A PREDICTIVE BLOCK-SIZE MODE SELECTION FOR INTER FRAME IN H.264

Jiajun Bu, Shuiyong Lou, Chun Chen, Jingjing Zhu

College of Computer Science, Zhejiang University
310027, Hangzhou, P. R. China
{bjj, loushuiyong, chenc, zjj}@zju.edu.cn

ABSTRACT

One of the new features adopted in the latest H.264/AVC video coding standard is the mode decision (MD), which greatly improves the rate-distortion performance but it also takes up a significant encoding time to determine the best mode. This paper presents a predictive block-size selection algorithm to extraordinarily improve the encoding efficiency in H.264. The proposed algorithm mainly takes advantage of two efficient predictive methods: one is predictive skipping checking the sub-macroblock-level modes according to the sum of absolute difference (SAD) of each macroblock and the other is filtering some sub-macroblock-level modes with an adaptive threshold obtained from the mode information in the previous determined macroblock. Experimental results and comparative analysis are given to verify that our proposed algorithm can achieve a speed-up factor with 30% compared with the current Fast Full Search algorithm, with negligible average PSNR loss of 0.071dB and bit rate increase of 2.78%.

1. INTRODUCTION

The advanced H.264/AVC video coding standard which provides better compression of video images together with a range of features significantly outperforms the existing video coding standards in terms of both PSNR and visual quality [1]. These satisfactory performances mainly benefit from some new methods propounded by H.264, which assure the higher compression efficiency in the encoder of H.264. Among these techniques, MD module which is the subject of this paper is one of the essential parts of the whole encoder and achieves a better coding performance by reducing the temporal redundancy between frames. Different from previous video coding standards, the H.264 reference software [2] adopts rate distortion optimization (RDO) to evaluate the best possible combinations from seven block size modes (16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4) for every macroblock. They can be classified into two categories: macroblock-level block sizes including 16×16, 16×8 and 8×16 and sub-macroblock-level block sizes including 8×8, 8×4, 4×8 and 4×4. The general equation of Lagrangian RD optimization to determine the best mode is given as:

$$J_{mode} = SSD + \lambda_{mode} \bullet R \tag{1}$$

In the $equation.1$, J_{mode} is the rate-distortion cost ($RDcost$) and λ_{mode} is the Lagrangian multiplier; SSD means the distortion measurement between the original macroblock and the reconstructed macroblock, mode indicates the best mode selected from a set of possible modes and R denotes the number of bits associated with choosing the mode, including the bits for the macroblock header, the motion vector and all the DCT coefficients. Moreover the equation is also available for every sub-macroblock.

Recently many fast and efficient algorithms for inter block mode decision have been explored to reduce the computation burden of H.264. Some brilliant mode decision methods are proposed by utilizing the quad-tree structure of the seven block types. In [3], a merging procedure with an adaptive threshold is proposed to expedite the variable-block-size motion estimation (ME) process; In [4], a fast search algorithm is applied to the seven block-sizes independently; A merge and split procedure for ME by using the correlation of the Motion Vectors (MVs) of the different block-size modes is introduced in [5]; Different from the above discussed algorithms, some mode decision strategies are proposed to directly filter unlikely modes to reduce the searching process, such as [6] adopts a very efficient classification method which can reduce the average number of potential modes.

In this paper, we propose a predictive block-size selection approach based on two efficient predictive schemes to efficiently reduce the potential candidate modes. Firstly, all sub-macroblock-level modes will skip the checking process according to the SAD value of each macroblock. Secondly, predictive determination for stationary 8×8 blocks which can be directly acted as the best mode for sub-macroblock. Finally, filtering some sub-macroblock-level modes with an adaptive threshold obtained from the mode information in the determined macroblock. Compared with JM8.5, experimental results show that our proposed algorithm significantly reduce the computational complexity compared with the full search algorithm, with negligible coding efficiency degradation.

The rest of this paper is organized as follows. Section 2 describes our proposed algorithm in detail. Experimental results are shown in Section 3. Finally, we summarize the whole paper in section 4.

2. PROPOSED ALGORITHM

2.1. Predictive skipping scheme

RDO optimization strategy insures the best coding quality and meanwhile it also enormously increases the computational complexity of the encoder in JM. Hence, how to efficiently filter unlikely modes while maintaining the good coding performance is very essential and significant.

