FAST MACROBLOCK INTER MODE DECISION AND MOTION ESTIMATION FOR H.264/MPEG-4 AVC

Zhi Zhou and Ming-Ting Sun
Department of Electrical Engineering, University of Washington
Seattle, Washington 98195, U.S.A.

ABSTRACT

In H.264/MPEG-4 AVC, MacroBlock (MB) mode decision and motion estimation (ME) is one of the most computationally expensive processes. This paper proposes a fast inter-mode decision and ME algorithm. Experimental results show that the proposed algorithm could achieve similar Rate-Distortion (R-D) performance with about 50% computation saving, compared to the JM low-complexity mode with a Fast Full-Search (FFS) Motion Estimation algorithm.

1. INTRODUCTION

In the H.264 video coding standard (MPEG-4 part-10), variable block-size MB modes are introduced to improve the coding efficiency. Four modes (16*16, 16*8, 8*16, and 8*8) are available for the MB level, and four modes (8*8, 8*4, 4*8, and 4*4) for the sub-MB level. In the JM high-complexity mode [1], an exhaustive rate-distortion comparison for all block-sizes is performed to choose the best one. For each block-size, the bit-rate and the distortion are calculated by actually encoding and decoding the video. Therefore, the encoder can achieve the best R-D performance, at the cost of calculation complexity. For the high-complexity mode, the calculation of ME and Mode Decision can consume as much as 66% of the whole encoding process [8]. In the JM low-complexity mode, the cost function used in the ME process is also used for the mode decision. The cost function is defined as:

$$J(\mathbf{m},\lambda_{MOTION}) = SA(T)D(s,c(\mathbf{m})) + \lambda_{MOTION} \cdot R(\mathbf{m}-\mathbf{p}) \quad (1)$$

with $\mathbf{m}=(m_x,m_y)^T$ being the motion vector (MV), and $\mathbf{p}=(p_x,p_y)^T$ being the prediction for the motion vector. The term $R(\mathbf{m}-\mathbf{p})$ represents the motion vector information and is computed by a table-lookup. λ_{MOTION} is given by $\lambda_{MOTION} = \sqrt{0.85 \cdot 2^{QP/3}}$ [1]. In the JM reference code, the ME and the mode decision are executed together. For each mode, ME is done first and the resulted motion cost is used for the mode decision. Although the low complexity mode speeds up the computation significantly, the complexity is still quite high.

To reduce the complexity of the block-size mode-decision, [7] proposes a fast inter-mode block-size mode-decision algorithm based on homogeneous region detections. In [6], fast merging and splitting algorithms are proposed to compose motion vector candidates for different block-size modes to cut down the computation. In this paper, we propose a fast inter-mode decision and ME algorithm by excluding the low-possibility modes in the mode-decision process. Experiments show that the proposed algorithm could achieve similar Rate-Distortion (R-D) performance with about 50% computation saving in block-size mode decision and ME, compared to the JM low-complexity mode with a Fast Full-Search Motion Estimation algorithm.

The organization of this paper is as follows. In section 2, the fast mode-decision and ME algorithm based on the mode exclusion is presented. The calculation analysis and experimental results are shown in section 3. Finally, we conclude this paper in section 4.

2. PROPOSED FAST MODE DECISION AND ME ALGORITHM BY MODE EXCLUSION

Usually in a picture, large areas of background may be either still or under global motion, which can be predicted well by the motion vectors of the neighboring blocks. In these areas, 16*16 usually is the best block-size since it has less MV overheads.

Another observation is that, intuitively, if the cost of a larger block-size mode is higher than the cost of the current block-size mode, then the even larger block-size modes can be excluded. Similarly, if the cost of a smaller block-size mode is higher than that of the current block-size mode, then the even smaller block-size modes can be excluded. For example, if the cost of both 16*8 and 8*16 modes are larger than the 8*8 mode, then it can be inferred that the 16*16 block-size most likely will also not be as good as the 8*8 mode, and so we don't need to check the 16*16 mode. By this way, unlikely modes can be excluded and computations can be saved. The proposed mode exclusion and ME algorithm is illustrated in Fig. 1 (for MB level) and Fig. 2 (for sub-MB level), and is described in details as follows.

