AN IMPROVED ZERO-BLOCK MODE DECISION ALGORITHM FOR H.264/AVC

Yu-Ming Lee and Yinyi Lin
Department of Communication Engineering
National Central University, Taiwan 32054, R.O.C.

ABSTRACT

In the previous work we presented a zero-block mode decision algorithm for H.264/AVC to reduce the computation time, based upon the number of zero blocks of 4x4 DCT coefficients between the current macroblock (MB) and the collocated macroblock. In this paper we extend the previous work and provide a more sophisticated algorithm, which uses an early zero-block detection method instead of DCT/Q computation, and incorporates adequate decision methods into semi-stationary and non-stationary regions of a video sequence. The simulation results demonstrate that the proposed algorithm can achieve up to 20% of time saving on average compared to the original zero-block mode decision algorithm, while maintaining a high coding performance.

Index Terms**—**H.264/AVC, DCT, zero-block mode decision, zero-block decision, early zero-block detection.

1. INTRODUCTION

The newest international video coding standard is H.264/AVC, and recently it has been approved by ITU-T as recommendation H.264 and by ISO/IEC as international standard MPEG-4 part 10 advanced video coding (AVC) standard. The emerging H.264 video coding standard achieves significantly better performance in both PSNR and visual quality at the same bit rate compared with priori video coding standards. One important technique is the use of variable block-size motion estimation/compensation. In the H.264/AVC, interframe motion estimation is performed for 7 different block sizes (denoted as modes), varying among 16x16, 16x8, 8x16, 8x8, 8x4, 4x8, and 4x4. These modes form a two-level hierarchy inside a MB. The first level L1 includes modes of 16x16, 16x8, 8x16, while the second level L2 includes modes of 8x8, 8x4, 4x8, and 4x4.

The rate distortion optimization (RDO) technique is used to check all possible inter-modes and find the best coding result to obtain the highest coding efficiency. Motion estimation in intermode decision is the most important part of a typical video encoder in computation. The computational complexity of H.264/AVC is dramatically increased due to this optimization technique and variable block-size modes performed.

Many fast and efficient methods for intermode decisions have been proposed in recent years to reduce the computation cost and maintain coding performance. Some schemes make use of temporal and/or spatial correlations to classify a video object into stationary (as well as homogeneous) or non-stationary (non-homogeneous) areas [2]. Some other algorithms attempt to compose motion vector candidates for different block-size modes to reduce the computation using bottom-up merging algorithms [3]-[4].

In the previous work [1] we have presented a zero-block mode decision algorithm for H.264/AVC to alleviate the encoder complexity while maintaining picture quality. The zero-block decision algorithm in the previous work makes use of the number of zero-blocks of 4x4 DCT coefficients between the current macroblock and the collocated macroblock to characterize a video object. The experimental results revealed that a significant reduction in computation could be achieved, especially for low bit-rate coding. However, the time saving performance is limited in the high bit-rate coding. In this paper, we investigate a substantially improved algorithm to reduce the computational complexity of H.264 encoder, in which we uses an early zero-block detection method to compute the number of zero-block instead of direct DCT and quantization (DCT/Q) computation, and incorporates adequate decision methods into semi-stationary and non-stationary regions of a video sequence.

2. A NEARLY SUFFICIENT CONDITION FOR EARLY ZERO-BLOCK DETECTION

Success of the zero-block decision is achieved by discarding the least possible modes according to the number of zero-blocks of 4x4 DCT coefficients between the current macroblock and the collocated macroblock. The number of zero-blocks is determined using direct DCT/Q computation. It then takes use of the number of zero-blocks to classify a video sequence into stationary, nearly stationary, semi-stationary and non-stationary regions, and different modes are performed in theses areas to reduce the computation time.

The 4x4 DCT block is called a zero-block (ZB) if all 4x4 DCT coefficient are quantized to zero when all the quantized coefficients Z_{ij} satisfies $|Z_{ij}|<1$. The DCT and quantization is given by

$$Z_{ij} = (W_{ij} \cdot M(QP\%6; r) + f) >> qbits \qquad (1)$$

where W_{ij} is the core 2D transform, $r = 2 - (i\%2) - (j\%2)$, % denotes the modular operator and the multiplication factor $M(QP\%6;r)$, f is $2^{qbit}/6$ for inter blocks, and >> represents a binary shift right.

