

An Early Detection Algorithm of All-Zero Blocks in H.264/AVC through the Vector Operations

Bo-Jhih Chen and Shen-Chuan Tai

Institute of Computer and Communication Engineering,
Department of Electrical Engineering,
National Cheng Kung University, Tainan, Taiwan
E-mail: cbj@lily.ee.ncku.edu.tw

Abstract - Motion estimation (ME) and discrete cosine transform (DCT) are major two parts for H.264/AVC, however, these two parts occupy most of computational time. Several studies focus on the fast motion estimation (FME) algorithm to reduce ME computations and then to accelerate the encoding process. FME has been optimized but we cannot neglect another parts on H.264/AVC; i.e., integer-DCT (I-DCT) / quantization. With this concern, we propose an efficient method through the vector operations to early detect all-zero block (AZB) before the transform and quantization on H.264. Experimental results show that the proposed method is superior to other algorithms in terms of the hit detection rate and the computational complexity reduction at the expense of insignificant degradation of video quality.

Keywords: H.264/AVC, DCT, I-DCT, AZB

1. Introduction

Most standard video coding technologies, like MPEG series, H.26x series [1]-[4] and H.264/AVC [5], are widely applied in various multimedia communications. Motion estimation, motion compensation, and discrete cosine transformation / quantization of H.264 are major processing functionalities for these codec standards. H.264/AVC is the latest video coding technology released by JVT team.

H.264/AVC outperforms existing video coding standards in terms of video coding performances [6], however, the considerable amount of computations of an H.264/AVC required encoder. A lot of research works have been dedicated on optimization of H.264/AVC including fast motion estimation and low-complexity transform and quantization [7]-[12].

Several studies concentrated on the optimization of motion estimation, but some works did not neglect to optimize the other portions to further speed up the whole of encoding in H.264. One way to accelerate the

encoder is to early detect an all-zero block (AZB) prior to transform process. Once a block is detected as an AZB, then transform/quantization can be skipped. An "AZB" block means that a 4×4 block of all sixteen transformed coefficients being zeros after quantization. Xuan [13] provided a simple method to verify that each residual block whether it omits the transform process. In [14], Sousa theoretically derived a sufficient condition for AZB detection. It based on the sum of absolute difference (SAD) and a quantization parameter (QP). Moon, et al. [15] proposed an approach derived from the theoretical observation of distinguish positions of transform coefficients in H.264.

In this paper, the proposed method for AZB detection is according to the significant relationship: a quantized block relative to multiplication factors (MF). Firstly, we use a novel model to form the transform signals in vector by the matrix direct product, and furthermore three criteria of AZB detection are derived by vector resultant. The detection rate and computational computations could be better than [14]-[15]. These will be presented in the experimental results. In addition, it ensures that our proposed method could be utilized to enhance the detection ability while reducing the computations relatively.

This paper is organized as follows. Integer DCT and quantization of the H.264, and previous works for detecting all-zero blocks are briefly introduced in Section 2. Our proposed method through vector operations is presented in Section 3. Section 4 introduces the experimental results to verify the efficiency of our proposed method for accelerating encode in H.264. The conclusions of this paper are given in Section 5.

2. Analysis of All Zero Block in encoder

2.1. Integer Discrete Cosine Transform and Quantization in H.264

For a 4×4 transform block Y . Each transform

coefficient, y_{uv} , $0 \leq u, v \leq 3$, is obtained by the integer DCT (I-DCT). I-DCT operation in H.264 can be defined as [5]

$$\mathbf{Y} = \mathbf{X}\mathbf{C}\mathbf{X}^T * \mathbf{E} = \mathbf{W} * \mathbf{E} \quad (1)$$

where \mathbf{X} is a 4×4 residual block, x_{ij} , $0 \leq i, j \leq 3$, \mathbf{C} and \mathbf{C}^T are integer transform matrices defined in H.264/AVC. The superscript “ T ” denotes the transpose of matrix and the operator “ $*$ ” represents the element-by-element multiplication in their corresponding indices. These two matrices are orthogonal matrices; i.e., $\mathbf{C}^T\mathbf{C} = \mathbf{C}\mathbf{C}^T = \mathbf{I}$, and \mathbf{E} is the post-scaling factor matrix defined in [5]. The size of transformed block of H.264 is 4×4 and the residual block is as well.