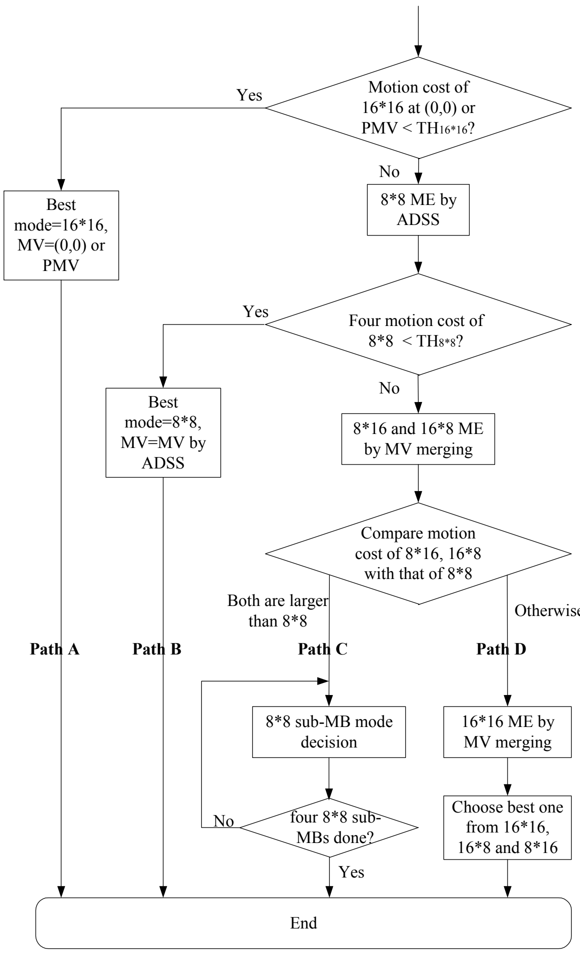

Fig.1 Fast mode-exclusion algorithm: MB level

In Figure 1, TH_{16*16} and TH_{8*8} are thresholds for 16*16 and 8*8 early-stop. PMV is the prediction motion vector, which is the median of the MVs of the neighboring macroblocks, except those by Directional Segmentation Prediction for some 16*8 and 8*16 modes as defined by JVT [1]. ADSS stands for "Adaptive Diversity Search Strategy", which is a diversity-based fast block motion estimation scheme proposed in [4].

As mentioned, often, a large portion of the frame is the background that is still (MV=(0,0)) or undergoes a translational global motion that can be well predicted by the PMV. The optimal mode for these background areas is often the 16*16 block-size mode. So, in the beginning, we check the 16*16 block-size with MV=(0,0) or MV=PMV. If the cost is smaller than the threshold, the mode decision is stopped, and the best mode is 16*16. Similarly, we can also set a threshold for the 8*8 mode. For performance consideration, the thresholds are conservatively set by the minimum of the costs of the 20 previous blocks with the same mode (16*16 or 8*8), plus a fixed value.

$$TH_{16*16} = \min_{i\in\{i|\ 1\le i\le 20,\ MB\ i\ with\ mode\ 16*16\}} (cost_i) + \Delta TH_{16*16}$$

$$TH_{8*8} = \min_{i\in\{i|\ 1\le i\le 20, Sub\text{-}MB\ i\ with\ mode\ 8*8\}} (cost_i) + \Delta TH_{8*8} \quad (1)$$

In our simulation, ΔTH_{16*16}=600 and ΔTH_{8*8} =150. In addition, these two parameters can also be used to control the tradeoff between the complexity and the quality. The larger they are set, the more modes can be excluded, and the faster speed the encoder can achieve.

