16th European Signal Processing Conference (EUSIPCO 2008), Lausanne, Switzerland, August 25-29, 2008, copyright by EURASIP

EARLY DETECTION ALGORITHMS FOR 4×4 AND 8×8 ALL-ZERO BLOCKS IN H.264/AVC

Qin LIU, Yiqing HUANG, Takeshi IKENAGA

IPS, Waseda University
2-7 Hibikino, Wakamatsu, Kitakyushu, 808-0135, Japan
phone: + (81-80-3943-1143), fax: +81-93-692-5319, email: kenny67@toki.waseda.jp

ABSTRACT

H.264 is the latest HDTV video compression standard, which provides a significant improvement in coding efficiency at the cost of huge computation complexity. Provided that discrete cosine transform coefficient(DCT) blocks can be quantized to all zeros, the process of transform and quantization on this all-zero block(AZB) can be skipped, which will reduce significant redundant computations. In this paper, a theoretical analysis is performed for the sufficient condition for AZB detection. As a result, two partial sum of absolute differences (SAD) based 4×4 AZB detection algorithm are derived. Furthermore, a quantization parameter (QP) oriented 8×8 AZB detection algorithm is proposed according to the AZB's statistical analysis. Experimental results show that the proposed algorithms outperform the previous methods in all cases and achieve major improvement of computation reduction in the range from 12.16% to 42.68% for 4×4 transform, from 4.86% to 73.55% for 8×8 transform. The computation reduction increases as the QP increases.

1. INTRODUCTION

Early termination is one popular method in video compression standards such as H.264 and MPEG 4, which does some early detection and skips some redundant operations for motion estimation. After some theoretical analysis, early termination method can be also utilized in the transform and quantization part. For 4×4 and 8×8 DCT transform and quantization, the conventional way shows in Fig. 1 (a), which execute the transform(T) and quantization(Q) for each residual block. However, if there is a method of early detecting the AZB whose transformation coefficients will be quantized to zeros, then we can skip the transformation and quantization and save considerable computation, which is presented in Fig. 1 (b).

Many works have been done in this field and reduced the computation based on the above principle. Zhou[1] first introduced AZB detection method into H.263 codec, which provided a SAD threshold value for 8×8 block. Since a 4×4 DCT-like integer transformation is used in H.264 instead of a DCT transformation, Sousa[2], Moon[3] and Su[4] both develop sufficient conditions for quantizing all 4×4 integer transformation coefficients to zero. More reduction in numbers of calculations have achieved in Su's proposal than Sousa and Moon's algorithm. However, Su' method is unfeasible to be implemented by hardware.

To extend these ideas, we proposed two improved SAD based sufficient conditions for AZB detection, for 4×4 block and 8×8 block, respectively. Owing to take consideration of feasibility of hardware implementation, we modified the conditions a little, which still keep high accuracy for AZB detection and acquire hardware-oriented feature at the same time.

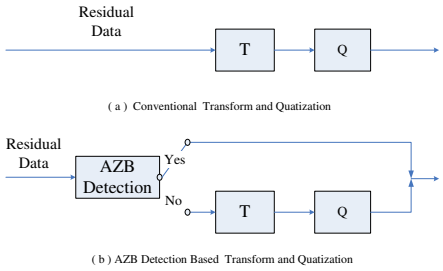


Figure 1: AZB detection diagram

The rest of the paper is organized as follows. Two enhanced SAD based 4×4 all-zero block detection algorithms will be introduced explicitly in section 2. In section 3, details about a 8×8 all-zero block detection algorithm will be explained. Experiments results both on detection accuracy and computation reduction are shown in the Section 4. At last, a conclusion will be presented in Section 5.

2. TWO PROPOSED 4×4 SAD BASED ALL ZERO BLOCK DETECTION ALGORITHMS

2.1 All-zero block detection

An approximation to the original 4×4 DCT transform is defined in H.264 by Eq. 1. CXC^T is a 'core' 2D transform, while E is a matrix of scaling factor and the symbol $\otimes$ indicates that each element of CXC^T is multiplied by the scaling factor in the same position in matrix E.

$$
\begin{aligned}
Y &= (CXC^T) \otimes E \\
C &= \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} \\
E &= \begin{bmatrix} a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2} \\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4} \end{bmatrix} a = \tfrac{1}{2}, b = \sqrt{\tfrac{2}{5}}
\end{aligned}
\tag{1}
$$

H.264 assumes a scalar quantizer. The basic forward quantizer operation is shown by Eq. 2.

$$Z_{ij} = round(Y_{ij}/Qstep) \tag{2}$$

where Y_{ij} is a coefficient of the transform described above, Qstep is a quantizer step size and Z_{ij} is a quantized coefficient. The rounding operation here rounds to the nearest smaller integer.

