A Novel Fast Inter Mode Decision Algorithm Based on Statistic and Adaptive Adjustment for H.264/AVC

Bin Zhan, Baochun Hou, Reza Sotudeh
University of Hertfordshire, School of Electronic, Communication and Electrical Engineering
Hatfield, Hertfordshire, AL10 9AB, United Kingdom
E-mail: {b.zhan, b.hou, r.sotudeh}@herts.ac.uk

Abstract: **The advanced video coding (AVC) standard H.264 achieves better video compression performance compared with the previous standards, and one of major features is variable block sizes based mode decision (MD). However, the computational complexity is one key challenge for the high efficient compression. Taking into consideration the statistical characteristics and the adaptive adjustment, a fast Inter MD algorithm is proposed in this paper. The proposed MD algorithm is effective, and it is easy to implement. The simulation results show that the new algorithm reduces 20% processing time on average. Mostly, the losses of video are around 0.02dB, and the increase of bitrate can be limited less than 1%.**

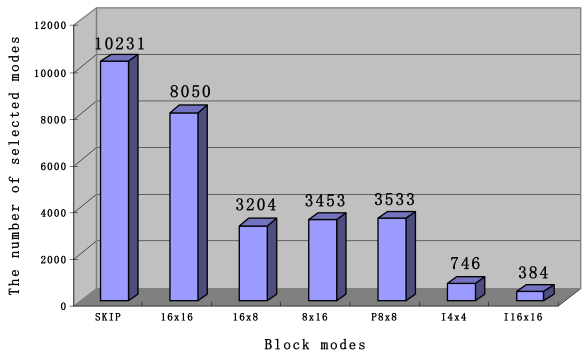
1. INTRODUCTION

Recently, the mobile video communications have received world-wide attention. The mobile video industry is keen looking for the solutions of low power consumption as well as high quality. There is no doubt, the quality of video can be further enhanced with unlimited power consumption. Unfortunately, the most of mobile based electronic products are battery-operational. Therefore, this is the trade-off between quality and system cost. Video coding standard takes the role of an commander to balance the system performance and the processing cost. H.264/AVC (also known as MPEG-4 part 10) is the latest video coding standard developed jointly by the ISO/IEC Moving Picture Experts Group (MPEG) and the ITU-T Video Coding Experts Group (VCEG) [1]. Compared with previous video coding standards, H.264 has introduced several complicated technologies such as smaller block partitions, quarter-pixel accurate motion compensation, multiple reference frames, in-the-loop deblocking filtering and small block-size transform [2, 3]. Variable block sizes based motion estimation (ME) with rate distortion optimization (RDO) enabled mode decision (MD) is one of key features of H.264. H.264 supports seven candidate modes: SKIP, Inter16×16, Inter16×8, Inter8×16, InterP8×8, INTRA4×4 and INTRA16×16. In the case of InterP8×8, each InterP8×8 mode can be further divided into 4 sub-macroblocks modes: Inter8×8, Inter8×4, Inter4×8, and Inter4×4. Multiple block sizes significantly enhance the video compression performance especially there are a large number of details in the video sequences. However, this excellent enhancement is achieved at the expenses of significant increasement of computational complexity. Total 21 MD operations: {SKIP MD, seven Inter MDs, INTRA4×4 MDs on 9 predicted directions and INTRA16×16 MDs on 4 predicted directions} consume about 50% compression time when RDO is enabled [4]. Long encoding processing time and huge power consumption make its difficulty for real-time and low power consumption applications such as video conference and mobile video communication. In order to reduce the computational complexity, several efforts [5, 6, 7, 8] have been made to simplify the mode selection by either classifying the modes into several groups or conjecturing from the selected best modes of neighboring blocks. In this paper, we proposed a fast Inter MD scheme based on statistic and adaptive adjustment.

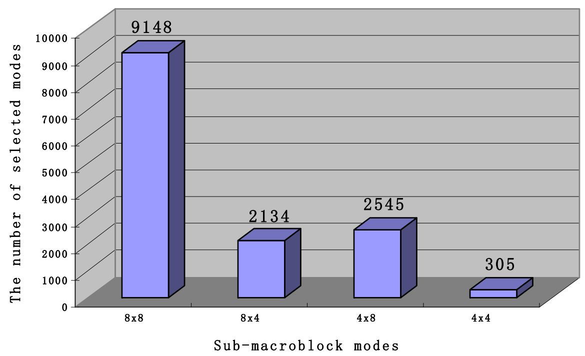
The rest of this paper is organized as follows: In Section 2 the proposed fast Inter MD algorithm is introduced. Simulation results are included in Section 3. In Section 4 the conclusions of this paper are given.