If either the cost for 16*8 or 8*16 is smaller than that of 8*8, that means larger block-sizes may produce smaller costs. So, the cost for 16*16 is checked and the best one is chosen from 16*16, 16*8, and 8*16. Smaller block-size modes (all sub-MB modes of 8*4, 4*8, and 4*4) are excluded.

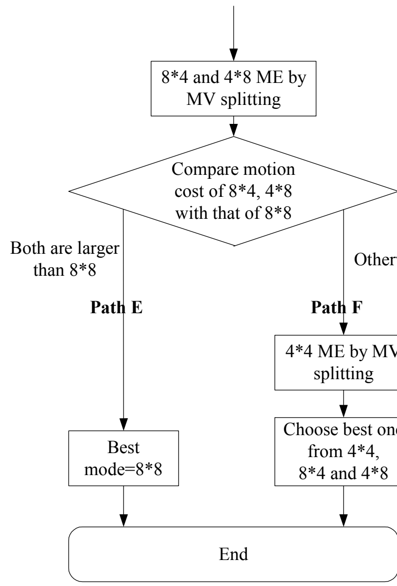

Fig.2 Fast mode-exclusion algorithm: sub-MB level

If the costs for both 16*8 and 8*16 are higher than that of 8*8 (path C), it means larger modes do not achieve lower costs and the sub-MB modes of 8*4 and 4*8 should be checked.

If the costs for both 8*4 and 4*8 are higher than that of 8*8 (path E), that means further separation is not necessary. Therefore, the 4*4 mode is excluded.

Otherwise, one or both of the costs for 8*4 and 4*8 are lower than that of 8*8 (path F). In this case, we then check the cost for 4*4 and choose the best one from 4*8, 8*4, and 4*4 block-size modes.

The algorithm is summarized in the following 7 steps:

1. Estimate the motion for the 16*16 block only at (0,0) or PMV. If either of the motion costs is smaller than

TH_{16*16}, then the mode decision process stops and returns the 16*16 mode.

2. Estimate the motion for the four 8*8 blocks, by ADSS [4].

3. If all of the four 8*8 motion costs are lower than TH_{8*8}, then the best mode is 8*8 for four sub-MBs, and the process stops.

4. For the two 16*8 and 8*16 blocks, estimate the motion by MV merging, following the fast motion estimation algorithm by merge and split proposed in [6].

5. Compare the motion cost for the four 8*8 blocks $MB\_Cost_{8*8} = \sum_{i=1}^{4} Cost_{8*8}(i)$, the two 16*8 blocks $MB\_Cost_{16*8} = \sum_{i=1}^{2} Cost_{16*8}(i)$, and the two 8*16 blocks $MB\_Cost_{8*16} = \sum_{i=1}^{2} Cost_{8*16}(i)$.

- If $MB\_Cost_{16*8} > MB\_Cost_{8*8}$ and $MB\_Cost_{8*16} > MB\_Cost_{8*8}$: the best mode is among the smaller ones. Go to step 6.
- Else (one or both costs are lower than that of 8*8): perform ME for the 16*16 block-size by MV merging, and choose the mode with the minimum motion cost from 16*16, 16*8, and 8*16.

6. For each of the four 8*8 sub-macroblocks: Estimate the MVs for the two 8*4 and 4*8 blocks by MV splitting, following the fast motion estimation algorithm by merge and split proposed in [6].

7. Compare the motion costs for the 8*8 blocks $Sub\_MB\_Cost_{8*8} = Cost_{8*8}(i)$, where i is the serial number of *ith* sub-MB.

The motion cost for the two 8*4 blocks is: $Sub\_MB\_Cost_{8*4} = \sum_{i=1}^{2} Cost_{8*4}(i)$, and for the two 4*8 blocks: $Sub\_MB\_Cost_{4*8} = \sum_{i=1}^{2} Cost_{4*8}(i)$.

- If $Sub\_MB\_Cost_{4*8} > Sub\_MB\_Cost_{8*8}$ and $Sub\_MB\_Cost_{8*4} > Sub\_MB\_Cost_{8*8}$: the best mode is 8*8, and the process stops.
- Else: Perform ME for the 4*4 block-size by MV merging, and choose the mode with the minimum motion cost from 4*4, 4*8, and 8*4.