Each component of transformed block (\mathbf{W}) will be quantized by a quantization parameter (QP). The quantization process of H.264/AVC as follows

$$q_{uv} = \text{sign}(w_{uv}) \cdot (|w_{uv}| \cdot \mathbf{MF}_{QP_rem, idx} + c) \gg QBits \quad (2)$$

$$\text{, sign}(q_{uv}) = \text{sign}(w_{uv}) \text{ where } 0 \leq u, v \leq 3$$

where $QBits = 15 + \text{floor}(QP/6)$, $QP_rem = QP \% 6$, the constant $c = 2^{QBits} / 3$ for intra blocks or $c = 2^{QBits} / 6$ for inter blocks, and the symbol “ \gg ” indicates a binary shift right operator. The multiplication factor, $\mathbf{MF}_{QP_rem, idx}$, have been predefined which depends on their position and QP, and are listed in Table 1.

Table 1. Multiplication Factor, MF.

| $(u, v)^{\text{th}}$ | (0,0), (0,2), (2,0), (2,2) | (1,1), (1,3), (3,1), (3,3) | Other positions |
|----------------------|-------------------------------|-------------------------------|--------------------|
| QP_rem | $idx = 2$ | 0 | 1 |
| 0 | 13107 | 5243 | 8066 |
| 1 | 11916 | 4660 | 7490 |
| 2 | 10082 | 4194 | 6554 |
| 3 | 9362 | 3647 | 5825 |
| 4 | 8192 | 3355 | 5243 |
| 5 | 7282 | 2893 | 4559 |

where $idx = 2 - (u \bmod 2) - (v \bmod 2)$. After the quantization processing, quantized signal block \mathbf{Q} is obtained by eq.(2) for all 16 indices.

2.2. Conventional AZB Detection Methods

We list the previous approaches which are mainly to early detect the all-zero blocks as follows. In eq.(3), a threshold is theoretically derived for detecting AZB block with QP. [14]

$$SAD < TH_Sousa, \text{ where } TH_Sousa = \frac{TH(QP, 0)}{4} \quad (3)$$

Moon, et al. [15] proposed an approach based on the theoretical observation of transform coefficients as the eq. (4).

$$SAD < TH_Moon, TH_Moon = \min\{Th1, Th2\}$$

$$\text{where } Th1 = TH_Sousa + \frac{\min^1}{4}, Th2 = \frac{TH(QP, 1)}{2},$$

$$\min^1 = \min\{S_1 + S_2, S_3 + S_4\},$$

$$|x(i, j)|, \forall (i, j) \in A_k, 1 \leq k \leq 4, \quad (4)$$

$$\text{and } S_k = \sum_i \sum_j A_k = \begin{cases} A_1 = \{(i, j) | i = 0, 3, j = 0, 3\} \\ A_2 = \{(i, j) | i = 0, 3, j = 1, 2\} \\ A_3 = \{(i, j) | i = 1, 2, j = 1, 2\} \\ A_4 = \{(i, j) | i = 1, 2, j = 0, 3\} \end{cases}$$

These two methods mentioned above are used to detect AZB with SAD value.

3. Proposed Method Based on the Vector Operation

3.1. The Representation of DCT signal

Integer-DCT of H.264 was shown as the eq.(1). It also says that the row-transform followed by the column-transform and it can be depicted a butterfly signal flow diagram as Fig. 1.

