformation and quantization blocks using H.264 and the conventional techniques.

Keywords *—H.264 standards, Quantization, Compression, Integer Transformation*

I. INTRODUCTION

The image or video CODEC converts the image or the motion-compensated residual data into another domain, the transform domain[4]. The choice of transform depends on a number of criteria like

1. Data in the transform domain should be de-correlated, i.e. separated into components within minimal interdependence, and compact, i.e. most of the energy in the transformed data should be concentrated into a small of values.
2. The transform should be reversible.

3. The transform should be computationally tractable, that is it should be of low memory requirement, achievable using limited-precision arithmetic, low number of arithmetic operations, etc.

Discrete Cosine Transform (DCT) operates on blocks of N× N image or residual samples and hence the image is processed in units of a block[3]. Block transforms have low memory requirements and are well suited to compression of block-based motion compensation residuals but tend to suffer from artifacts at block edges (‘blockiness’). Image-based transforms operate on an entire image or frame or a large section of the image known as a ‘tile’[2].

The Discrete Cosine Transform (DCT) operates on X, a block of N × N samples, typically image samples or residual values after prediction, to create Y, an N × N block of coefficients[6]. The action of the DCT can be described in terms of a transform matrix A. The forward DCT (FDCT) of an N × N sample block is given by:

$$Y = AXA^T \tag{1.1}$$

Where X is a matrix of samples, Y is a matrix of coefficients and A is an N × N transform matrix. The elements of matrix ‘A’ are:

$$A_{ij} = C_i \cos\frac{(2j+1)i\Pi}{2N} \tag{1.2}$$

where

$$C_i = \sqrt{\frac{1}{N}}(i=0), C_i = \sqrt{\frac{2}{N}}(i>0)$$

$$Y_{xy} = C_x C_y \sum_{i=0}^{N-1}\sum_{j=0}^{N-1} X_{ij} \cos\frac{(2j+1)y\Pi}{2N} \cos\frac{(2i+1)x\Pi}{2N} \tag{1.3}$$

DCT of previous video coding standards provide transformation but produced inverse transform mismatch problems, due to floating point DCT. In order to overcome the mismatch problems, H.264 standard deals with integer transform[4].

II. H.264 INTEGER TRANSFORMATION AND QUANTIZATION

H.264 integer transform is free from multipliers and it deals with additions and shifts, which makes it low complex. In earlier standards, there was an obvious boundary between the transform, converting a block of image samples into a different domain, and quantizations, reducing the precision of transform coefficients [2].

This boundary is less obvious in an H.264 codec, with an overlap of the transform and quantization stages. This, together with the new approach of exactly specifying a reversible integer transform core, makes the H.264 transform and quantization stage significantly different from earlier compression standards [3].

The previous video coding standards relied on Discrete Cosine Transform (DCT) that provided the transformation but produced inverse transform mismatch problems.H.264/MPEG-4 part 10, uses an integer transform with a similar coding gain as a 4x4 DCT. It is multiplier-free, involves additions, shifts in 16-bit arithmetic, thus minimizing computational complexity, especially for low-end processes [1].

Integer transform is achieved by:

- Using a core transform, an integer transform, that can be carried out using integer or fixed point arithmetic
- Integrating a normalization step with the quantization process to minimize the number of multiplications required to process a block of residual data [1].

The scaling and inverse transform processes carried out by a decoder are exactly specified in the standard so that every H.264 implementation should produce identical results, eliminating mismatch between different transform implementations [5].

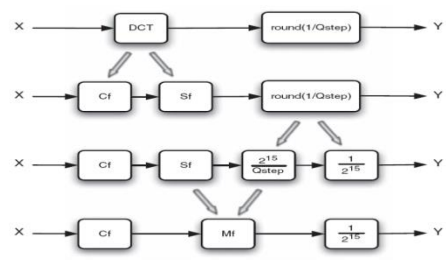

Figure.1. Development of integer transform and quantization [1]

Consider a block of pixel data that is processed by a two-dimensional Discrete Cosine Transform (DCT) followed by quantization, i.e. rounded division by a quantization step size, Q_{step}.

$$Y = AXA^T \tag{2.1}$$

$$A = \begin{bmatrix} a & a & a & a \\ b & c & -c & -b \\ a & -a & -a & a \\ c & -b & b & -c \end{bmatrix} \tag{2.2}$$

Where,

$$a = \frac{1}{2}, b = \sqrt{\frac{1}{2}}\cos(\frac{\Pi}{8}) = 0.6532$$

$$c = \sqrt{\frac{1}{2}}\cos(\frac{3\Pi}{8}) = 0.2706$$

$$C_{f4} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \tag{2.3}$$

$$A = C_f \bullet R_{f4} \tag{2.4}$$

Where,

$$R_{f4} = \begin{bmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{10}} & \frac{1}{\sqrt{10}} & \frac{1}{\sqrt{10}} & \frac{1}{\sqrt{10}} \\ \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \frac{1}{\sqrt{10}} & \frac{1}{\sqrt{10}} & \frac{1}{\sqrt{10}} & \frac{1}{\sqrt{10}} \end{bmatrix} \tag{2.5}$$

$$\begin{aligned} Y &= \left[C_{f4} \bullet R_{f4}\right] \cdot X \cdot \left[C_{f4}^T \bullet R_{f4}^T\right] \\ &= \left[C_{f4} \cdot X \cdot C_{f4}^T\right] \bullet \left[R_{f4} \bullet R_{f4}^T\right] \\ &= \left[C_{f4} \cdot X \cdot C_{f4}^T\right] \bullet S_{f4} \end{aligned} \tag{2.6}$$