In order to simplify the calculations, The post-scaling factor E in Eq. 1 is absorbed in forward quantizer. It is implemented in the JM software as a multiplication by a factor $MF_{(QP\%6,Pos)}$ and a right shift, avoiding any division operations. Therefore, in the integer arithmetics, the quantization is implemented as follows:

$$\begin{aligned} &W = (CXC^T) \\ &|Z_{ij}| = round(|W_{ij}| \cdot MF_{(QP\%6,Pos)} + f) \gg qbits \\ &sign(Z_{ij}) = sign(W_{ij}) \\ &qbits = 15 + floor(QP/6) \end{aligned} \tag{3}$$

where f is $2^{qbits}/3$ for Intra blocks or $2^{qbits}/6$ for Inter blocks.

If all the Z_{ij} in Eq. 3 in the 4×4 matrix are less than 1, this 4×4 block is AZB.

$$\begin{aligned} &|Z_{ij}| < 1 \\ \Leftrightarrow &|W_{ij}| < \frac{2^{qbits} - f}{MF_{(QP\%6,Pos)}} \\ \Leftrightarrow &|[CXC^T]_{ij}| < \frac{2^{qbits} - f}{MF_{(QP\%6,Pos)}} \end{aligned} \tag{4}$$

If we can calculate all the W_{ij} and check the criteria above, then we can find whether this 4×4 block is an all-zero block. But owing to the complicated calculation for W_{ij}, we should find a simple criteria for it. When we do the motion estimation, we will calculate the SAD value. So if we can reuse this value, we can do some early detection for AZB without transform and quantization.

2.2 Enhanced partial SAD based 4×4 all-zero block detection algorithms

After analyzing the absolute value of the core transform $W = CXC^T$

$$\begin{aligned} W &\le ([|C_{ij}|]_{4\times4}[|X_{ij}|]_{4\times4}[|C_{ij}|]^T_{4\times4}) \\ &= \begin{bmatrix} S & S+S_0+S_2 & S & S+S_1+S_3 \\ S+S_0+S_1 & 2S+2S_0-S_3 & S+S_0+S_1 & 2S+2S_1-S_2 \\ S & S+S_0+S_2 & S & S+S_1+S_3 \\ S+S_2+S_3 & 2S+2S_2-S_1 & S+S_2+S_3 & 2S+2S_3-S_0 \end{bmatrix} \end{aligned} \tag{5}$$

where $S_0 = |x_{00}| + |x_{03}| + |x_{30}| + |x_{33}|, S_1 = |x_{01}| + |x_{02}| + |x_{31}| + |x_{32}|, S_2 = |x_{10}| + |x_{13}| + |x_{20}| + |x_{23}|, S_3 = |x_{11}| + |x_{12}| + |x_{21}| + |x_{22}|, S = S_0 + S_1 + S_2 + S_3 = SAD$.

When SAD have been calculated in ME part, we can get S_0,S_1,S_2 and S_3. It will not increase the computation complexity for software implementation. So we can get $|W_{ij}|$'s upper bounds for 3 different position.

$$|W_{ij}| \le \begin{cases} 2S + 2max(S_i) - min(S_i) & \text{Pos=2} \\ S + 2max(S_i) & \text{Pos=1} \\ S & \text{Pos=0} \end{cases} \tag{6}$$

As we can conclude from Eq. 4, for three different positions, there are three thresholds according to different MF's value.