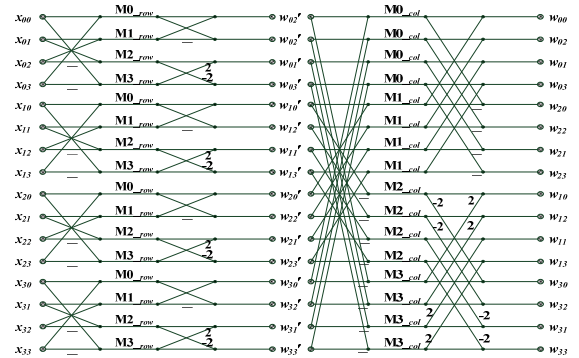


Figure 1. Signal Flow of 4x4

A 4×4 residual block is an input signal, x_{ij} , w_{uv} is a temporal result by row-transform, and the transformed block is an output signal, w_{uv} , $0 \leq u, v \leq 3$, through the row/column transformation. M0-M4 with subscript “row” and “col” are mediate signals as the row and column transformation, respectively.

The transformed signal can be composed of residual signals with weight values. These weight values can be defined as a 4×4 matrix. a transformed signal of $(u, v)^{\text{th}}$ can be form as the following:

$$w_{uv} = \vec{v}^{uv} \cdot * \mathbf{X}$$

$$= \begin{bmatrix} v_{00} & v_{01} & v_{02} & v_{03} \\ v_{10} & v_{11} & v_{12} & v_{13} \\ v_{20} & v_{21} & v_{22} & v_{23} \\ v_{30} & v_{31} & v_{32} & v_{33} \end{bmatrix}^{uv} * \begin{bmatrix} x_{00} & x_{01} & x_{02} & x_{03} \\ x_{10} & x_{11} & x_{12} & x_{13} \\ x_{20} & x_{21} & x_{22} & x_{23} \\ x_{30} & x_{31} & x_{32} & x_{33} \end{bmatrix} \quad (5)$$

Each residual signal (x_{ij}) is multiplied by a weight value (v_{ij}) in the same position in weight matrix \vec{v}^{uv} .

To simplify the implementation later, we use the strategy of “matrix direct product” to represent the transform process, which can be represented as

$$\text{vec}(\mathbf{W}) = (\mathbf{H} \cdot \text{vec}(\mathbf{X})) \quad (6)$$

where $\text{vec}(\cdot)$ denotes a vector, which size of 16×1 , that is, $\text{vec}(\mathbf{X})$ is a one-dimension vector of 4×4 residual block, and a transform block is represented as $\text{vec}(\mathbf{W})$.

$$\text{vec}(\mathbf{X}) = (x_{00} \ x_{01} \ x_{02} \ x_{03} \ x_{10} \ \cdots \ x_{33})^T \quad (7)$$

$$\text{vec}(\mathbf{W}) = (w_{00} \ w_{01} \ w_{02} \ w_{03} \ w_{10} \ \cdots \ w_{33})^T$$

The matrix \mathbf{H} is a 16×16 matrix and is defined as

$$\mathbf{H} = \mathbf{C} \otimes \mathbf{C} \quad (8)$$

where the operator “ \otimes ” denotes the “matrix direct product” with element defined by

$$h_{mn} = c_{ij} c_{kl} \quad (9)$$

where $m = 4i + k$ and $n = 4j + l$. \mathbf{C} is a 4×4 matrix defined in H.264. \mathbf{H} is also equal to p row vectors

$$\mathbf{H} = \begin{bmatrix} \mathbf{H}_0 \\ \mathbf{H}_1 \\ \mathbf{H}_2 \\ \vdots \\ \mathbf{H}_p \end{bmatrix} = \begin{bmatrix} (h_{00} \ h_{01} \ h_{02} \ \cdots \ h_{015}) \\ (h_{10} \ h_{11} \ h_{12} \ \cdots \ h_{115}) \\ (h_{20} \ h_{21} \ h_{22} \ \cdots \ h_{215}) \\ \vdots \\ (h_{p0} \ h_{p1} \ h_{p2} \ \cdots \ h_{p15}) \end{bmatrix} \quad (10)$$

and $p = 4u + v$.