2.1 A sufficient condition in Moon [5] (denoted as Moon)

The sum of the absolute difference (SAD) SAD16x16 can be rewritten as

$$\begin{aligned} SAD_{16\times16} &= \sum_{i=0}^{15}\sum_{j=0}^{15}\left|MB_c(i,j)-MB_r(i,j)\right| \\ &= \sum_{l=0}^{3}\sum_{k=0}^{3}\left\{\sum_{i=0}^{3}\sum_{j=0}^{3}x^{l,k}(i,j)\right\} = \sum_{l=0}^{3}\sum_{k=0}^{3}X^{l,k} \end{aligned} \tag{2}$$

where $X^{l,k}$ is the sum of the absolute difference (SAD) between 4x4 blocks located at (*4l,4k*) in the MBs.

Moon et al. derived a precise sufficient condition for zero-block detection in H.264/AVC, which is summarized as follows.

(1) If $X^{l,k} \le T(0)$, then $X^{l,k}$ is a ZB, and where $T(0)=[\frac{5}{6}\cdot 2^{15+\lfloor QP/6\rfloor}]/[4\cdot M(QP\%6;0)]$.

(2) If $T(0) > X^{l,k}$ and $X^{l,k} \le \min\{T(0)+\gamma/2, T(1)\}$, then $X^{l,k}$ is also a ZB. The parameters $T(1)$ and γ are respectively given by
$T(1)=[\frac{5}{6}\cdot 2^{15+\lfloor QP/6\rfloor}]/[2\cdot M(QP\%6;1)]$,
and
$\gamma=\min\{\sum_{j=0}^{3}[x^{l,k}(0,j)+x^{l,k}(3,j)], \sum_{j=0}^{3}[x^{l,k}(1,j)+x^{l,k}(2,j)]]$

As can be seen, the condition varies with x_{ij}. An intensive study indicates that this sufficient condition varies in a range $2.2Q_{step} \sim 2.5Q_{step}$.

2.2 A nearly sufficient condition (denoted as 3.5Q)

The sufficient condition mentioned in [5] to determine whether 4x4 DCT is a zero-block is sufficient but not necessary. Although some zero-blocks of 4x4 DCT coefficients can be detected correctly using these sufficient conditions, there are still numerous zero-blocks remaining undetected, especially for high bit-rate coding or low quantization parameters (QPs). In this section, a nearly sufficient condition is derived based upon the ensemble average of all 4x4 DCT coefficients by summing up all 4x4 DCT coefficients Y_{ij}. The summation over all DCT coefficients can be written as

$$\begin{aligned} \sum_{i=0}^{3}\sum_{j=0}^{3}Y_{ij} &= C_{00}\cdot x_{00}+C_{01}\cdot x_{01}+C_{02}\cdot x_{02}+\cdots+C_{33}\cdot x_{33} \\ &= \sum_{i=0}^{3}\sum_{j=0}^{3}C_{ij}x_{ij} \end{aligned} \tag{3}$$

Define the ensemble average of all DCT coefficients as $Y_{av}=\frac{1}{16}\sum_{i=0}^{3}\sum_{j=0}^{3}Y_{ij}$ and $C_{\max}=\max\{|C_{00}|,|C_{01}|,\cdots|C_{33}|\}$.

By using the following inequality

$$\left|\sum XY\right| \le \sum|X|\cdot|Y| \le \max(|Y|)\cdot\sum|X| \tag{4}$$

The ensemble average $|Y_{av}|$ can be upper-bounded as follows:

$$\begin{aligned} |Y_{av}| &= \tfrac{1}{16}\left|\sum_{i=0}^{3}\sum_{j=0}^{3}Y_{ij}\right| \\ &\le \tfrac{C_{\max}}{16}[|x_{00}|+|x_{01}|+|x_{02}|+\cdots+|x_{33}|] \end{aligned} \tag{5}$$

or

$$|Y_{av}| \le \tfrac{C_{\max}}{16}X^{l,k} \tag{6}$$

After some manipulations, $C_{\max}$ is found to be 3.7975, and for inter block encoding, the DCT coefficient is quantized to zero when Z_{ij} satisfies $|Z_{ij}|<1$, i.e.,

$$|Z_{ij}| = |Y_{ij}|/Q_{step}+1/6<1 \tag{7}$$

or

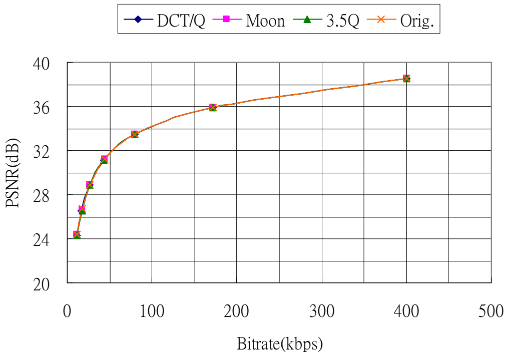
$$|Y_{ij}| < \tfrac{5}{6}Q_{step} \tag{8}$$