After testing a variety of video sequence, we detect that most macroblock-level block-size modes including 16×16, 16×8 and 8×16 are usually determined as the best mode for the macroblock with a small SAD value while for the macroblock with a large SAD value, all block-size modes need to be checked. Further investigations in-

The work was supported by National Natural Science Foundation of China (60573176) and Key Technologies R&D Program of Zhejiang Province (2004C11052 & 2005C23047).

dicate that the mode selection is related to the SAD value of the macroblock. So a threshold $T1$ needs to be selected to determine when the macroblock only requires the macroblock-level block sizes to act as the best potential candidate modes and to reduce the computational expenditure for searching every block-size mode.

In order to predetermine the value of threshold $T1$, three types sequences are tested including limited motion sequence, global motion sequence and the large distorted motion sequence. $Table1$ shows the experimental results, the upper table means the proportion of macroblock-level block sizes that are usually selected as the best modes based on the different SAD value of one macroblock and the lower table demonstrates the little feasibility that the sub-macroblock block size that can act as the best modes based on the different SAD value of one macroblock. It can also be seen that the accuracy can be pledged if SAD is betweent 800 and 1000; moreover, different sequences can lead to different value of $T1$.

Considering that the larger the value of $T1$ selected, the worse the accuracy obtained but the better encoding speed gained, we need select a better threshold $T1$ which assures both the good accuracy and the fast encoding speed. After testing many video sequences, we find that for most video situations, if QP=30 and the SAD value of one macroblock is less than 1000, i.e. $T1$=1000, the macroblock-level block sizes are usually selected as the best modes and we only need check 16×16, 16×8 and 8×16 block sizes. For different value of QP, the threshold $T1$ is different, we set 800 for QP=24, 1000 for QP=28, 32 and 1200 for QP=36, 40.

Table 1. The proportion of macroblock-level and sub-macroblock-level block sizes based on the SAD of one macroblock, QP=30.

Seq.	macroblock-level				
	800	**900**	**1000**	**1100**	**1200**
akyio	98.23%	94.78%	92.65%	91.37%	90.47%
news	96.34%	94.12%	92.69%	91.34%	89.02%
flower	95.26%	93.28%	91.21%	90.52%	87.36%
mobile	94.12%	93.36%	92.37%	89.09%	86.98%
foreman	92.98%	92.15%	91.63%	87.48%	83.65%
Seq.	sub-macroblock-level				
	800	**900**	**1000**	**1100**	**1200**
akyio	1.23%	1.32%	1.79%	6.41%	8.21%
news	1.57%	1.95%	1.21%	7.09%	9.32%
flower	1.94%	2.34%	1.84%	8.04%	11.66%
mobile	2.02%	2.43%	2.92%	8.26%	13.52%
foreman	3.36%	3.94%	4.53%	10.23%	15.21%

Experimental results ($Fig.1$) demonstrate that the predictive skipping scheme enormously decreases the computational complexity and meanwhile it succeeds in controlling PSNR decrease and bit rate increase.

2.2. Prediction for stationary 8×8 blocks

In H.264, seven block sizes are used in inter-frame motion estimation and blindly searching all modes causes an additional computation burden. However, we observe that in many video communication applications, such as video conference and video telephone, the video sequences mostly adopt the 16×16, 16×8 and 8×16 block sizes. Moreover, in the zooming, distorted large motion sequences, sub-macroblock-level block sizes including 8×8,

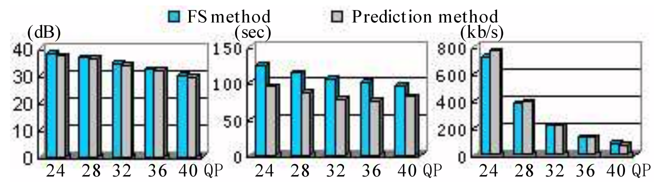

Fig. 1. The performance after using predictive skipping scheme with 50 frames of foreman sequence and extEPZS pattern

8×4, 4×8 and 4×4 also take few proportions. So if we proposed a method to predict 8×8 blocks is homogeneous or stationary which can skip the process of ME, computations on most of the low-layer blocks sizes are skipped and only the up-layer block sizes need to be checked, which can reduce the computational complexity by 40% compared with the original RDO strategy.