For instance, the transform signal of $(0,0)^{\text{th}}$ is given by $w_{00} = \mathbf{H}_0 \cdot \text{vec}(\mathbf{X})$, transform signal of $(0,1)^{\text{th}}$ is $w_{01} = \mathbf{H}_1 \cdot \text{vec}(\mathbf{X})$ and so on. Therefore, each element of a transformed block can be depicted as

$$w_p = \mathbf{H}_p \cdot \text{vec}(\mathbf{X}) \quad (11)$$

where \mathbf{H}_p is a 16×1 row matrix of p .

3.2. Proposed Method of AZB Detection by Vector Resultants

To determine each transform coefficient is equal to zero after quantization. It requires the huge amount of computations to judge if quantized signal is less than

$\text{TH}(QP_rem,idx)$.

Given a quantization parameter (QP), we can easily get a multiplication factor (MF) listed in Table 1 to find a $\text{TH}(QP_rem,idx)$. The transform block (\mathbf{W}) and the thresholds (TH) are regarded as two 4×4 matrices, which specifies the relative coordinate of the $(u,v)^{\text{th}}$ position. These two matrices are represented in Fig. 2.

| | | | |
|-------|-------|-------|-------|
| (0,0) | (0,1) | (0,2) | (0,3) |
| (1,0) | (1,1) | (1,2) | (1,3) |
| (2,0) | (2,1) | (2,2) | (2,3) |
| (3,0) | (3,1) | (3,2) | (3,3) |

(a)

| | | | |
|-------|-------|-------|-------|
| (*,2) | (*,1) | (*,2) | (*,1) |
| (*,1) | (*,0) | (*,1) | (*,0) |
| (*,2) | (*,1) | (*,2) | (*,1) |
| (*,1) | (*,0) | (*,1) | (*,0) |

(b)

Figure 2. (a) A 4×4 transform signal block of w_{uv} , (b) A 4×4 matrix of threshold, $\text{TH}(*,idx)$.

As the QP_rem is fixed, the threshold $\text{TH}(*,idx)$ is constant on $idx = \{0, 1, 2\}$. These 16 transform elements can be classified into three sets depending on the $(u,v)^{\text{th}}$ position. S_{idx} is the set of transform signals according to the value of idx .

If all transform signals of these three sets in this transform block are less than threshold $\text{TH}(QP_rem,idx)$ concurrently, this transform block will be an AZB. According to eq.(11), transform signals can be represented as follows

$$S_{idx} = \begin{cases} S_2 = \{\mathbf{H}_p \cdot \text{vec}(\mathbf{X}) \mid u\%2 + v\%2 = 0\} \\ S_1 = \{\mathbf{H}_p \cdot \text{vec}(\mathbf{X}) \mid u\%2 + v\%2 = 1\} \\ S_0 = \{\mathbf{H}_p \cdot \text{vec}(\mathbf{X}) \mid u\%2 + v\%2 = 2\} \end{cases} \quad (12)$$

where $0 \leq p \leq 15$ and $p = 4u + v$.

In order to reduce the computations of AZB detection, find vector resultants to represent these three sets respectively. Using the “vector addition” operation, an identical vector (e.g. vector resultant) is obtained by row vectors \mathbf{H}_p in which of S_{idx} . Three identical vectors of S_{idx} are defined as