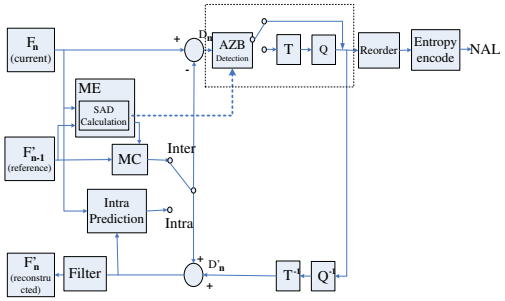


Figure 2: AZB detection in H.264 encoder

Thus, from Eq. 4 and Eq. 6, our proposed AZB detection criteria for different position is shown below

$$\begin{aligned} SAD &< \frac{2^{qbits} - f}{2 \times (MF_{(QP\%6,0)})} - max(S_i) + \frac{min(S_i)}{2} \\ &= 2T(0) - max(S_i) + \frac{min(S_i)}{2} \text{for Pos=2} \\ SAD &< \frac{2^{qbits} - f}{(MF_{(QP\%6,1)})} - 2max(S_i) \text{for Pos=1} \\ &= 2T(1) - 2max(S_i) \\ SAD &< \frac{2^{qbits} - f}{(MF_{(QP\%6,2)})} \text{for Pos=0} \\ &= T(2) \end{aligned} \tag{7}$$

Thus, the minimum value of three thresholds is our proposed SAD based sufficient condition's threshold for AZB detection: $min(2T(0) - max(S_i) + \frac{min(S_i)}{2}, 2T(1) - 2max(S_i), T(2))$. Compared with Su's idea, we modified the threshold for Pos 2, which achieves more AZBs' detection. This proposal for AZB detection is called P1 in the following text.

Although our proposed threshold is more accurate, it is not easy to be implemented in hardware. Both conventional SAD tree and PPSAD SAD calculation hardware architectures calculate the SAD row by row[5], which they can fetch the computation data in the regular and sequential way with low power consumption. So it is not easy to get the S_0, S_1, S_2, S_3 values in hardware implementation, which is irregular SAD accumulation. Thus, a hardware-oriented AZB detection is proposed. Owing to the space continuity for the image, we utilize R_0, R_1, R_2, R_3 to instead of S_0, S_1, S_2, S_3. R_0, R_1, R_2, R_3 are the sums of difference for each row, which are easy to get by the SAD hardware.$R_0 = |x_{00}| + |x_{01}| + |x_{02}| + |x_{03}|, R_1 = |x_{10}| + |x_{11}| + |x_{12}| + |x_{13}|, R_2 = |x_{20}| + |x_{21}| + |x_{22}| + |x_{23}|, R_3 = |x_{30}| + |x_{31}| + |x_{32}| + |x_{33}|$.

Therefore, our second threshold for SAD to detect AZB is $min(2T(0) - max(R_i) + \frac{min(R_i)}{2}, 2T(1) - 2max(R_i), T(2))$, which is called P2 in the following paragraph.

Fig. 2 shows the whole diagram for the encoder. When the ME part calculates the SAD, the partial SADs for each row(R_0, R_1, R_2, R_3) will help AZB detection module to set the threshold for AZB according to Eq. 7(dashed arrow in Fig. 2). AZB detection module compares threshold value with SAD value to decide whether to skip the transformation and quantization process.

2.3 Compared with conventional all-zero block detection algorithm in H.264

Sousa got this simple condition based on SAD for AZB detection

$$SAD < \frac{2^{qbits} - f}{4 \times (MF_{(QP\%6,0)})} \tag{8}$$

This criteria is a very rigor sufficient condition which uses $4S$ as W_{ij}'s upper bound and use the largest $MF_{(QP\%6,0)}$ instead of $MF_{(QP\%6,1)}$ and $MF_{(QP\%6,2)}$ as the divisor in Eq. 4. Thus, the value of Sousa's proposal is too small to detect the AZB in many conditions.

Moon got following criteria

$$SAD < \frac{2^{qbits} - f}{C(r) \times (MF_{(QP\%6,r)})} = T(r) \tag{9}$$

After some carefully examination for W_{ij}, he use following threshold for SAD

$$TH = min(T(0) + \lambda/2, T(1)) \tag{10}$$

Moon's method use a smaller upper bound for W_{ij}. But it only take into consideration of Pos 0 and 1, which many AZBs are not able to be detected by this method.

