MVA2009 IAPR Conference on Machine Vision Applications, May 20-22, 2009, Yokohama, JAPAN

Inter Mode Decision Algorithm For Advanced Video Coding

Fang-Hsuan Cheng
Department of Computer Science & Information Engineering, Chunag Hua University
Hsinchu, Taiwan 300
fhcheng@chu.edu.tw

Yea-Shuan Huang
Department of Computer Science & Information Engineering, Chunag Hua University
Hsinchu, Taiwan 300

Abstract

Variable block size used for inter coding is one of the key technologies in H.264/AVC. When different objects contained in the same macroblock have different motions, smaller block sizes probably achieve better predictions. However, this feature results in extremely high computational complexity when all the block sizes are considered to decide a best one. This paper proposes a new inter mode decision algorithm to reduce the number of inter modes that has to be checked, and then encoding time is reduced. We use the co-located macroblock in previous frame and its neighbors as candidates, and check whether an edge of moving object is crossing the middle of these candidates by using the score given to the modes. The experimental results show that the proposed algorithm is able to reduce 31%-41% total encoding time and about 41%-54% motion estimation time with a negligible PSNR loss of 0.05 dB and bit-rate increment of 2% on the average.

Keyword: Variable Block Size; Motion Estimation; Mode Decision

1. Introduction

Video Compression plays an important role in digital video communication, transmission and storage. H.264/AVC [1-4] is the latest video coding standard developed by the JVT (Joint Video Team) of ISO/IEC Moving Picture Experts Group (MPEG) and ITU-T Video Coding Expert Group (VCEG). While the H.264 belongs to H.26L family of VCEG and the AVC (Advanced Video Coding) belongs to MPEG-4 part 10. This standard has been designed in order to provide higher coding efficiency and network adaptation, which includes a Video Coding Layer (VLC) and a Network Abstraction Layer (NAL). While the VCL represents the video content, and the NAL provides a network-friendly interface.

Comparing to the previous video coding standards, H.264/AVC achieves significant improvement in coding efficiency. This is due to the fact that a number of new techniques are adopted in this standard such as variable block size (VBS) motion estimation, multiple reference frames, quarter-pixel motion estimation, directional prediction of intra coded blocks, in-loop deblocking filter, integer DCT transform and context-based adaptive binary arithmetic coding (CABAC) etc.. As a result, H.264 can save over half bitrate compared with that of MPEG-2 under the same quality.

Motion estimation (ME) is used as a main method for removing the redundantly information between frames in many video coding standards. H.264, like other video encoders, adopts block-based motion estimation to find a best block matching from a pre-defined search area, and performs variable block size motion estimation to indicate individual motion object in a macroblock. Figure 1 shows the seven block sizes and corresponding mode number/symbol in H.264. We can divide these seven block sizes into two levels which are macroblock level and sub-macroblock level. In macroblock level, there are four inter modes and an additional skip mode (mode 0) which uses the same size with mode 1. If the macroblock is processed in sub-macroblock level, it can be further partitioned into 8x8, 8x4, 4x8 and 4x4 block sizes. The same works will be done in the four sub-macroblocks, and the order of process is from left to right and top to bottom even in mode 2 or 3.

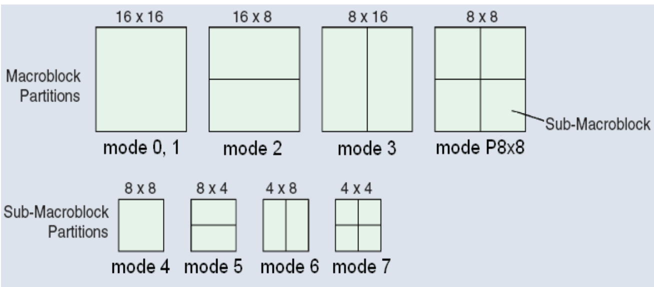


Figure 1. Variable block sizes and corresponding mode number.

According to the new technique described above, H.264/AVC achieves higher coding efficiency than prior video coding standards. However, the large amount of computation makes the encode time extremely increase, thus, it is difficult to be used in practical applications especially in real-time environment. It can be seen that inter modes still take the biggest part of computation. For the reason, we propose a new inter mode decision algorithm to reduce the encoding time with negligible loss of coding efficiency.