2. THE PROPOSED FAST MODE DECISION ALGORITHM

Video sequence normally possesses strong temporal similarity. Therefore, this feature can be employed in video encoder to remove the frame-to-frame redundancy by using block-based motion matching during Inter compression. Only residual and motion vectors (MV) will be delivered to the next encoding stage. There are seven variable block sizes used in ME: {16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4}. Although smaller block sizes mean more accurate block matching and less energy in the residual, more MVs also can increase the bitrate. The RDO plays the role of balancing the mode selection in order to get the most suitable encoding performance. However, the RDO enabled MD can choose the best mode only after all 41 possible ME results [9] for one macroblock have been obtained. If the system can prejudge which mode(s) could be abandoned during the early stage of encoding process at the minimum cost in terms of PSNR's degradation and increase of bitrate, the significant encoding time could be reduced drastically. It has been showed that most of video sequences have certain regulation inherently,



(a) Macroblock mode selection



(b) Sub-macroblock mode selection

Figure 1 - Mode selection for foreman_300_QCIF.yuv

which can help to reduce the number of candidate Inter modes after a few frames have been encoded.

Figure 1 shows an example of the best modes selected from 300 frames of Foreman QCIF video sequence. This is based on the JM H.264 software model recommended by the standard body using full search ME and optimized MD. It is also shown that the frequency of selected modes is very different, and this phenomenon was found more obviously for some sequences with low-speed motion or large static scene background. This characteristic of video sequences can be used to select a set of the most dominant modes. According to the above analysis, a fast MD algorithm was proposed. There are two stages in the algorithm. In Stage One, the Inter MD is carried out using normal MD procedure. The statistic information is collected from a pre-established amount of frames, and mode filtering will be processed at the end of this stage. Taking into account the actual application, there could be sudden change existing in the video sequence, employing selected modes in the first stage of proposed algorithm possibly leads to poor coding performance for the new scenes. In order to avoid this downgrade, an adaptive adjustment scheme of MD is necessary and is proposed as the second part of algorithm.

2.1 Steps of Proposed Algorithm

The algorithm is summarized as following steps:

Stage One: Statistical Learning Encoding Stage

STEP1: Encode the first N P-frames using all modes. Accumulate the frequencies of selected modes for Inter macroblock and Inter sub-macroblock modes, separately and frame by frame. Calculate and update the ratio (r_x) of each mode used in the first N P-frames as statistic frequency.

A: Inter MB	**B: Inter sub-MB**
r_2 - 16×8 r_3 - 8×16 r_8 - P8×8	r_4 - 8×8 r_5 - 8×4 r_6 - 4×8 r_7 - 4×4

STEP2: Setup a moving window to record the most recent 30 encoded frames' bitrate for bitrate adaptive adjustment.

Stage Two: Reduced Modes Encoding Stage

STEP3: If (r_8 < threshold λ_1)
{Disable sub-modes: 8×8, 8×4, 4×8, 4×4}
If r_x is less than λ_1
{Disable mode x }
where λ_1 is mode elimination threshold. Then a new list of modes will be used for the rest sequence.

STEP4: Encode next frame and monitor the undulation of bitrate.

STEP5: If the fluctuating of bitrate exceeds the threshold λ_2 for 5 consecutive frames, where λ_2 is the bitrate adjustment threshold, turn on all disabled modes and go back to STEP1. Otherwise, go back to STEP4 until the last frame of current sequence has been encoded.

2.2 Selection of Parameters

The parameters (thresholds λ_1, λ_2 and frame number N) employed in the algorithm directly affect the efficiency and the performance of encoder.

A). Inter modes elimination threshold λ_1

The mode elimination threshold λ_1 can be used to disable Inter modes with low-frequency. Too low λ_1 value could introduce unnecessary modes, and undermine improvements in encoding time; but higher value could disable necessary Inter modes and lead to significant degradation of quality of encoded sequence. It is found through our extensive experiments that λ_1 set at 5% yields better performance.