Instead of $|Y_{ij}|$, the ensemble average of DCT coefficients $|Y_{av}|$ is applied to (8) the following upper bound for zero-block detection can be obtained:

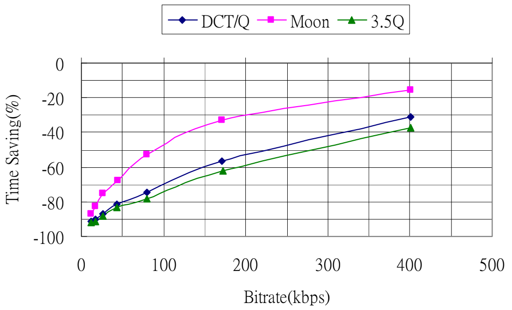
$$X^{l,k} < 3.5Q_{step} \tag{9}$$

2.3 Early zero-block detection instead of direct DCT/Q computation

The coding performance and time saving of zero-block mode decision algorithm using direct DCT/Q computation, Moon's condition and proposed nearly sufficient condition respectively are displayed Table 1 for *QP*=28. And simulation conditions are shown in Table 2. Fig. 1 shows the RD curve and corresponding saving compared to the original full search algorithm, carried on CONTAIN.CIF sequence for *QP*=28. As shown, all the criteria can achieve high video quality as the original algorithm. The zero-block decision algorithm using the nearly sufficient condition, however, can achieve the best performance in computation reduction. It saves about 8% of encoding time than the one using direct DCT/Q computation partly due to that it needs not to compute DCT/Q in determining the number of zero-blocks. The nearly sufficient condition is used for further studied in the proposed zero-block mode decision algorithm.



(a) RD curve



(b) Time saving

Fig.1 Performance comparison for “CONTAINER.CIF” (IPPP),

Table 1 Performance comparison for *QP*=28

(a) PSNR comparison

△PSNR(dB)	Sequence	DCT/Q	Moon	3.5Q
CIF	saleman	-0.046	-0.011	-0.057
	paris	-0.072	-0.024	-0.089
	container	-0.030	-0.009	-0.037
QCIF	foreman	-0.091	-0.042	-0.116
	grandma	-0.039	-0.004	-0.069
	mthr_dotr	-0.067	-0.041	-0.067
	Avg.	-0.057	-0.022	-0.072

(b) Bitrate comparison

△Rate(%)	Sequence	DCT/Q	Moon	3.5Q
CIF	saleman	0.84	0.05	0.85
	paris	1.59	0.82	1.92
	container	0.01	0.08	0.26
QCIF	foreman	1.05	0.30	1.12
	grandma	0.23	0.86	0.20
	mthr_dotr	0.89	0.82	1.35
	Avg.	0.77	0.49	0.95

(c) Time saving comparison

△Time(%)	Sequence	DCT/Q	Moon	3.5Q
CIF	saleman	-56.55	-27.01	-62.67
	paris	-49.18	-28.08	-53.57
	container	-56.81	-32.73	-62.37
QCIF	foreman	-42.37	-26.24	-45.41
	grandma	-62.75	-33.64	-67.57
	mthr_dotr	-47.39	-33.33	-51.78
	Avg.	-52.51	-30.17	-57.23

3. AN IMPROVED ZERO-BLOCK MODE DECISION USING A NEARLY SUFFICIENT CONDITION

In previous work [1], we employ the number of zero-blocks (N) to classify MBs into stationary, nearly stationary, semi-stationary and non-stationary MBs. The algorithm is summarized as follows:

(1) For N=16, it is a stationary MB, perform and choose SKIP mode (m_0) as the best.

(2) For $8 \le N \le 15$, it is a nearly stationary MB, perform SKIP and $m_1(16x16)$. If the RD cost of $m_1(16x16)$ is less than that of SKIP, then further perform modes $m_2(16x8)$ and $m_3(8x16)$. Choose the best mode.