$$\begin{aligned} \mathbf{H}_{id(2)} &= \mathbf{H}_0 \oplus \mathbf{H}_2 \oplus \mathbf{H}_8 \oplus \mathbf{H}_{10} \\ \mathbf{H}_{id(1)} &= \mathbf{H}_1 \oplus \mathbf{H}_3 \oplus \mathbf{H}_4 \oplus \mathbf{H}_6 \oplus \\ &\quad \mathbf{H}_9 \oplus \mathbf{H}_{11} \oplus \mathbf{H}_{12} \oplus \mathbf{H}_{14} \\ \mathbf{H}_{id(0)} &= \mathbf{H}_5 \oplus \mathbf{H}_7 \oplus \mathbf{H}_{13} \oplus \mathbf{H}_{15} \end{aligned} \quad (13)$$

where $\mathbf{H}_{id(idx)}$ is an identical vector of S_{idx} , and the operator “ \oplus ” denotes the “vector addition” operation. Eq. (12) can be rewritten as

$$S_{idx} = \begin{cases} S_2 = \mathbf{H}_{id(2)} \cdot \text{vec}(\mathbf{X}) \\ S_1 = \mathbf{H}_{id(1)} \cdot \text{vec}(\mathbf{X}) \\ S_0 = \mathbf{H}_{id(0)} \cdot \text{vec}(\mathbf{X}) \end{cases} \quad (14)$$

According to the above procedures, the sufficient condition of AZB detection is defined as

$$S_{idx} < \text{TH}(QP_rem, idx) \text{ for } idx = 0, 1, \text{ and } 2 \quad (15)$$

Consequently, the transformed signals have been converged on three classes shown in Fig. 3.

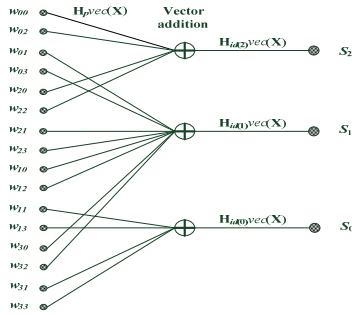


Figure 3. Signal flow after processing

4. Experimental Results

In this section, H.264 reference software JM9.8 [16] is implemented for the experiments. Six testing video sequences (“Akiyo”, “Coastguard”, “Foreman”, “News”, “Silent”, and “Table Tennis”), each of which is CIF format (352×288), are performed to evaluate the performance of our proposed method. The configuration of our simulation is one reference frame with IPPP frame structure (100 frames), rate-distortion optimization off (RDO = 0), and quantization parameters (QP) was set to 24, 28, 32, 36, and 40. For comparing our proposed method with other algorithms discussed in [14] and [15] are shown as follows.

Our proposed method in terms of the peak signal-to-noise ratio (PSNR) is compared with JM reference software and other algorithms. The degradation of PSNR ranges from 0.07 dB to 0.25 dB via the six video sequences and five QPs are shown in Table 2. In average, PSNR drop is about 0.15 dB. It is so small that it does not influence the human being’s visual quality at all.

In order to evaluate the overall performance and conveniently compare to the capacity for detection of AZBs, hit detection rate (*HDR*) and false detection rate (*FDR*) are critical values and are defined as

$$HDR = \frac{N_{\text{detected_AZB}}}{N_{\text{actual_AZB}}} \times 100\%, \quad (16)$$

$$FDR = \frac{N_{\text{false-detected_NAZB}}}{N_{\text{actual_NAZB}}} \times 100\%$$

where $N_{\text{detectedAZB}}$ is the number of AZBs detected by three early detection algorithms, $N_{\text{miss-detectedAZB}}$ is the number of AZBs without being detected, and $N_{\text{false-detectedAZB}}$ is the number of NAZBs (Non all-zeros blocks), that not all transformed coefficients are equal to zeros, being detected as AZBs. As far as these three values are concerned, the higher *HDR* presents more efficiently the algorithm can more detect AZBs correctly. If the *FDR* is smaller, it yields less the PSNR degradation and the accuracy of visual quality.