In the JVT reference software, the algorithm first adopts RDO strategy to evaluate the best block size from 16×16, 16×8 and 8×16 for a macroblock and gets the optimal $RDcost$. Then it divides the macroblock into four sub-macroblocks and uses RDO strategy to evaluate the best block size from 8×8, 8×4, 4×8 and 4×4 for every sub-macroblock. Finally it gains the minimal $RDcost$ of the best combinations of block sizes for the macroblock. Compare the two $RDcost$, the best block size can be determined with the less value of $RDcost$. So if the current 8×8 blocks can be predicted as stationary in advance, the sub-macroblock only uses the 8×8 block sizes and the searching process for other sub-macroblock-level block sizes can be skipped.

Further analysis demonstrates that these 8×8 blocks can be predicted as stationary blocks if the average SAD of these blocks is less than a predetermined threshold $T2$. We use two performance indexes including accuracy and amount to evaluate the best threshold value. It needs to be mentioned that accuracy performance means the ratio between the actual stationary 8×8 blocks and all predetermined 8×8 blocks while the amount performance denotes the total number of correctly predetermined 8×8 blocks. After experimental analysis with many test sequences, as $Fig.2$ shown, based on the two performance indexes, we can find that the best range of $T2$ to decide the stationary 8×8 blocks is between 150 and 250. Considering that the larger the value of $T2$ selected, the worse the coding quality obtained, we finally select $T2$ = 150 as the optimal value to predetermine the stationary 8×8 blocks and the $T2$ also can act as the best threshold for QP=24, 28, 32, 36 and 40 respectively.

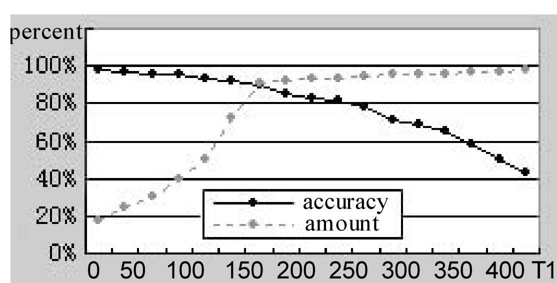

Fig. 2. The two performance based on different threshold to predetermine the stationary 8×8 blocks

Table 2. The proportion of the sub-macroblock-level block sizes obtained by Fast Full search, QP=30 and search range = [-16,15].

sequence	8×8	8×4	4×8	4×4
akyio	1.35%	0.31%	0.35%	0.21%
news	1.94%	0.88%	0.57%	0.32%
coastguard	3.64%	1.63%	1.53%	0.89%
carphone	3.95%	1.87%	1.89%	1.11%
foreman	4.88%	2.12%	2.21%	1.44%

2.3. Filter sub-macroblock-level block sizes based on an adaptive threshold

In most video sequences, especially in a large portion of homogeneous regions exists in video sequences. The selected sub-macroblock-level block sizes take up only about 7% in all block sizes but they influence on the visual quality more or less. It can be seen from the $Table 2$ gotten by testing various video sequences that the proportion of the sub-macroblock-level block sizes.

From the above statistical data, we conclude that the 8×8 block size usually takes the largest proportion among all sub-macroblock-level block sizes. In order to dwindle the best potential candidate block sizes thereby improve the efficiency of searching process, we calculate the $RDcost$ for the 8×8 blocks size in one sub-macroblock and take the $RDcost$ as a benchmark value. If the $RDcost$ for 8×8 block size is less than a threshold T', the searching process for a certain sub-macroblock-level block size can be skipped. How to select a precise threshold is very important. However, different sequences, even different frames within the same sequence lead to different thresholds. As a result, an adaptive threshold is introduced to handle this problem. It can be achieved as follows:

1. **Initialization:** $TRDcost$=0, N=0, where $TRDcost$ denotes the accumulated value of rate-distortion cost required for a certain block size which can be the 8×8, 4×8 or 4×4 block size in current frame and N represents the accumulated number of sub-macroblock used the block size.

2. **Repetition:** For every sub-macroblock, we first check the 8×8 block size and get the value of $RDcost$. If $RDcost < T'$, skip checking such block size. Else we need check the block size for current sub-macroblock and calculate the value of $RDcost$ required for the block size; then update T' by following steps.

$$\begin{aligned} TRDcost &= TRDcost + RDcost \\ N &= N + 1 \\ T' &= TRDcost/N \end{aligned} \tag{2}$$

It needs to be mentioned that the above equations are available for 8×4, 4×8 and 4×4 block sizes. Simply explained, T' is the average RDcost value of those sub-macroblocks which have already been determined as a certain block size in current frame.