The rest of this paper is organized as follows. Section 2 introduces some related works of inter mode decision in H.264. The proposed new inter mode decision algorithm is described in Section 3. The experimental results are shown in Section 4. And a conclusion will be given in Section 5.

2. Mode Decision in Video Coding

There are a number of fast mode decision algorithms proposed for H.264/AVC video coding [5-15] recently. In [7], they select five neighboring macroblocks as candidates by using the concept of 3DRS motion estimation algorithm which is proposed in prior standard. The modes and motion vectors belong to these candidates are used for checking the best prediction in current macroblock. By calculating the rate/distortion cost, the candidate with minimum cost will be chosen as the prediction result.

Reference [6] identifies a macroblock into two types. One is called stationary block and the other is homogeneous block. Stationary block refers to the "stillness" between consecutive frames in the temporal dimension. A region is homogeneous if the textures in the region have similar spatial property. This kind of blocks in the picture would have similar motion and are very seldom split into smaller blocks. The way of determining homogeneous block is to use edge information. An edge map is created for each frame using Sobel operator. Then, they calculate the amplitude of edge by summing up the edges of x and y direction. Finally, they check whether it is a homogeneous block according to the amplitude.

3. Proposed Method

Taking the properties of the seven block sizes into consideration, we can observe that small block sizes usually appear at the edge of motion object. This is due to the size of macroblock is fixed to 16x16 and the context may not just belong to one object, for instance, an object and the background are included in one macroblock. Furthermore, if the motions of objects are more complex and half partitioning (mode 2 or mode 3) a macroblock is not enough, small block sizes will achieve better prediction by partitioning the macroblock into four 8x8 sub-macroblock.

Similar to the idea of the method [7] described in Section 2, a region with complex textures and the motion is not stillness, should be partitioned into small block size. That is, a macroblock, which belongs to the edge of a moving object, is more likely to be partitioned using small block size. Although the mode may be chosen to mode 2 or mode 3 when the edges of object just fall into one side of the macroblock (i.e. one of the two blocks in mode 2 or mode 3), it is not often occurred because the shape of objects belonged to the nature is usually irregular. It means that if a macroblock can be identified that it contains the edge of a moving object, we can determine that this macroblock should be partitioned into small block size (i.e. P8x8 mode).

According to the property of a common video sequence, there is less change between two frames. The co-located macroblock in previous frame has the highest correlation, so it can be used to predict the mode for current macroblock. For the enhancement of mode prediction, not only the co-located macroblock but also the neighbors of that are taken as the candidates. Referring to figure 2, the macroblock named 'C' is the co-located one in previous frame, and the eight neighbors surrounded C are also chosen as candidates. For convenience, we assign number 1 to 9 for each of the nine candidates from top left to bottom right. In order to find the edge, we use the four patterns to check whether the 'C' is at the edge of moving object. The four patterns are shown in figure 2. We collect the three macroblocks, which belong to each pattern, as a MB set, and the middle of them definitely are 'C'. That is, we can obtain four MB sets, and each one indicates its own direction.

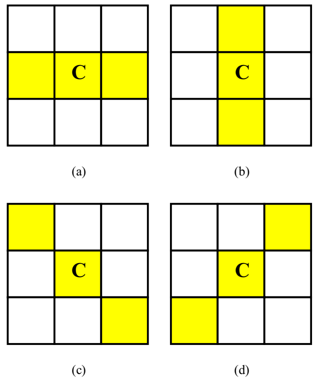


Figure 2. The four directional MB sets.

Considering (a) in figure 2, macroblock '4', '5' and '6' are collected to a MB set in horizontal direction. It means that we can check whether 'C' is crossing a horizontal edge of moving object by using this MB set. If so, current macroblock is probably at the edge. (b), (c) and (d) have the same meanings in vertical, diagonal from top left to bottom right and diagonal from top right to bottom left respectively.