Table 2. Comparisons of PSNR

| QP | JM | [14] | [15] | Proposed | JM | [14] | [15] | Proposed |
|----|------------|-------|-------|--------------|--------------|-------|-------|--------------|
| | Akiyo | | | | News | | | |
| 24 | 42.07 | 42.06 | 42.07 | 41.94 | 40.66 | 40.64 | 40.66 | 40.53 |
| 28 | 39.75 | 39.75 | 39.75 | 39.61 | 38.12 | 38.12 | 38.12 | 38.00 |
| 32 | 37.15 | 37.15 | 37.15 | 37.02 | 35.33 | 35.33 | 35.33 | 35.21 |
| 36 | 34.79 | 34.78 | 34.79 | 34.66 | 32.65 | 32.63 | 32.65 | 32.54 |
| 40 | 32.35 | 32.35 | 32.35 | 32.25 | 29.96 | 29.96 | 29.96 | 29.90 |
| | Coastguard | | | | Silent | | | |
| 24 | 37.36 | 37.36 | 37.36 | 37.17 | 38.51 | 38.51 | 38.51 | 38.43 |
| 28 | 34.36 | 34.36 | 34.36 | 34.11 | 35.81 | 35.81 | 35.81 | 35.72 |
| 32 | 31.27 | 31.27 | 31.27 | 31.03 | 33.21 | 33.21 | 33.21 | 33.13 |
| 36 | 28.66 | 28.64 | 28.66 | 28.44 | 31.00 | 31.01 | 31.00 | 30.93 |
| 40 | 26.50 | 26.50 | 26.50 | 26.33 | 28.94 | 28.94 | 28.94 | 28.86 |
| | Foreman | | | | Table Tennis | | | |
| 24 | 39.06 | 39.05 | 39.06 | 38.85 | 38.31 | 38.3 | 38.31 | 38.14 |
| 28 | 36.67 | 36.67 | 36.67 | 36.46 | 35.54 | 35.54 | 35.54 | 35.35 |
| 32 | 34.23 | 34.23 | 34.23 | 34.03 | 32.77 | 32.77 | 32.77 | 32.62 |
| 36 | 32.02 | 32.01 | 32.02 | 31.83 | 30.53 | 30.52 | 30.53 | 30.39 |
| 40 | 29.77 | 29.77 | 29.77 | 29.60 | 28.75 | 28.75 | 28.75 | 28.63 |

Table 3. Comparisons of the false detection rate (FDR) of three methods

| QP | [14] | [15] | Proposed | [14] | [15] | Proposed |
|----|------|------|-------------------|---------------------|------|----------|
| | | | Akiyo | News | | |
| 24 | 0 | 0 | 7.39 | 0 | 0 | 7.56 |
| 28 | 0 | 0 | 9.64 | 0 | 0 | 8.82 |
| 32 | 0 | 0 | 9.32 | 0 | 0 | 9.39 |
| 36 | 0 | 0 | 11.95 | 0 | 0 | 11.04 |
| 40 | 0 | 0 | 13.23 | 0 | 0 | 12.95 |
| | | | Coastguard | Silent | | |
| 24 | 0 | 0 | 3.58 | 0 | 0 | 4.78 |
| 28 | 0 | 0 | 6.30 | 0 | 0 | 6.82 |
| 32 | 0 | 0 | 9.45 | 0 | 0 | 9.78 |
| 36 | 0 | 0 | 12.91 | 0 | 0 | 12.74 |
| 40 | 0 | 0 | 16.40 | 0 | 0 | 15.09 |
| | | | Foreman | Table Tennis | | |
| 24 | 0 | 0 | 7.65 | 0 | 0 | 5.41 |
| 28 | 0 | 0 | 10.42 | 0 | 0 | 8.19 |
| 32 | 0 | 0 | 11.66 | 0 | 0 | 10.58 |
| 36 | 0 | 0 | 12.14 | 0 | 0 | 12.59 |
| 40 | 0 | 0 | 12.81 | 0 | 0 | 12.73 |

The comparative results of *HDR* are illustrated in Fig. 4, the horizontal axis is QP and the vertical axis is the *HDR*, where we can see that our proposed method is superior to the others two algorithms because the proposed method defines the optimal condition to early detect the all-zero blocks. The detection of AZBs of the proposed method is more efficient than the others two algorithms and thereby mostly all zeros block in the video sequence were detected. These blocks correspond to approximately 60% - 94% of the total AZBs detected by the proposed method.