Repeat step 6 until all four sub-MBs are completed.

3. COMPLEXITY ANALYSIS AND EXPERIMENTAL RESULTS

3.1 Calculation Complexity Analysis

For different modes, the number of excluded modes varies from 1 to 7, as shown in Table-1.

Table-1 Mode decision path and number of modes excluded

Mode Decision Path	Best Mode	Number of Modes Excluded
A	16*16	7 (All)
B	8*8	6 (All except 8*8)
D	16*16, 16*8, 8*16	3 (4*8, 8*4, 4*4)
C->E	8*8	2 (16*16 and 4*4)
C->F	4*8, 8*4, 4*4	1 (16*16)

3.2 R-D Performance and Calculation Reduction

The R-D performance and the calculation comparisons by the proposed algorithm and the JM Low-Complexity Mode (LCM) with Fast Full-Search (FFS) and fast Merge-Split Search (MSS) algorithm for ME [6] are shown in Table-2. In JM, the FFS avoids calculating SAD for each mode repeatedly by SAD-reuse.

Table-2 R-D performance and calculation comparison*

Foreman QCIF

QP		Bit-rate (kb/s)	PSNR (dB)	Run Time**
15	LCM-FFS	1171.03	45.90	1
	LCM-MSS	1193.45	45.87	0.69 (-31%)
	Proposed	1209.89	45.82	0.52 (-48%)
25	LCM-FFS	328.49	38.25	1
	LCM-MSS	348.82	38.25	0.71 (-29%)
	Proposed	351.96	38.19	0.52 (-48%)
35	LCM-FFS	95.33	31.66	1
	LCM-MSS	100.54	31.56	0.66 (-34%)
	Proposed	105.70	31.55	0.48 (-52%)

Akiyo CIF

QP		Bit-rate (kb/s)	PSNR (dB)	Run Time**
15	LCM-FFS	1312.46	47.58	1
	LCM-MSS	1329.50	47.60	0.65 (-35%)
	Proposed	1341.00	47.60	0.48 (-52%)
25	LCM-FFS	390.62	41.82	1
	LCM-MSS	394.01	41.85	0.62 (-38%)
	Proposed	398.69	41.85	0.45 (-55%)
35	LCM-FFS	145.87	35.78	1
	LCM-MSS	147.46	35.82	0.64 (-36%)
	Proposed	149.21	35.82	0.43 (-57%)

Mobile & Calendar CIF

QP		Bit-rate (kb/s)	PSNR (dB)	Run Time**
15	LCM-FFS	9195.10	45.56	1
	LCM-MSS	9167.35	45.55	0.72 (-28%)
	Proposed	9217.85	45.53	0.53 (-47%)
25	LCM-FFS	3812.88	36.98	1
	LCM-MSS	3854.23	36.98	0.70 (-30%)
	Proposed	3883.99	36.94	0.52 (-48%)
35	LCM-FFS	1017.22	28.60	1
	LCM-MSS	1055.57	28.61	0.70 (-30%)
	Proposed	1084.99	28.61	0.51 (-49%)

* GOP structure: IPPPPIPPPP, search range = [-15 15].

** The time is the total encoding run time. The percentage in parentheses is the time saving compared with LCM-FFS.

The tables show that, the proposed algorithm can reduce the calculation by about half, with a slight bit-rate increase compared to the JM low-complexity mode. The bit-rate increase arises from the mode-misjudge. There exists some probability that even both costs of 16*8 and 8*16 are higher than that of 8*8, 16*16, it can still produces lower cost than 8*8. However, under this situation, the cost difference is usually rather small and the distortion is usually negligible. Similarly, for some sub-MBs, although both costs of 8*4 and 4*8 are higher than that of 8*8, there still exists some probability that 4*4 could be the best mode. Similarly, under this situation, the cost difference is usually negligible.