Experimental results ($Fig.3$) demonstrate that the proposed algorithm efficiently decreases the computational cost by using prediction and filtering schemes respectively and meanwhile it also triumphantly controls the PSNR decrease and the bit rate increase.

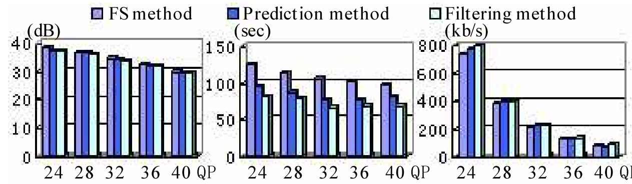

Fig. 3. The performance after using prediction and filtering method with 50 frames of foreman sequence and extEPZS

2.4. Search pattern selection

As for search patterns, we can use the Full Search (FS) or Diamond Search (DS) strategy [7] to get the motion vector (MV) for those decided macroblock. However, using FS strategy will incur great computational burden while for some relatively complicated sequences, DS strategy can be easily trapped in a local optimal position. To avoid such two limitations and to enhance the searching performance, a new search pattern named extend EPZS (extEPZS) [8] with better coverage is introduced.

2.5. Algorithm Summary

The predictive approach for block-size selection can be summarized as follows:

Step 1: Calculate the $SAD_{16\times16}$ value for current macroblock at zero-motion position and compare the $SAD_{16\times16}$ with $T1$. If $SAD_{16\times16} < T1$, check 16×16, 16×8 and 8×16 block-size modes; then go to step 5. Else go to step 2.

Step 2: Divide the current macroblock into four 8×8 blocks; then calculate the $SAD_{8\times8}$ for those sub-macroblocks at zero-motion position.

Step 3: For every sub-macroblock, compare the $SAD_{8\times8}$ with the threshold $T2$ respectively. If $SAD_{8\times8} < T2$, select 8×8 block-size mode for current sub-macroblock; then go to step4. Else check 8×8 block-size mode and calculate the rate-distortion cost ($RDcost$) for 8×8 block-size mode; then check condition C1.

- C1: If $RDcost < T'_{8\times4}$, check 8×4 block-size mode and update $T'_{8\times4}$ based on $eqn.2$; then check condition C2. Else check condition C2 (skip searching 8×4 process).
- C2: If $RDcost < T'_{4\times8}$, check 4×8 block-size mode and update $T'_{4\times8}$ based on $eqn.2$; then check condition C3. Else check condition C3 (skip searching 4×8 process).
- C3: If $RDcost < T'_{4\times4}$, check 4×4 block-size mode and update $T'_{4\times4}$ based on $eqn.2$; then go to step 4. Else go to step 4 (skip searching 4×4 process).

Step 4: Repeat step 3 to get the best mode for four sub-macroblocks; then get the best combination block-size mode for current macroblock. Check 16×16, 16×8, 8×16 and the combination block-size modes.

Step 5: Adopt the extEPZS method to choose the best mode among above checked modes.

3. EXPERIMENTAL RESULTS

We implemented the proposed algorithm for block-size mode decision into the reference JVT software version 8.5 (JM8.5). In the simulation, six video sequences with CIF format are tested and the frame rate is 30fps. In the experiments, the baseline profile encoder is used throughout all simulations and the quantization parameter (QP) has a range of 24, 28, 32, 36 and 40 respectively. Search range is set from -16 to 15 and only 1 reference frame is used. Moreover, the resolution and search range of the motion vectors are set to 1/4 pixel and 8 pixels. Three performance indexes including PSNR gain, bit rate gain and speedup gain, which follow the specification in [9] are adopted to evaluate the efficiency between our proposed mode decision algorithm and the Full Search (FS) algorithm in JM8.5. The experimental results are tabulated in $Table3$.

Table 3. Result of the proposed algorithm compared with Full Search algorithm in JM8.5.