Table 3 shows the results of the *FDR* among our proposed method and the others two algorithms. The *FDR* of Sousa[14] and Moon[15] both are equal to zero due to they are lossless algorithms that NAZB was not classified to be an AZB for these two algorithms. In average, the *FDR* of our proposed method is about 3% - 16% in simulation, especially under the higher quantization parameter. This result can explain why video quality degrades in Table. 2 when applying our proposed in the procedure of video coding.

To further demonstrate the proposed method, we can consider the overall computations required for

detection of AZB in the whole encoding process. Required computations of algorithms for AZB detection are shown in Table 4.

Computation saving rate (*CSR*) is defined as

$$CSR = 1 - \frac{C_{method}}{C_{JM}} \times 100\% \quad (17)$$

where C_{method} is the encoding operations of DCT/Q for a method of AZB detection, and C_{JM} is the total encoding operations of DCT/Q in original encoder (JM).

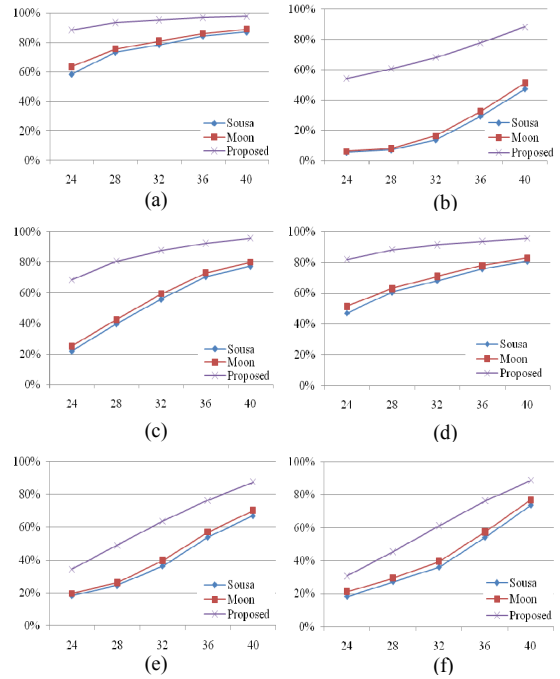


Figure 4. HDR curve of six video sequences. (a) Akiyo (b) Coastguard (c) Foreman (d) News (e) Silent (f) Table Tennis

Table 4. Required computational operations of DCT/Q per block

| Algorithms | Transform / Quantization | | (Inverse-)Transform / Quant. | |
|--|--------------------------|---------------|------------------------------|---------------|
| | ADD / MUL | SHIFT / CMP | ADD / MUL | SHIFT / CMP |
| 4×4 DCT | 64 / 0 | 16 / 0 | 64 / 32 | 16 / 0 |
| Additional operations (Prediction of AZB) | | | | |
| [14] | 15 / 0 | 0 / 1 | 0 | 0 |
| [15] | 18 / 2 | 0 / 3 | 0 | 0 |
| Proposed | 28 / 6 | 0 / 3 | 0 | 0 |

Fig. 5 shows the comparisons of *CSR* of methods for detecting the AZB. It means how many computations can be reduced over the encoding process. In average, it clearly observes that the proposed

method can get the better performance in reducing the computations than the other algorithms.