It should be noted that when comparing the proposed algorithm and LCM-MSS, the calculation reduction is (0.52-0.69)/0.69 = 25% (for Foreman QCIF, QP=15).

3.3 Mode-Exclusion Rate

The number of modes excluded can also be taken to analyze the calculation reduction, which is shown in Table-3.

4. CONCLUSIONS AND DISCUSSION

A fast block-size mode-decision and ME algorithm is proposed by exploiting the correlation of the mode costs. Experimental results show that the computation can be reduced by about 1/4 compared to the JM LCM by the MSS ME algorithm, with negligible quality degradation. When compared to JM LCM by FFS, the saving is about 1/2. In addition, the proposed method can be applied to the high complexity mode directly.

Some adaptive algorithms could be further exploited. It is known that smaller block-sizes are preferred at high bit-rates, while larger block-sizes are preferred at low bit-rates, since the motion vector overheads become more significant at low bit-rates. Therefore, for high bit-rate applications, sub-MB modes could be checked first to obtain the best mode faster, which can save calculations for larger blocks (16*8, 8*16, and 16*16). For low bit-rates, larger modes could be checked first. By comparing the best modes weighted by the calculation saved (modes excluded), a judgment to check larger modes first or smaller modes first can be drawn and more calculation reductions can be expected.

5. REFERENCES

[1] "Working draft number 2, revision 8 (WD-2 rev 8)," in *Joint Video Team of ISO/IEC MPEG and ITU-T VCEG,* JVT-B118r8, T. Wieg, Ed., Apr. 2002.

[2] Yu-Kuang Tu, Jar-Ferr Yang, Yi-Nung Shen, Ming-Ting Sun, "Fast Variable-Size Block Motion Estimation Using Merging Procedure With an Adaptive Threshold", *Proc. IEEE Conference on Multimedia & Expo (ICME) 2003,* Baltimore, MD, USA, July 2003.

[3] Mehmet Kucukgoz and Ming-Ting Sun, "Early-stop and Motion Vector Re-using for MPEG-2 to H.264 Transcoding", *SPIE Proc. Visual Comm. Image Proc. (VCIP),* San Jose, CA, USA, Jan. 2004 (to appear).

[4] Jun Xin, Ming-Ting Sun, and Vincent Hsu, "Diversity-based Fast Block Motion Estimation", *Proc. IEEE Conference on Multimedia & Expo (ICME) 2003,* Baltimore, MD, USA, July 2003.

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

[6] Zhi Zhou, Ming-Ting Sun and Yuh-Feng Hsu, "Fast Variable Block-Size Motion Estimation Based on Merging and Splitting Procedures for H.264/MPEG-4 AVC", *IEEE International Symposium on Circuits and Systems (ISCAS) 2004,* Vancouver, BC, Canada, May 2004.

[7] K. P. Lim, S. Wu, D. J. Wu, S. Rahardja, X. Lin, F. Pan, Z. G. Li, "Fast INTER Mode Selection", *Document JVT-I020,* San Diego, USA, 2-5 Sep. 2003.

[8] X. Zhou, E. Q. Li and Y. K. Chen, "Implementing H.26L Decoder on General-Purpose Processors with Media Instructions", *SPIE Conf. on Image and Video Communications and Processing,* San Diego, USA, Jan. 2003.

Table-3 Mode-exclusion rate and calculation reduction analysis
Foreman QCIF,

Mode Decision Path	Best Mode	# of Modes excluded	Percentage		
			QP=15	QP=25	QP=35
A	16*16	7	17%	25.3%	27.7%
B	8*8	6	34%	20%	4.3%
D	16*16, 16*8, 8*16	3	7.1%	31.7%	48.6%
C->E	8*8	2	16.7%	7.5%	17.1%
C->F	4*8, 8*4, 4*4	1	24%	15.3%	0.5%
Average # of modes excluded per MB			4.02	4.23	4
Calculation reduction			57%	60%	57%