Table 1 - Proportion of Macroblock Mode

Sequence	SKIP	16×16	16×8	8×16	P8×8	I4×4	I16×16
foreman	23.19%	27.63%	11.49%	13.50%	21.43%	2.22%	0.54%
salesman	79.44%	12.20%	0.95%	1.26%	5.88%	0.27%	0.01%
carphone	36.96%	27.11%	7.47%	10.25%	15.57%	1.30%	1.34%
news	75.32%	14.25%	1.90%	2.51%	5.09%	0.81%	0.10%
bridge-far	96.74%	2.93%	0.07%	0.03%	0.04%	0.10%	0.08%
coastguard	12.90%	50.58%	9.79%	9.79%	13.73	2.87%	0.34%
highway	58.92%	17.30%	9.20%	5.76%	7.49%	0.55%	0.78%
mobile	1.67%	56.40%	6.12%	4.65%	30.66%	0.48%	0.01%

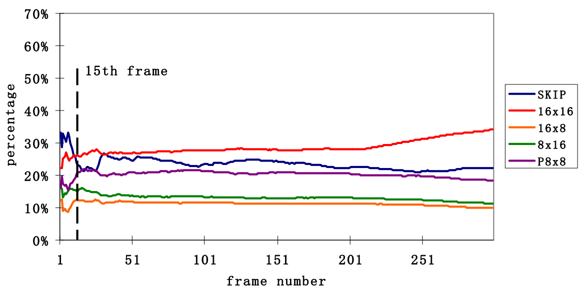
Table 2 - Proportion of Sub-Macroblock Mode

Sequence	8×8	8×4	4×8	4×4
foreman	62.00%	16.43%	18.78%	2.79%
salesman	61.84%	15.35%	17.06%	5.75%
carphone	63.04%	16.79%	16.69%	3.47%
news	61.59%	13.36%	21.31%	4.74%
bridge-far	87.50%	6.25%	6.25%	0%
coastguard	63.18%	14.55%	18.64%	3.64%
highway	62.69%	18.09%	15.41%	3.82%
mobile	60.39%	16.57%	16.71%	6.33%

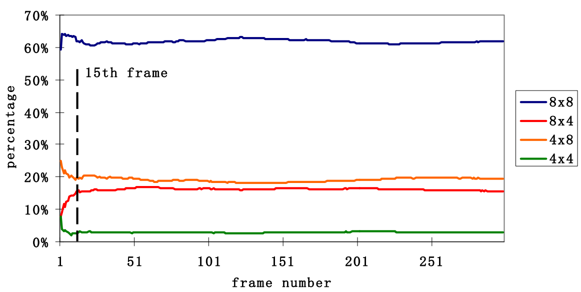
Table 1 and Table 2 list the frequencies of all modes for 8 different video sequences that proved that 5% is a suitable value for λ_1.

B). Learning period parameter N

N frames decide the length of the learning period for the statistical accumulation of distribution of inter modes. It is desirable that enough information is obtained in fewer frames; however shorter statistical learning period could

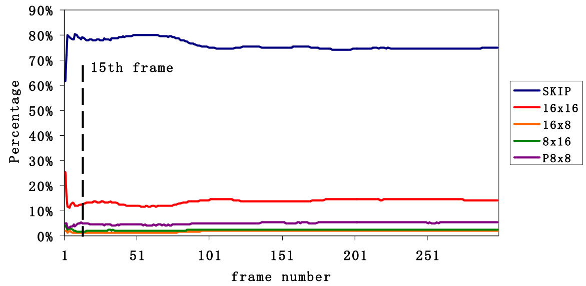


(a) Macroblock mode

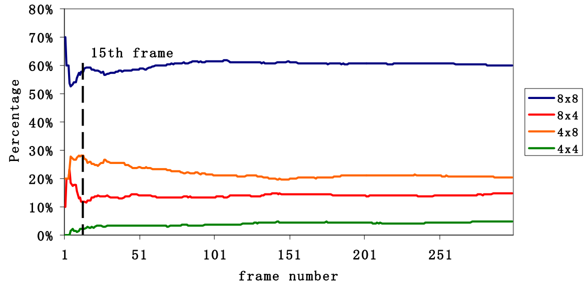


(b) Sub-macroblock mode

Figure 2 - Proportion of selected modes vs. frame number for foreman_300_QCIF.yuv



(a) Macroblock mode



(b) Sub-macroblock mode

Figure 3 - Proportion of selected modes vs. frame number for news_300_QCIF.yuv

result in excursion from the actual data. Our large amount simulations show that the system reaches relatively steady state after about 15 frames. Figure 2 and Figure 3 are two examples illustrating the distribution of each selected mode for Foreman and News QCIF sequences with total number of frames equaling to 300.