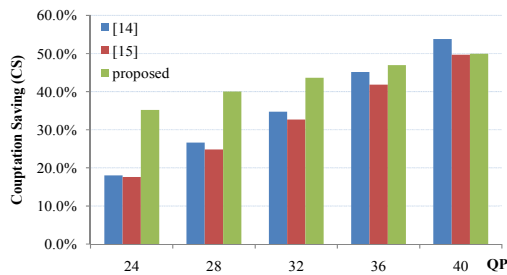


Figure 5. Computation Saving Rate(CSR) of the proposed method compared to [14] and [15] over six video sequences

5. Conclusions

In this paper, we propose an efficient method through vector operations to detect the AZB prior to transformation and quantization. Simulation results show that significant improvement in the detection rate and can be achieved with negligible video-quality degradation. Furthermore, it ensures that our proposed method could be utilized to enhance the detection ability while reducing the computations relatively.

Acknowledgments

This work was supported by the National Science Council, Taiwan, R.O.C. under the Grant No. NSC 96-2221-E-006-014.

References

[1] G. K. Wallace, "The JPEG still picture compression standard," *Communication ACM*, vol. 34, pp.30-44, Apr. 1991.

[2] "The MPEG-2 international standard," ISO/IEC, *Reference number ISO/IEC 13818-2*, 1996.

[3] "MPEG-4 Video Verification Model Version 18.0," *ISO/IEC JTC1/SC29/WG11 N3908*, Pisa, Italy, Jan. 2001.

[4] "Video Coding for Low Bit Rate Communication," ITU-T Recommendation. H.263, Feb. 1998.

[5] ISO/IEC MPEG 14496-10:2003, "Coding of Audiovisual Object – Part 10: Advanced Video Coding," 2003, also ITU-T Recommendation H.264 "Advanced video coding for generic audiovisual services."

[6] T. Wiegand, H.Schwarz, A. Joch, and f. Kossentini, "Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. on Circuits and Syst. for Video Technology*, vol. 13, pp. 488-703, July 2003.

[7] C. Zhu, W. S. Qi, and W. Ser, "Predictive fine granularity successive elimination for fast

optimal block-matching motion estimation," *IEEE Trans. Image Processing*, vol. 14, no. 2, pp. 213–221, Feb. 2005.

[8] S.-C. Tai, Y.-R. Chen, S.-J. Li, Low complexity variable-size block-matching motion estimation for adaptive motion compensation block size in H.264, in *Proc. the IEEE Asia-Pacific Conference on Circuits and Syst.*, pp. 613-616, 2004.

[9] A. Ahmad, N. Khan, S. Masud, and M. A. Maud, "Efficient block size selection in H.264 video coding standard," *Electronics Letters*, vol. 40, no. 1, pp. 19–21, Jan. 2004.

[10] A. Hallapuro, M. Karczewicz, and H. Malvar, "Low complexity transform and quantization," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Docs. JVT-B038 and JVT-B039*, Jan. 2002.

[11] A. Hallapuro and M. Karczewicz, "Low complexity (I) DCT," in *Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Docs. VCEV-N43*, Sept. 2001.

[12] H. S. Malvar, "Low-Complexity length-4 transform and quantization with 16-bit arithmetic," in *ITU-T SG16, Doc. VCEG-N44*, Sept. 2001.

[13] Z. Xuan, Z. Yu and S. Yu, "Method for detecting all-zero DCT coefficients ahead of discrete cosine transformation and quantization," *Electronics Letters*, vol. 34, No. 19, pp. 1839-1840, Sep. 1998.

[14] L. A. Sousa, "General method for eliminating redundant computations in video coding," *Electronics Letters*, vol. 36, no. 4, pp. 306-307, Feb. 2000.

[15] Y. H. Moon, G. Y. Kim, and J. H. Kim, "An improved early detection algorithm for all-zero blocks in H.264 video encoding," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 15, no. 8, pp. 1053-1057, Aug. 2005.

[16] H.264 Reference Software JM9.8, [Online] Available: <http://iphome.hhi.de/suehring/tml/>