(3) For $5 \le N \le 7$, it is a semi-stationary MB, perform SKIP, $m_1(16x16)$, $m_2(16x8)$ and $m_3(8x16)$.

(4) For $0 \le N \le 4$, it is a non-stationary MB, perform all seven modes as well as SKIP mode.

Note that step (1) is equivalent to an early skip termination. To improve the computation efficiency, in improved zero-block algorithm (as depicted in Fig. 2), we use the nearly sufficient condition to determine the number of zero-blocks, instead of direct DCT/Q computation. In addition, we incorporate splitting and merging algorithms into semi-stationary and non-stationary MBs instead of full search.

3.1 Splitting algorithm for semi-stationary MBs

In the top-down splitting algorithm, mode $m_1(16x16)$ is performed first, and the predicted motion vectors (PMVs) of other smaller block-size modes are obtained using the splitting rules. The PMV is used as the center of search point, and based on the initial point the full search refinement with ±2 pixels is performed to obtain a more accurate MV.

3.2 Merging algorithm for non-stationary MBs

In the bottom-up merging algorithm, mode $m_7(4\times4)$ is performed first, and the predicted motion vectors (PMVs) of other larger block-size modes are obtained using the merging procedure. The PMV is used as the center of search point, and based on the initial point the full search refinement with ±2 pixels is performed to obtain a more accurate MV. It was shown that more than 90% accuracy could be achieved with $\pm2-pel$ refinement. The merging algorithm is described as follows:

Step 1. Perform mode $m_7(4\times4)$ and find the estimated MVs.

Step 2. Predict MVs for larger block-size modes using the merging rules.

Step 3. Calculate the absolute difference Diff of PMVs with the same block-size modes.

Step 4. If $Diff \le \beta$, then perform $\pm2-pel$ refinement. Otherwise perform the full search estimation.

Note that for comparison purpose, the motion estimation for mode $m_7(4\times4)$ is performed using full search instead of fast search. The threshold β is a variable, and a larger β leads to a more computation reduction but with more severe degradation, and vice versa.

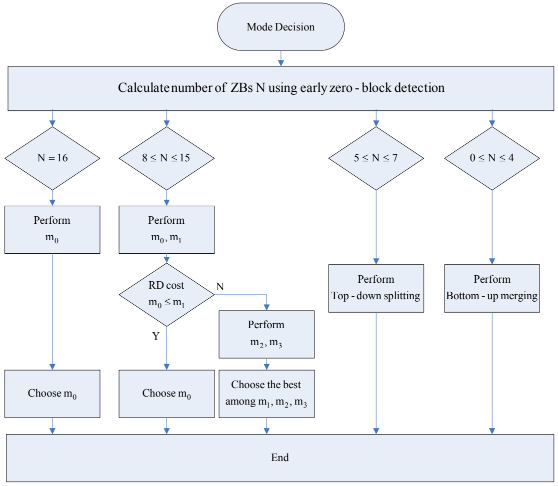


Fig. 2 Improved zero-block decision algorithm

4. EXPERIMENTAL RESULT

We implement our proposed algorithm in JM86 with the test conditions given as follows:

Table 2 Simulation conditions

Profile: Baseline	*Number of Frames: 100*
Reference Frames: 5	*Entropy Coding: UVLC*
RDO: on	*Hadamard: on*
Search Range: ±16	*QP: 24,28,32,36*

Six video sequences are tested: Foreman, Grandma and Mthr_dotr in QCIF format, and Salesman, Paris and Container in CIF format. The comparison of total encoding time, PSNR loss and bit-rate increment are displayed in Table 3 for *QP*=28. As shown, both algorithms can achieve a coding performance (PSNR and bit-rate) and the improved zero-block decision algorithm outperforms the original one in computation time. The result reveals that 20% in computation reduction on average can be obtained compared to the original zero-block decision algorithm. For comparison purpose, we also compare the rate-distortion curve and time saving of both algorithms for various *QPs*, conducted on the TEMPETE.CIF sequence. The result is depicted in Fig. 3.

5. CONCLUSION

In this paper we extended the previous work and presented an improved zero-block mode decision algorithm for H.264/AVC to reduce computation load. In the proposed algorithm, we employed a nearly sufficient condition to early determine the number of zero-block, instead of direct DCT/Q computation. In addition, we incorporated splitting and merging algorithms into semi-stationary and non-stationary MBs instead of full search. The simulation results reveal that the proposed algorithm can save another 20% of computation time on average compared to the original zero-block mode decision algorithm, while maintaining a high coding performance.