C). Adaptive adjustment threshold λ_2

The video contents are highly correlated in temporal, and they possess very similar details such as objects and background. However, some special cases, high-speed motion and sudden moving of camera, can not be ignored. Those abnormal factors could affect the coding performance such as increasing of output bits for each encoded frame. Therefore, an adaptive adjustment scheme based on the difference of output bitrate is proposed to adjust MD adaptively.

After I-frame has been coded, the output bit stream of each encoded P-frame is recorded as b_i for the i^{th} P-frame. The average output bits of every 30 frames denotes as X_Bit_n shown in (1). For example, X_Bit_1 is the average output bits for 30 P-frames between [1^{st}, 30^{th}], and X_Bit_2 represents the average output bits for 30 P-frames between [2^{nd}, 31^{st}]. This forms a moving window to average 30-frame output bits. The difference between X_Bit_n and X_Bit_{n-1} is denoted as d_n and is defined in (2). If the consecutive five (five frames can avoid inaccurate adjustment according to the casual changes of the video scenes) d_n values exceed threshold λ_2, there could exist sudden change in the video stream, and current MD scheme needs to be re-evaluated.

$$X\_Bit_n = \frac{b_n + b_{n-1} + \cdots + b_{n-29}}{30} \quad (1)$$

$$d_n = ABS(\frac{X\_Bit_n - X\_Bit_{n-1}}{X\_Bit_n}) \times 100\% \quad (2)$$

According to the simulation results, the threshold λ_2 has been set as 3%, and this can be used to effectively turn on the disabled modes.

3. SIMULATION RESULTS

Eight video sequences have been selected for simulations (foreman in QCIF, bridge-far in QCIF, coastguard in QCIF, news in QCIF, m_d in QCIF, mobile in QCIF, akiyo in CIF and hall in CIF), and each sequence contained 300 frames. Our fast MD scheme was integrated into reference software JM10.1 [10]. The baseline profile was chosen for comparison purpose. The search range was set to ±16. The sequence type used was IPPP. The Full Search scheme was employed in ME. One frame was used for ME reference. The simulations were carried out on PC with an AMD-processor 1.81 GHz and 256MB memory. The evaluation of system performance was focused on the change rate of encoding time, PSNR and output bitrate, i.e., △Time(%), △PSNR(dB) and △Bitrate(%). They were defined as the difference between original coding processing and proposed coding processing, and were shown in (3), (4) and (5):

$$\Delta Time = \frac{Time_B - Time_A}{Time_A} \times 100\% \quad (3)$$

$$\Delta PSNR = PSNR_B - PSNR_A \quad (4)$$

$$\Delta Bitrate = \frac{Bitrate_B - Bitrate_A}{Bitrate_A} \times 100\% \quad (5)$$

where A is based on the standard JM10.1 MD algorithm, and B is based on the proposed MD algorithm.

Table 3 lists the simulation results. It clearly shows that the proposed fast MD algorithm reduced encoding time up to 43.77%, and the decrease of PSNR and the increase of bitrate are very limited. Figure 4 and Figure 5 present the comparisons of PSNR values and output bit stream sizes between proposed algorithm and JM reference software model using Foreman sequence, respectively. The results are very comparable. Figure 6 gives an example that the proposed algorithm can adaptively adjust the mode selection when sudden change existing in the video sequence.

Table 3 - Threshold = 0.05 @ 15th frame

Sequence	ΔTime(%)	ΔPSNR	ΔBitrate(%)
QP = 28			
foreman	-9.04%	-0.02dB	0.11%
bridge-far	-43.77%	-0.02dB	0.54%
coastguard	-7.42%	0.01dB	0.03%
news	-22.81%	-0.02dB	1.58%
m_d	-19.84%	-0.06dB	0.55%
mobile	-7.89%	0.01dB	0.22%
akiyo (CIF)	-14.43%	0.01dB	0.69%
hall (CIF)	-21.71%	-0.03dB	1.63%
QP = 32			
foreman	-5.92%	0.01dB	-0.28%
bridge-far	-38.41%	0dB	0.00%
coastguard	-9.16%	-0.01dB	0.51%
news	-23.65%	-0.03dB	2.05%
m_d	-31.72%	-0.16dB	2.08%
mobile	-15.76%	-0.01dB	0.21%
akiyo (CIF)	-14.70%	-0.04dB	0.62%
hall (CIF)	-13.37%	0.01dB	0.82%