To find out which MB set may be the edge, we assign a score for each mode first. Then, sum up the total value by using the score of modes corresponding to the three neighbors in each MB set and put it into a variable called "center_value". Each MB set has its own center_value. Finally, choose the maximum one from the four center_value and check whether the center_value is greater than a threshold. If so, it can be thought as an edge of object, and smaller block sizes such as P8x8 mode is used for the current macroblock. At this time, large block sizes such as mode 1, 2 and 3 are disabled. Table I shows the scores corresponding to each mode. And we can observe that more P8x8 macroblock appear in the neighbors, the probability of choosing the current macroblock to P8x8 is higher. Furthermore, if P8x8 neighbors concentrate in one of the four MB sets, there must be one moving edge crossing this direction.

On the other hand, if the maximum center_value is small enough, that is, all the neighbors contain large block sizes especially skip mode, it can be thought as the background and more modes can be disabled. Another threshold is also used for this checking.

Table 1. Modes and Corresponding Scores

mode	score
0	0
1	1
2	2
3	2
P8x8	4
9 (intra)	4
10 (intra)	4

The two thresholds are called ***upper_thd*** and ***lower_thd***. The former is used to divide all the inter mode into large block sizes and small block size, which the small one is P8x8 mode and the large ones are 16x16, 16x8 and 8x16. The latter is used to divide the 16x16 mode from the three ones. The algorithm is shown below:

if **center_value** > *upper_thd*
disable mode 1, 2 and 3
else if **center_value** < *lower_thd*
disable mode 2, 3 and P8x8
else
disable mode P8x8

According to experimental results, the upper value is set to 8 and the lower value is set to 2 will obtain the best performance. It means that there must be more than two P8x8 macroblock in the MB set which has the maximum center_value, and there must be at most one 16x16 macroblock in the MB set respectively.

Here we discuss all the possible combinations of the three neighbors that belong to the same MB set. If the center_value is greater than upper_thd, there are three situations which are 9, 10 and 12. Note that the value of center_value is impossible to be 11 under this score of modes. When 9, there must be two P8x8 (or Intra) macroblocks and one 16x16 macroblock. When 10, there must be two P8x8 (or Intra) and one 16x8 or 8x16, because the scores of mode 2 or mode 3 are the same. When 12, there must be three P8x8 (or Intra). On the other hand, if the center_value is smaller than lower_thd, there are two situations which are 0 and 1. When 0, it means that all the three neighbors must be skip mode. When 1, only one 16x16 macroblock in the MB set is allowed.

4. Experimental Results

The proposed mode decision method is implemented in H.264 reference software JM12.2 [16]. We select seven QCIF sequences which are commonly used for video compression testing in our experiments. The form of QCIF is 176 pixels in width and 144 pixels in height, that is, there are eleven macroblocks in width and nine macroblocks in height, and totally ninety nine macroblocks contained in one frame. The first 100 frames of every sequence are tested by our method. The GOP (Group of Picture) structure is IPPP, i.e. the first picture is coded as I-picture and remaining pictures are coded as P-pictures. The Full Search algorithm is used in all experiments with a ±16 pixels search window.

Firstly, total encoding time and total motion estimation time derived from the JM reference software for each sequence are listed in Table II. The total encoding time is counted all the needed time from the first frame to the last one for encoding, and the total motion estimation time is just only counted the time performing motion search in P-frame. The main difference of them is the time performing the reconstruction of macroblock for mode decision which includes inter modes and intra modes. From the table, we can observe that the time of motion estimation is about 50%-60% of total encoding time and not the 80% as we known in the prior standard. This may be caused by the amount of intra modes in H.264 and the improvement of Full Search.

In the second experiment, we inspect the PSNR change, bit rate change in percentage and time change in percentage. The experimental results are shown in Table III. The positive values mean increments whereas negative values mean decrements. In the time change, we list both total encoding time and motion estimation time. For the comparison, we also implement the Wu's method [6] and list the results in Table III. These changes are calculated by the following equations.

$$\Delta PSNR(dB) = PSNR_{proposed} - PSNR_{reference}$$
$$\Delta Bitrate(\%) = \frac{Bitrate_{proposed} - Bitrate_{reference}}{Bitrate_{reference}} \times 100\% \quad (6)$$
$$\Delta Time(\%) = \frac{Time_{proposed} - Time_{reference}}{Time_{reference}} \times 100\%$$