sequence	QP	PSNR gain (dB)	bit rate gain (%)	speedup gain (%)
akyio	24	-0.11	1.38%	45.88%
	28	-0.06	1.92%	42.67%
	32	-0.04	1.90%	41.56%
	36	-0.05	1.23%	39.13%
	40	-0.06	2.79%	36.46%
news	24	-0.12	2.72%	38.21%
	28	-0.08	3.11%	37.32%
	32	-0.05	3.52%	32.45%
	36	-0.04	4.34%	31.76%
	40	-0.04	4.45%	30.90%
coastguard	24	-0.07	0.34%	19.66%
	28	-0.06	0.47%	18.98%
	32	-0.05	0.77%	19.55%
	36	-0.06	0.85%	20.47%
	40	-0.08	1.96%	22.68%
flower	24	-0.08	1.17%	20.00%
	28	-0.07	1.44%	20.82%
	32	-0.06	1.71%	21.46%
	36	-0.04	2.42%	22.57%
	40	-0.06	3.12%	24.09%
foreman	24	-0.13	4.80%	23.45%
	28	-0.11	4.21%	25.88%
	32	-0.10	5.14%	26.68%
	36	-0.09	6.45%	27.96%
	40	-0.07	7.17%	27.25%
average		**-0.071**	**2.78%**	**28.71%**

By contrast, the efficiency of the proposed algorithm more outperforms that of the JM8.5 algorithm in terms of three performance indexes. The experimental data also demonstrate that our proposed algorithm achieves an efficient tradeoff between complexity and quality compared with full search. For these limited motion sequences with static backgrounds such as "news" and "akyio", our proposed algorithm achieves about 37.63% speedup compared with FS algorithm, with 0.065 PSNR loss and 2.74% bit rate increase. For these global motion sequences, such as "flower" and "coastguard", our proposed algorithm also achieves 21.03% speedup with 0.063 PSNR loss and 1.23% bit rate increase. For "foreman" sequence with zooming, large distorted motion, our proposed algorithm achieves about 26.24% speedup compared with FS algorithm, with 0.100 PSNR loss and 5.55% bit rate increase. Therefore, it can be concluded that our proposed algorithm can adapt in varieties of motion scenarios, especially in some specific and limited motion situations, such as video surveillance and video telephone.

4. CONCLUSIONS

This paper presents a predictive block-size mode selection algorithm for H.264 video coding. By using two efficient and robust predictive methods, the algorithm significantly reduces the computational complexity at the cost of a slight degradation in the compression performance of the coding. Experimental results show that the proposed algorithm can achieve a good tradeoff between computational complexity and coding efficiency. The computational complexity can be reduced to about 70% of JM8.5 encoder with Fast Full Search algorithm, while the PSNR drop and the bit rate increase are negligible. Also, we believe that the performance of the proposed algorithm could be further improved by using some better predictive schemes. In the future work, more sophisticated strategies will be explored to filter more candidate modes to further decrease the complexity of encoder and multiple references will be applied in our algorithm.

5. REFERENCES

[1] P. List D. Marpe J. Ostermann, J. Bormans, "Video coding with h.264/avc: tools, performance, and complexity," *IEEE Circuits and Systems Magazine*, vol. 4, pp. 7–28, 2004.

[2] http://iphome.hhi.de/suehring/tml/download/, "Joint video team (jvt) reference software version jm8.5," .

[3] Y.N. Shen M.T. Sun Y.K. Tu, J.F. Yang, "Fast variable-size block motion estimation using merging procedure with an adaptive threshold," in *Proceedings of International Conference on ICME'03*, July 2003, vol. 2, pp. 789–792.

[4] G. Qiu K.K. Ma, "An improved adaptive rood pattern search for fast block-matching motion estimation in jvt/h.26l," in *Proc. IEEE International Symposium on Circuits and Systems(ISCAS) 2003*, Bangkok, Thailand, May 2003, pp. 708–711.

[5] Y.F. Hsu Zh. Zhou, M.T. Sun, "Fast variable block-size motion estimation algorithms based on merge and split procedures for h.264/mpeg-4 avc," in *In Proc. of ISCAS 2004*, May 2004, pp. 725–728.

[6] D.J. Wu S. Rahardja X. Lin K.P. Lim, S. Wu, "Fast inter mode selection," Tech. Rep., 9th Joint Video Team (JVT) Mtg, San Diego, CA, USA, Sep. 2003.

[7] S. Zhu K.K. Ma, "A new diamond search algorithm for fast block-matching motion estimation," *IEEE Trans. on Image Processing*, vol. 9, pp. 287–290, Feb. 2000.

[8] A.M. Tourapis H.Y. Cheong, "Fast motion estimation within the h.264 codec," in *in proceedings of 2003 IEEE International Conference on MultiMedia & Expo (ICME-2003)*, Baltimore, MD, July 6-9 2003.

[9] Gisle Bjontegaard, "Calculation of average psnr differences between rd-curves," in *VCEG-M33*, Austin, Texas, USA, April. 2001, 13th meeting.