Su's threshold for SAD is shown below.

$$\begin{aligned} TH = & min(4T(0) - 5max(S_i), 2T(1) - 2max(S_i), \\ & T(2)) \end{aligned} \tag{11}$$

S_i is part of SAD. Su's first upper bound(UB) for Pos 2 is $S + 5max(S_i)$. Thus, from Eq. 6 and Eq. 11, we can prove that our upper bound is less than Su' idea which means more accurate for AZB detection. The smaller the upper bound of all three position's W_{ij} is, the larger threshold is, and the more AZBs are detected. More AZBs are detected and the detection accuracy decreases. Our proposal reach a good tradeoff between detection numbers and detection accuracy. Thus, we prove the upper bounds' relationship of W_{ij} in Pos 2 between proposed P1 and Su' method in the following:

$$\begin{aligned} UB_{our} - UB_{su} &= 2S + 2max(S_i) - min(S_i) - (S + 5max(S_i)) \\ &= S - 3max(S_i) - min(S_i) \\ &= S_1 + S_2 - 2max(S_i) <= 0 \\ \text{assume } & max(S_i) = S_0, min(S_i) = S_3 \end{aligned} \tag{12}$$

Our proposed upper bound of W_{ij} for Pos 2 is smaller than Su's method. Thus, our proposed threshold of P1 is larger than Su's threshold, which means that P1 is easy to detect much more AZBs. After some experimental examinations, our proposed P2 method's threshold is also larger than Su's threshold. At last, according to above analysis, it is concluded:

$$TH_{P1} > TH_{P2} > TH_{SU} > TH_{Moon} > TH_{Sousa} \tag{13}$$

Our two proposal P1 and P2 algorithm both can achieve better performance in AZB detection than any other conventional algorithms. Furthermore, P2 can obtain the feasibility to be implemented by hardware.

3. ENHANCED QP ORIENTED 8×8 ALL ZERO BLOCK DETECTION ALGORITHM

The concept of adapting the transform size to the block size of motion-compensated prediction has proven to be a promising coding tool within the H.264/AVC video coding layer design. Gordon[6] proposed a seamless integration of a new 8x8 transform into H.264. It is more suitable for high resolution application. A new 8×8 array C and new MF table are defined in [6].

For H.263's 8×8 block, Zhou proposed a simple AZB detection condition:

$$SAD < 8 \times QP \tag{14}$$

But for H.264's 8×8 block, this condition is not accurate. From Fig. 3, we can see the Zhou's threshold always makes error detection for AZB, especially for low QP value. For example QP=28, the threshold's value of Zhou is 224. There are 119057 AZBs. But Zhou's method regard 188243 block as AZBs. 77466 out of 188243 blocks are error detection.

Thus, we proposed a SAD based 8×8 block AZB detection algorithm. For most all-zero 8×8 block, the luminance difference for neighbor pixel is very small. If we regard all the residue data as the same, the core transform "CXC^T" in Eq. 1 become

$$|W| \approx C(I \otimes x)C^T. \tag{15}$$

I is a matrix, whose elements in all positions are 1. while x is the residue data which are assumed as all the same for 8×8 matrix. After calculation of right hand side part in Eq. 15, we can simplify the evaluation for the upper bound of W_{ij} in Eq. 16.

$$|W_{ij}| \leq \begin{cases} S & \text{Pos=0} \\ 60.0625/64 \times S & \text{Pos=1} \\ 36/64 \times S & \text{Pos=2} \\ 62/64 \times S & \text{Pos=3} \\ 48/64 \times S & \text{Pos=4} \\ 46/64 \times S & \text{Pos=5} \end{cases} \tag{16}$$