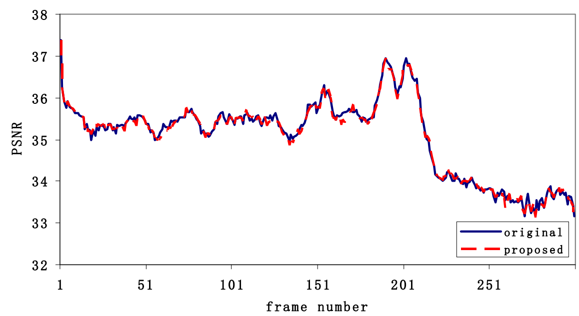


Figure 4 - Comparison of PSNR values between proposed and original algorithms for foreman.yuv

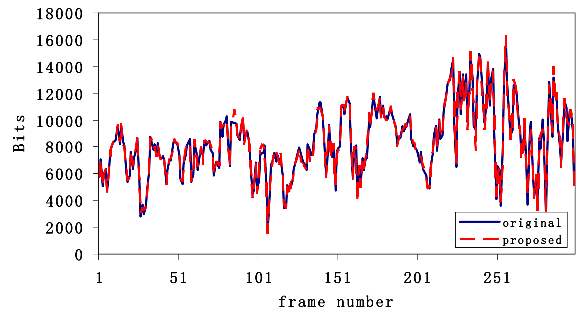


Figure 5 - Comparison of bit steam sizes between proposed and original algorithms for foreman.yuv

4. CONCLUSIONS

In this paper, a new fast Inter MD algorithm based on statistic and adaptive adjustment has been proposed. The algorithm can reduce the computational time significantly with slight loss of PSNR and Bitrate increase. Moreover, the proposed algorithm does not require complex computation

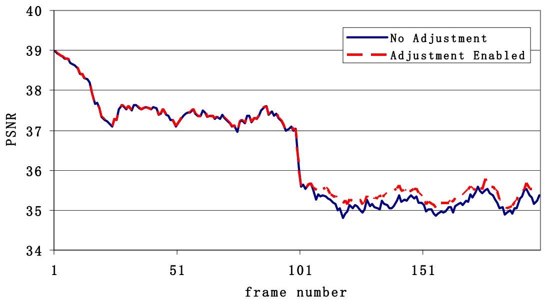


Figure 6 - PSNR vs. frame number as adaptive adjustment is enabled and disabled, respectively

and large hardware cost such as large memory to buffer a lot of video information. Therefore, it is very attractive to real-time and low power consumption applications.

REFERENCES

[1] "Draft ITU-T Rec. and FDIS of Joint Video Spec. (H.264 | ISO/IEC 14496-10 AVC)," JVT of MPEG and VCEG, Doc. JVT-G050rl, May 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," IEEE Trans. Circuits and Systems for Video Technology, vol.13, no.7, pp. 560-576, July 2003.

[3] I. E. G. Richardson, "H.264 and MPEG-4 Video Compression, Video Coding for Next-generation Multimedia," John Wiley &Sons, England, September 2003.

[4] B. Jeon and J. Lee, "Fast mode decision for H.264," ITU-T Q.6/16, Doc. #JVT-J033, 2003.

[5] A. C. Yu, "Efficient block-size selection algorithm for interframe coding in h.264/mpeg-4 avc," Proc. IEEE ICASSP, vol.3, pp. 169-172, May 2004.

[6] H. Zhu, C. K. Wu, W. Y. Li and Y. Fang, "Fast mode decision for H.264/AVC based on macroblock correlation," Proc. AINA'05. 19th International Conference, vol.1, pp.775-780, March 2005.

[7] K. H. Han and Y. L. Lee, "Fast macroblock mode determination to reduce H.264 complexity," IEICE Trans. Fundamentals, vol.E88-A, no.3, pp. 800-804, March 2005.

[8] C. Grecos and M. Y. Yang, "Fast inter mode prediction for P slices in the H264 video coding standard," IEEE Trans. Broadcasting, vol.51, no.2, pp. 256-263, June 2005.

[9] S. Y. Yap and J. V. McCanny, "A VLSI architecture for variable block size video motion estimation," IEEE Trans. Circuits and Systems-II: Express Briefs, vol.51, no.7, pp. 384-389, July 2004.

[10] JVT reference software unofficial version JM10.1, http://iphome.hhi.de/suehring/tml/download/jm10.1.zip.