Where,

$$S_{f4} = R_{f4} \bullet R_{f4}^T = \begin{bmatrix} \frac{1}{4} & \frac{1}{2\sqrt{10}} & \frac{1}{4} & \frac{1}{2\sqrt{10}} \\ \frac{1}{2\sqrt{10}} & \frac{1}{10} & \frac{1}{2\sqrt{10}} & \frac{1}{10} \\ \frac{1}{4} & \frac{1}{2\sqrt{10}} & \frac{1}{4} & \frac{1}{2\sqrt{10}} \\ \frac{1}{2\sqrt{10}} & \frac{1}{10} & \frac{1}{2\sqrt{10}} & \frac{1}{10} \end{bmatrix} \quad (2.7)$$

Scale the quantization process by a constant (2^{15}) and compensate by dividing and rounding. Combine Sf4 and the quantization process into Mf4.

$$M_f \approx \frac{S_f \cdot 2^{15}}{Q_{step}} \quad (2.8)$$

$$M_{f4} = m(QP\%6, n) / 2^{floor(QP/6)} \quad (2.9)$$

$$Y = round\left(\left[C_{f4} \right] \cdot \left[X \right] \cdot \left[C_{f4}^T \right] \bullet \frac{m(QP\%6, n)}{2^{floor(QP/6)} \cdot 2^{15}} \right) \quad (2.10)$$

Multiplication by two can be performed either through additions or through left shifts, so that no actual multiplication operations are necessary. Thus, the transform is multiplier-free [2].

III. MATHEMATICAL CALCULATIONS

1.DCT and Quantization

For the mathematical calculations of the DCT and the quantization the matrix A is taken as defined by the given standards. Assuming the matrix X is assumed

$$A = \begin{bmatrix} 0.5 & 0.5 & 0.5 & 0.5 \\ 0.6532 & 0.2706 & -0.2706 & -0.6532 \\ 0.5 & -0.5 & -0.5 & 0.5 \\ 0.2706 & -0.6532 & 0.6532 & -0.2706 \end{bmatrix} \qquad X = \begin{bmatrix} 58 & 64 & 51 & 58 \\ 52 & 64 & 56 & 66 \\ 62 & 63 & 61 & 64 \\ 59 & 51 & 63 & 69 \end{bmatrix}$$

Transformed and quantized output matrix $Y = AXA^T$

$$A \cdot X = \begin{bmatrix} 115.5 & 121 & 115.5 & 128.5 \\ -3.3592 & 8.7622 & -9.1914 & -6.644 \\ 1.5 & -6 & -1.5 & -1.5 \\ 6.2614 & 2.8646 & 0.0188 & -4.283 \end{bmatrix} \qquad Y = A \cdot X \cdot A^T = \begin{bmatrix} 240.25 & -7.0033 & 3.75 & -7.1104 \\ -5.2162 & 7.00387552 & -4.787 & -10.83842464 \\ -3.75 & 0.7419 & 3.75 & 3.7512 \\ 2.4309 & 7.659 & -0.452 & 0.995 \end{bmatrix}$$

$QP = 6 \Rightarrow Q_{step} = 1.25$

$$Y1 = \frac{A \cdot X \cdot A^T}{Q_{step}} = \begin{bmatrix} 192.2 & -5.60264 & 3 & -5.6883 \\ -4.17296 & 5.6 & -3.8296 & -8.6707 \\ -3 & 0.5936 & 3 & 3.0009 \\ 1.9448 & 6.1272 & -0.3616 & 0.796 \end{bmatrix}$$

2. Integer DCT and Quantization.

In the case of Integer DCT and Quantization technique the predefined matrix C is given below and the values of C are integres reducing the complexcity of the whole procedure . The matrix X is taken to be the same as in the previous case. In here the final values are rounded off.