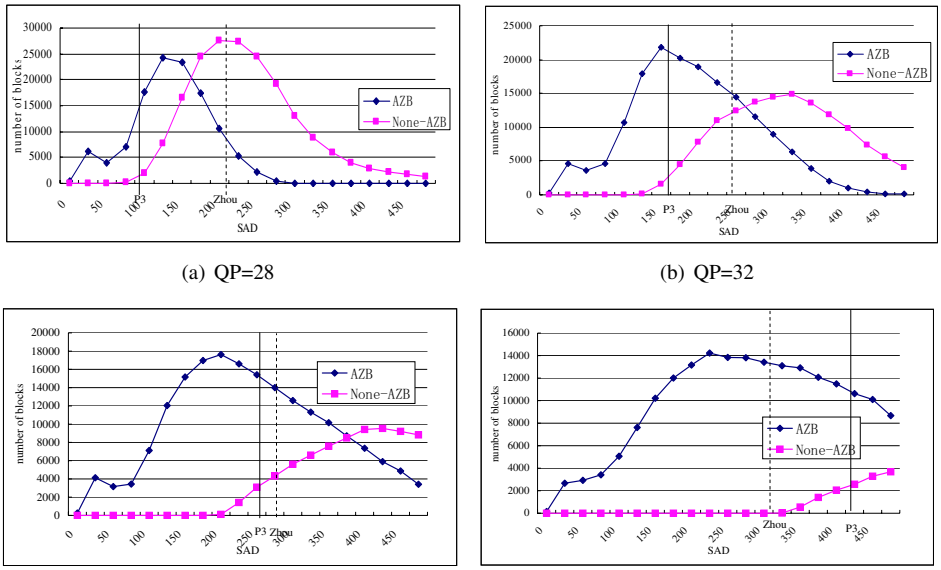
Then, we can easily obtain these 6 thresholds for 6 Positions: Eq. 17, 18, 19. Our proposed 8×8 block all-zero block is min(TH0, TH1, TH2, TH3, TH4, TH5). This proposed 8×8 block all-zero block algorithm is called P3 in the following paragraphs.

$$\begin{aligned} SAD &< \frac{2^{qbits} - f}{(MF_{(QP\%6,0)})} = TH0 \\ SAD &< \frac{(2^{qbits} - f) \times 64}{(MF_{(QP\%6,1)}) \times 60.025} = TH1 \end{aligned} \tag{17}$$

$$\begin{aligned} SAD &< \frac{(2^{qbits} - f) \times 64}{(MF_{(QP\%6,2)}) \times 36} = TH2 \\ SAD &< \frac{(2^{qbits} - f) \times 64}{(MF_{(QP\%6,3)}) \times 62} = TH3 \end{aligned} \tag{18}$$

$$\begin{aligned} SAD &< \frac{(2^{qbits} - f) \times 64}{(MF_{(QP\%6,4)}) \times 48} = TH4 \\ SAD &< \frac{(2^{qbits} - f) \times 64}{(MF_{(QP\%6,5)}) \times 46} = TH5 \end{aligned} \tag{19}$$

This threshold is only related to QP value, without any relationship with the content of the video sequence. Thus, it can be decided in advance without further calculation. In Fig. 3, two vertical lines show our proposed P3 and Zhou's threshold's value(dashed line). For small QP, Zhou's method always does wrong detection. While for large QP, P3's threshold are larger than Zhou's threshold, which means more easily to detect more AZBs.



(a) QP=28 (b) QP=32

(c) QP=36 (d) QP=40

Figure 3: Numbers of AZB and None-AZB for different 8×8 SAD

Table 1: Number of AZBs detection for different 4×4 methods

	QP	Sousa	Su	P1	P2
foreman	28	365964	836497	897841	875644
	32	755727	1294655	1354023	1325895
	36	1165968	1707791	1765995	1742257
	40	1501342	1953454	1979795	1960217
mother	28	797426	1124757	1161634	1153141
	32	1079646	1476421	1476421	1512426
	36	1389551	1803165	1851698	1844428
	40	1678141	1678141	2044912	2034783
news	28	726971	1143829	1184081	1138335
	32	1144093	1562228	1602430	1579778
	36	1582186	1869107	1900881	1889201
	40	1810813	2017781	2033500	2026335

Table 2: The accuracy of AZBs detection for different 4×4 methods

	QP	Sousa	Su	P1	P2
foreman	28	100%	99.97%	99.97%	99.95%
	32	100%	99.98%	99.98%	99.97%
	36	100%	99.61%	99.49%	99.57%
	40	100%	99.73%	99.71%	99.75%
mother	28	100%	99.93%	99.92%	99.86%
	32	100%	99.89%	99.87%	99.87%
	36	100%	99.79%	99.72%	99.74%
	40	100%	99.95%	99.95%	99.95%
news	28	100%	99.96%	99.96%	99.96%
	32	100%	99.99%	99.98%	99.97%
	36	100%	99.86%	99.81%	99.83%
	40	100%	99.95%	99.94%	99.93%

4. EXPERIMENTAL RESULTS AND PERFORMANCE ANALYSIS

To evaluate our proposed algorithms, lots of experiments have been done both on the calculation burden and video quality. QP is set as 28,32,36,40. We implement our algorithms based on H.264's reference software JM12.0.