Table 3 Performance comparison for *QP*=28

(a) PSNR comparison

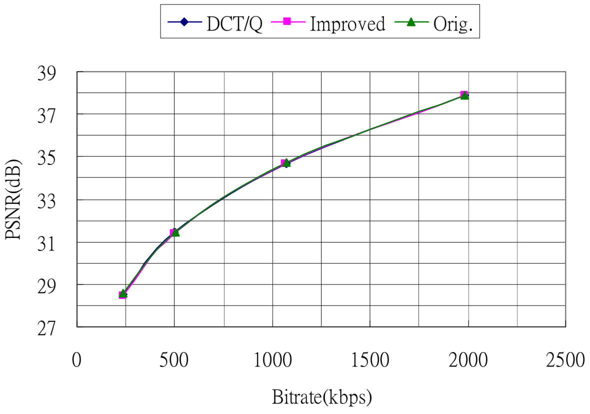
△PSNR(dB)	**Sequence**	**DCT/Q**	**Improved**
CIF	saleman	-0.046	-0.030
	paris	-0.072	-0.062
	container	-0.030	-0.063
QCIF	foreman	-0.091	-0.098
	grandma	-0.039	-0.036
	mthr_dotr	-0.067	-0.050
	Avg.	-0.057	-0.056

(b) Bitrate comparison

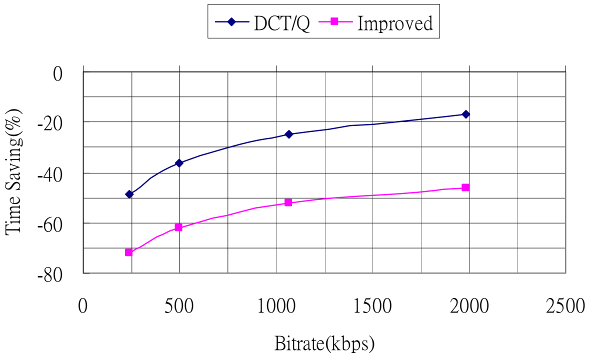
△Rate(%)	**Sequence**	**DCT/Q**	**Improved**
CIF	saleman	0.84	0.00
	paris	1.59	1.41
	container	0.01	0.84
QCIF	foreman	1.05	0.12
	grandma	0.23	0.33
	mthr_dotr	0.89	0.81
	Avg.	0.77	0.59

(c) Time saving comparison

△Time(%)	**Sequence**	**DCT/Q**	**Improved**
CIF	saleman	-56.55	-75.30
	paris	-49.18	-67.88
	container	-56.81	-78.60
QCIF	foreman	-42.37	-61.47
	grandma	-62.75	-79.05
	mthr_dotr	-47.39	-65.54
	Avg.	-52.51	-71.31



(a) RD curve



(b) Time saving

Fig. 3 Performance comparison for "TEMPETE.CIF" (IPPP)

6. REFERENCE

[1] Y.M. Lee and Y. Lin, "An Efficient Mode Decision for Low Bit Rate Coding in H.264/AVC," *IASTED European Conference on Internet and Multimedia Systems and Applications*, Chamonix, France, March 14-16, 2007.

[2] D. Wu, F. Pan, K. P. Lim, S. Wu, Z. G. Li, X. Lin, S. Rahardja and C. C. Ko, "Fast Intermode Decision in H.264/AVC Video Coding," *IEEE Transaction on Circuits and System for Video Technology*, vol.15, no. 6, pp. 953-958, July 2005.

[3] Z. Zhou, M.T. Sun, and Y.F. Hsu, "Fast Variable Block-Size Motion Estimation Based on Merge and Split procedures for H.264/MPEG-4 AVC," *Proceeding of IEEE International Symposium on Circuits and Systems*, vol.3, pp. 725-728, 2004.

[4] K. C. Hou, M. J. Chen and C. T. Hsu, "Fast Motion Estimation by Motion Vector Merging Procedure for H. 264," *Proceeding of IEEE International Conference on Multimedia & Expo*, July 2005, pp. 1444–1447.

[5] 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 Transaction on Circuits System and Video Technology*, vol. 15, pp. 1053–1057, Aug. 2005.