$$Y = round\left(\left[C_{f4} \right] \cdot \left[X \right] \cdot \left[C_{f4}^{T} \right] \bullet \frac{m(QP\%6, n)}{2^{floor(QP/6)} \cdot 2^{15}} \right)$$

$$C_f = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \qquad X = \begin{bmatrix} 58 & 64 & 51 & 58 \\ 52 & 64 & 56 & 66 \\ 62 & 63 & 61 & 64 \\ 59 & 51 & 63 & 69 \end{bmatrix}$$

$$C_f \cdot X \cdot C_f^T = \begin{bmatrix} 961 & -41 & 15 & -48 \\ -34 & 72 & -30 & -104 \\ -15 & 3 & 15 & 24 \\ 13 & 81 & -5 & 8 \end{bmatrix} \qquad QP = 6$$

$$m(0,n) = \begin{bmatrix} 13107 & 8066 & 13107 & 8066 \\ 8066 & 5243 & 8066 & 5243 \\ 13107 & 8066 & 13107 & 8066 \\ 8066 & 5243 & 8066 & 5243 \end{bmatrix}$$

$$C_f \cdot X . C_f^T \cdot m(0,n) = \begin{bmatrix} 12595827 & -330706 & 196605 & -387168 \\ -274244 & 377496 & -241980 & -545272 \\ -196605 & 24198 & 196605 & 193584 \\ 104858 & 424683 & -40330 & 41944 \end{bmatrix}$$

$$Y = \begin{bmatrix} 192 & -5 & 3 & -6 \\ -4 & 5 & -3 & -8 \\ -3 & 0 & 3 & 3 \\ 1 & 6 & 0 & 0 \end{bmatrix}$$

IV. SIMULATION RESULTS

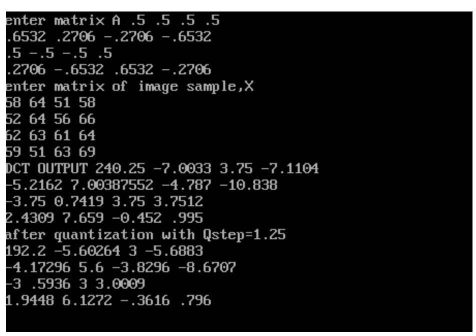
```
enter matrix A .5 .5 .5 .5
.6532 .2706 -.2706 -.6532
.5 -.5 -.5 .5
.2706 -.6532 .6532 -.2706
enter matrix of image sample,X
58 64 51 58
52 64 56 66
62 63 61 64
59 51 63 69
DCT OUTPUT 240.25 -7.0033 3.75 -7.1104
-5.2162 7.00387552 -4.787 -10.838
-3.75 0.7419 3.75 3.7512
2.4309 7.659 -0.452 .995
after quantization with Qstep=1.25
192.2 -5.60264 3 -5.6883
-4.17296 5.6 -3.8296 -8.6707
-3 .5936 3 3.0009
1.9448 6.1272 -.3616 .796
```

Figure.2. DCT and quantization output

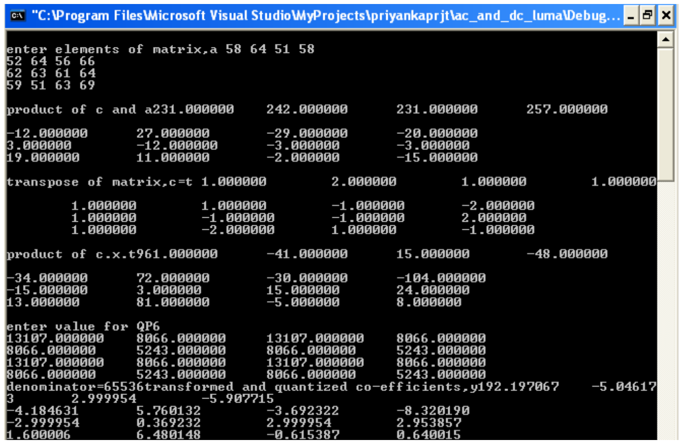

Figure.3.Integer DCT and Quantization output

V. CONCLUSION

From the above mathamatical calculations and the simulation results it is made clear that the H.264 gives us the most identitical compression performance to the DCT. The simulation of Integer DCT and Quantization output shown above is the stage before the rounding off is carried out. The interger DCT and quantization technique the output matrix Y is made simpler and reduced in complexicty as compared to the normal DCT and quantization technique only by the usage of basic mathematical operations like addition and substraction along with some phase shifters.

REFERENCES

[1] Iain E. Richardson, The H.264 *'Advanced Video Compression Standard'*, 2nd edition,Vcodex limited, UK, 2010

[2]Thomas Wiegand, Gary J.Sullivan, Gisle Bjonteguard and Ajay Luthra, *Overview of the H.264/AVC Video Coding Standard*,IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, July 2003.

[3] Tien- Ying Kuo and Chen-Hung Chan, *Fast Variable Block Size Motion Estimation for H.264 Using Likelihood and Correlation of Motion Field*, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 16, no. 10, October 2006.

[4] A.Ahmed,N.Khan.S.Masud,MA.Moud ,*Performance Evaluation of Advanced Features of H.26L Video Coding Standards* . Proceedings IEEE INMIC 2003

[5] H. Schwarz, D. Marpe and T. Wiegand, *Analysis of Hierarchical B Pictures and MCTF*, IEEE International Conference on Multimedia and Expo (2006), pp. 1929–1932.

[6] P. List, A. Joch, J. Lainema, G. Bjontegaard and M. Karczewicz, *Adaptive Deblocking Filter*, IEEE Transactions on Circuits and Systems for Video Technology, vol. 13, no. 7, July 2003