Table 1 lists the numbers of AZBs detected in different methods. It is explicit that P1, has the largest number of detected AZBs in all cases. As we proved previously, the P1's SAD based threshold for AZB is superior to Su's method's threshold. Furthermore, the number of detected AZBs for our proposed P2, are slightly less than P1's number, still more than Su's method. But P2 achieves the feasibility for hardware implementation at a cost of slightly accuracy loss.

Table 2 shows the comparison of the accuracy of AZBs detection for different methods. If the detected AZB is indeed an all-zero block, we take this AZB detection as correct. From Table 2, we can conclude: Sousa's method is absolutely correct, while other methods are with slight error detection. Our proposed methods, P1 and P2 keep 99.9% accuracy for all the sequences and QP values, almost as same as Su's method. For 8×8 all-zero block detection, we compared our proposed P3 and Zhou's idea in Table 3. When the value of QP is low, Zhou's method will detect more AZBs with large error detection rate. Our proposed P3 always keep almost 99.9% correct detection rate for difference QPs.

To examine how much calculation is saved, we calculate the total number of operations needed for the proposed method and compare it with that of the other methods respectively. Table 4 shows the calculation consumed in each AZB detection method. According to the analysis in [3], a 4×4 non-all-zero block(NAZB) requires 80 additions, 16 multiplications, and 32 shift operations to perform transformation and quantization. A 8×8 NAZB consumes 64 additions and 20 shift operations[6]. For sousa's method, since its threshold is a fixed value, it does not need any extra calculation for generating threshold. Su's method requires 4 comparisons(3 for getting the $max(S_i)$ and 1 for the final min operation), 2 multiplication, and 2 additions. Compared with Su's method,

Table 3: Performance comparison of different 8×8 methods

	QP	Num. of AZB		Accuracy of AZBs	
		Zhou's	P3	Zhou's	P3
foreman	28	188243	15320	58.85%	99.43%
	32	150882	50380	81.65%	99.42%
	36	126265	101621	94.63%	98.44%
	40	110018	164036	100%	97.87%
mother	28	85186	1124757	75.89%	99.24%
	32	157766	119299	94.07%	99.49%
	36	159123	150481	98.76%	99.54%
	40	160405	193600	100%	99.72%
news	28	217713	62847	72.77%	99.99%
	32	206787	121451	92.89%	99.96%
	36	208694	183769	98.98%	99.93%
	40	202110	231544	100%	99.87%

Table 4: Calculation of AZBs detection for different methods

		ADD	MUL	SHIFT	CMP
C_{4x4}	NAZB	80	16	32	0
C_{8x8}	NAZB	64	0	20	0
C_{Sousa}	AZB	0	0	0	0
C_{Su}	AZB	2	2	0	5
C_{P1}	AZB	3	0	2	6
C_{P2}	AZB	3	0	2	6
C_{P3}	AZB	0	0	0	0

our proposed method, P1 and P2, need 1 extra comparison for getting the $min(S_i)$ and 1 extra addition to generate threshold of Pos 2. The value of P3's threshold is decided in advanced, thus it needs not extra calculation process.

Let N be the total number of 4×4 blocks in a video sequence, N_{NAZB} and N_{AZB} are the number of none-all-zero block and all-zero block. Since the comparison of SAD and threshold is required for each 4×4 block or 8×8 block, the total calculation(TC) for different methods equal to:

$$\begin{aligned} TC_{JM} &= N \times C_{NAZB} \\ TC_{Sousa} &= N_{NAZB} \times C_{NAZB} + N_{AZB} \times C_{Sousa} + N \\ TC_{Su} &= N_{NAZB} \times C_{NAZB} + N_{AZB} \times C_{Su} + N \\ TC_{P1} &= N_{NAZB} \times C_{NAZB} + N_{AZB} \times C_{P1} + N \\ TC_{P2} &= N_{NAZB} \times C_{NAZB} + N_{AZB} \times C_{P2} + N \\ TC_{P3} &= N_{NAZB} \times C_{NAZB} + N_{AZB} \times C_{P3} + N \end{aligned} \tag{20}$$