Table 2. Total Encoding Time and ME Time

Time / Sequences	Total Encoding Time (sec)	Total ME Time (sec)
Akiyo	127.686	52.030
Coastguard	207.802	119.152
Container	153.245	76.112
Foreman	169.844	89.693
M & D	138.190	64.234
Silent	153.073	73.527
Suzie	156.942	86.407

We can observe that the time saving of total encoding time in our proposed method exceeds 30% in all sequences with negligible quality loss and bit rate increase. The increase of bit rate in Akiyo is a little high because the motion of this sequence is too small. In fact, at the beginning of this

Table 3. Changes of PSNR, Bitrate and Time

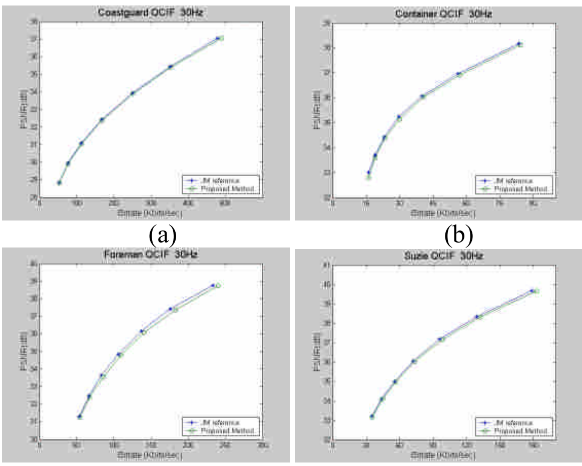
sequences	Wu's				Proposed			
	ΔPSNR (dB)	ΔBitrate (%)	ΔTotal Encoding Time (%)	ΔME Time(%)	ΔPSNR (dB)	ΔBitrate (%)	ΔTotal Encoding Time (%)	ΔME Time(%)
Akiyo	-0.02	0.71	-24.73	-32.80	-0.07	2.17	-36.22	-49.81
Coastguard	-0.02	1.28	-26.82	-36.01	-0.03	1.13	-33.59	-44.57
Container	-0.01	1.44	-20.69	-21.09	-0.05	1.46	-41.36	-54.14
Foreman	-0.09	2.14	-27.13	-33.31	-0.08	2.84	-31.39	-41.32
M & D	-0.08	0.75	-37.90	-48.35	-0.08	0.46	-36.41	-47.30
Silent	-0.06	2.46	-26.36	-37.23	-0.04	4.14	-36.53	-50.82
Suzie	-0.04	2.18	-35.89	-47.37	-0.01	2.75	-33.71	-43.87

sequence, Akiyo did not move last a while time. As a result, the motion is difficult to detect. Otherwise in Silent, small-sized blocks are concentrated on her arms because they move in a very high speed. But it is hard to distinguish inside or outside from her arms because there is about only one macroblock in width of that.

In the third experiment, we calculate the average processed modes per macroblock and show it in Table IV. If one of macroblocks is decided as P8x8 mode by the upper threshold, it performs four modes, else three ones will be performed. On the other hand, if the lower threshold works, only one mode will be performed. Figure 3 shows RD curves of four video sequences.

Table 4 Average Processed Modes per Macroblock

Sequence	modes
Akiyo	2.01
Coastguard	3.07
Container	1.89
Foreman	3.13
M & D	2.38
Silent	2.54
Suzie	3.03



(a) (b) (c) (d)

Figure 3. RD curves. (a)Akiyo (b) Container (c)Foreman (d)Suzie

5. Conclusion

This paper presented a new inter mode decision algorithm for video coding. Because the less change between frames, the co-located macroblocks with its neighbors in previous frame are used for predicting the mode of the current macroblock. This fast inter mode decision algorithm uses at most 3 modes as the candidates for checking the best one, and reduces 31%-41% of total encoding time with negligible quality loss about 0.05 dB and bit-rate increase about 2% on average. And the additional memory space used for storing the mode number could be less as possible that it just only enough to denote the mode of each macroblock.