TC_{JM} is the total calculation for JM software without all-zero block detection. Define the total calculation reduction percentage(TCRP) as following.

$$\begin{aligned} TCRP1 &= \frac{TC_{Sousa} - TC_{Su}}{TC_{Sousa}} \times 100\% \quad & TCRP2 &= \frac{TC_{Sousa} - TC_{P1}}{TC_{Sousa}} \times 100\% \\ TCRP3 &= \frac{TC_{Sousa} - TC_{P2}}{TC_{Sousa}} \times 100\% \quad & TCRP4 &= \frac{TC_{JM} - TC_{P3}}{TC_{JM}} \times 100\% \end{aligned} \tag{21}$$

Table 5 shows TCRPs in details. Our proposed 4×4 block AZB detection algorithm outperforms the previous methods in all cases and achieves major improvement of computation reduction in the range from 6.7% to 42.3% compared to Sousa's method, a little improvement compared with Su's method. Furthermore, our proposed algorithm P2 is quite easy to be implemented by hardware. For P3 method, from 0.24% to 73.55% computation reduction achieves compared with original JM software without all-zero block. Furthermore, it can be concluded that the larger the value of QP is, the more the computation reduction achieves. The reduction percentage is related to the content of video sequence.

Table 5: The TCRP for different methods

	QP	TCRP1	TCRP2	TCRP3	TCRP4
foreman	28	22.16%	24.48%	23.38%	4.86%
	32	30.51%	32.83%	31.08%	16.01%
	36	39.87%	42.38%	40.33%	32.24%
	40	41.57%	42.68%	40.22%	52.04%
mother	28	17.40%	18.54%	17.99%	27.02%
	32	25.61%	26.95%	26.47%	37.85%
	36	34.48%	36.32%	35.53%	47.74%
	40	37.48%	35.01%	33.32%	61.42%
news	28	22.48%	23.74%	20.95%	19.94%
	32	28.55%	29.70%	27.78%	38.53%
	36	24.24%	24.21%	22.55%	58.3%
	40	15.76%	13.61%	12.16%	73.55%

5. CONCLUSION

In this paper, two all-zero block detection methods are proposed for 4×4 block and 8×8 block, respectively. It achieve feasibility of hardware implementation at cost of slight accuracy loss. Experimental results show that the proposed algorithm outperforms the previous methods in all cases and achieves major improvement of computation reduction in the range from 12.16% to 42.68% for 4×4 blocks, from 4.86% to 73.55% for 8×8 blocks compared to previous methods.

Acknowledgments

This research was supported by Ambient SoC Global COE Program of Waseda University of the Ministry of Education, Culture, Sports, Science and Technology, Japan. And it was also supported by CREST and CLUSTER project.

REFERENCES

[1] X. Zhou and Z. H. Yu and S. Y. Yu, "Method for detecting all-zero DCT coefficients ahead of discrete cosine transformation and quantization," *Electronics Letters*, vol. 34, pp. 1839–1840, Sept. 1998.

[2] L. A. Sousa, "General method for eliminating redundant computations in video coding," *Electronics Letters*, vol. 36, pp. 306–307, Feb. 2000.

[3] Y. H. Moon and G. Y. Kim and J. H. Kim, "An improved early detection algorithm for all-zero blocks in H.264 video encoding," *IEEE Transactions on CSVT*, vol. 15, pp. 1053–1057, Aug.2005.

[4] C.Y. Su, "An Enhanced Detection Algorithm for All-Zero Blocks in H.264 Video Coding," *IEEE Transactions on Consumer Electronics*, vol. 52, pp. 598–605, May. 2006.

[5] C.Y. Chen and S.Y. Chien and Y.W. Huang and T.C. Chen and T.C. Wang and L.G. Chen, "Analysis and Architecture Design of Variable Block-Size Motion Estimation for H.264/AVC," *IEEE Transactions on Circuits and Systems*, vol. 53, pp.578–593, Feb. 2006.

[6] S.Gordon, "Simplified Use of 8×8 Transforms," *document JVT-K028, Joint Video Team (JVT) of ISO/IEC MPEG ITU-T VCEG*, Mar. 2004.