Acknowledgement

We would like to thank National Science Council of Taiwan for her partial support of this research with grand no. NSC96-2221-E-216-039-MY2.

References

[1] J. Ostermann, J. Bormans, P. List, D. Marpe, M. Narroschke, F. Pereira, T. Stockhammer, T. Wedi, “Video coding with H.264/AVC: Tools, Performance, and Complexity,” IEEE Circuits and Systems Magazine, vol. 4, no. 1, pp. 7-28, First Quarter 2004.

[2] Gary J. Sullivan, and Thomas Wiegand, “Video Compression – From Concepts to the H.264/AVC Standard,” Proceedings of the IEEE, vol. 93, no. 1, pp. 18-31, Jan. 2005.

[3] D. Marpe, T. Wiegand, Gary J. Sullivan, “The H.264/MPEG-4 Advanced Video Coding Standard and its Applications,” IEEE Communications Magazine, vol. 44, no. 8, pp. 134-143, Aug. 2006.

[4] Iain E. G. Richardson, “H.264/MPEG-4 Part 10,” in H.264 and MPEG-4 Video Compression: Video Coding for Next-generation Multimedia, New York: John Wiley & Sons, 2003, ch. 6, pp.159-223.

[5] Libo Yang, Keman Yu, Jiang Li, and Shipeng Li, “An Effective Variable Block-Size Early Termination Algorithm for H.264 Video Coding,” IEEE Transactions on Circuits and Systems for Video Technology, vol. 15, no. 6, pp. 784-788, Jun. 2005.

[6] 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 Transactions on Circuits and Systems for Video Technology, vol. 15, no. 6, pp. 953-958, Jul. 2005.

[7] N.A. Khan, S. Masud, A. Ahmad, “A Variable Block Size Motion Estimation Algorithm for Real-time H.264 Video Encoding,” Signal Processing: Image Communication, vol. 21, no. 4, pp. 306-315, Apr. 2006.

[8] Christos Grecos, Ming Yuan Yang, “Fast Inter Mode Prediction for P Slices in the H.264 Video Coding Standard,” IEEE Transactions on Broadcasting, vol. 51, no. 2, pp. 256-263, Jun. 2005.

[9] Jongmin You, Wonkyun Kim, and Jechang Jeong, “16x16 Macroblock Partition Size Prediction for H.264 P Slices,”, IEEE Transactions on Consumer Electronics, vol. 52, no. 4, pp. 1377-1383, Nov. 2006.

[10] Bin Feng, Guang-Xi Zhu, Wen-Yu Liu, “Fast Adaptive Inter-Prediction Mode Decision Method for H.264 Based on Spatial Correlation,” IEEE International Symposium on Circuits and Systems 2006, pp. 21-24, May 2006.

[11] Donghyung Kim, and Jechang Jeong, “A Fast Mode Selection Algorithm in H.264 Video Coding,” IEEE International Conference on Multimedia and Expo 2006, pp. 1709-1712, Jul. 2006.

[12] Yun Cheng, Silian Xie, Jianjun Guo, Zhiying Wang, Minlian Xiao, “A Fast Inter Mode Selection Algorithm for H.264,” 2006 1st International Symposium on Pervasive Computing and Applications, pp. 821-824, Aug. 2006.

[13] Zhenyu Wei, and King Ngi Ngan, “A Fast Macroblock Mode Decision Algorithm for H.264,” IEEE Asia Pacific Conference on Circuits and Systems 2006, pp. 772-775, Dec. 2006.

[14] Chih-Hung Kuo, Meiyin Shen, and C.-C. Jay Kuo, “ Fast Motion Search with Efficient Inter-prediction Mode Decision for H.264,” Journal of Visual Communication & Image Representation, vol. 17, pp. 217-242, Jul. 2005.

[15] Zhi Zhou, Jun Xin, Ming-Ting Sun, “Fast Motion Estimation and Inter-mode Decision for H.264/MPEG-4 AVC Encoding,” Journal of Visual Communication & Image Representation, vol. 17, pp. 243-263, Jul. 2005.

[16] H.264/AVC Reference Software JM version 12.2, http://iphome.hhi.de/suehring/tml/download/old_jm/