FAST PREDICTIVE VARIABLE-BLOCK-SIZE MOTION ESTIMATION FOR H.264/AVC

Zhi Yang, Jiajun Bu, Chun Chen, and Xu Li

College of Computer Science, Zhejiang University, Hang Zhou, P. R. China

ABSTRACT

The emerging H.264 advanced video coding (H.264/AVC) standard achieves significant improvement in coding efficiency compared with previous standards such as MPEG-2 and H.263. One major contribution of its gain profits from variable-block-size motion estimation, which also leads to higher computational complexity. In this paper, we propose a fast predictive variable-block-size motion estimation algorithm for H.264/AVC. The algorithm takes advantage of three effective predictive schemes - stationary block prediction, predictive search for non-stationary blocks and predictive multi-pattern refinement search in merging process. Experimental results and comparative analysis have shown that our proposed algorithm can reduce the computational complexity up to about 2% of fast full search motion estimation algorithm, with negligible average PSNR loss of 0.065 dB and bit rate increase of 2.43%.

1. INTRODUCTION

H.264 is by far the latest video coding standard[1]. Due to some new techniques, such as intra block prediction, variable-block-size motion estimation, multiple reference frames and in-loop deblocking filter, etc., have been introduced into H.264, H.264 significantly outperforms the previous existing video coding standards in terms of coding efficiency and visual quality[2]. At the same time, the complexity of H.264 encoder has been greatly increased. How to reduce the high encoding complexity while preserving good coding performance is an essential research topic, especially for real-time video coding applications.

H.264 applies block-matching motion estimation (BMME) to eliminate temporal redundancy between adjacent frames. Motion estimation (ME) module in H.264 still consumes most of the computing time. However, different from previous standards, variable-block-size ME with a tree-structured hierarchical macroblock partition (*Fig.1*) is introduced into H.264. The traditional fast BMME algorithms are based on several following schemes: fast search pattern, such as newly well-known diamond search (DS) algorithm[3] and hexagon-based search (HEXBS) algorithm[4]; motion vector (MV) prediction according to spatio-temporal correlation[5]; early search termination based on statistical thresholds[6]; accelerating the matching-error computation, such as partial distance based search algorithm[7]. Although the aforementioned schemes can more or less contribute to alleviate the computational load of ME, they do not make use of the variable-block-size feature. Recently some fast variable-block-size ME algorithms have been proposed. In JVT reference software[8], a fast full search (FFS) algorithm is used for variable-block-size ME. In[9], merging procedure with an adaptive threshold is proposed to accelerate the variable-block-size ME process.

Different from previous work, in this paper, we propose a novel fast variable-block-size ME algorithm based on predictive schemes. The proposed algorithm includes three stages. First, top-down predict the stationary blocks. Next, predictive MV search with early-termination scheme is employed for each non-stationary 4×4 blocks. The early-termination threshold is obtained according to all-zero coefficients prediction. Finally, bottom-up merging with predictive multi-pattern refinement search is introduced. Experimental results show that our proposed algorithm can significantly reduce the complexity with negligible coding efficiency degradation, compared with FFSME algorithm.

The rest of this paper is organized as follows. In *Section 2*, we analyze the predictive schemes. Based on the schemes, a fast predictive variable-block-size ME algorithm is presented. Experimental results and comparative analysis are shown in *Section 3*. In *Section 4*, we draw the conclusions and present the future works.

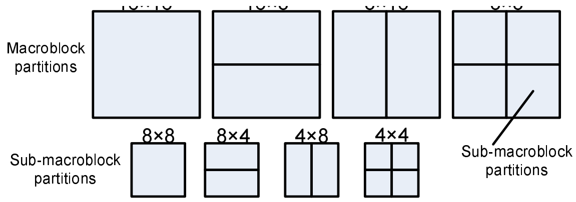


Fig. 1. Variable-block-size partitions in H.264

2. FAST PREDICTIVE VARIABLE-BLOCK-SIZE ME

2.1. Stationary Block Prediction

In many video communication applications, such as video conference and video telephone, there is little motion between adjacent frames and most blocks can be regarded as stationary. Moreover, through full search, we also observed that mode 16×16 has large proportion in macroblock partitions and mode 8×8 has large proportion in sub-macroblock partitions. *Table 1* shows the experimental results. It can be seen that if we can predict the stationary block at the beginning of ME, significant reduction of the computational cost is possible. Further investigations indicate that the average SAD (sum of absolute difference) of these stationary blocks is much smaller than that of non-stationary blocks at zero-motion position. Therefore, the prediction of stationary block can be made by first computing the SAD of zero-motion and comparing it with a predetermined threshold T. If SAD is less than T, the current block can be decided to be stationary without performing the remaining MV search.

---

[*] The work was supported by National Natural Science Foundation of China (60203013), 863 Program (2004AA1Z2390) and Key Technologies R&D Program of Zhejiang Province (2005C23047 & 2004C11052).

Table 1. Proportion of 16×16 and 8×8 modes through full search algorithm. (MB: macroblock; M&D: mother and daughter; M&C: mobile and callendar)

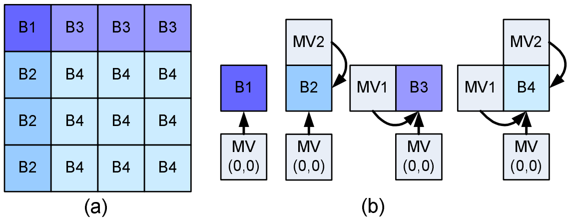
sequence	16×16 **in MB**	**Stationary** 16×16	8×8 **in subMB**	**Stationary** 8×8
news	91.84%	87.34%	58.24%	25.76%
M&D	90.33%	70.81%	67.94%	20.43%
foreman	81.03%	20.81%	60.16%	12.11%
M&C	69.60%	8.91%	53.11%	6.43%
coastguard	68.33%	3.81%	55.76%	6.75%

Since most of block modes are 16×16 and 8×8, we consider two thresholds for predict the 16×16 and 8×8 stationary blocks. After experimental analysis with many test sequences, the average SAD for stationary 16×16 block is $450 - 1000$ and the average SAD for stationary 8×8 block is $150 - 350$. Large thresholds leads to large ME error. Thus, we choose $T1 = 500$ and $T2 = 160$ to predetermine 16×16 and 8×8 stationary blocks respectively. These two thresholds can achieve good speedup gain with negligible ME accuracy degradation.

It need to be mentioned that: for those video sequence (*Table 1*), such as " foreman", "mobile and calendar (M&C)" and "coastguard", which contain large disordered or global motion contents, they cannot benefit much from the stationary block prediction.

2.2. Predictive MV search with early-termination scheme

After stationary block prediction, the MV search will be employed to 4×4 blocks in non-stationary 8×8 blocks. According to[5], spatio-temporal correlation can be used to predict the MV of current block. However, utilizing temporal correlation needs recording the entire previous MV fields and utilizing traditional spatial correlation needs referring to MV fields of neighboring macroblocks. In order to avoid additional storage usage and avoid much if-clause operation, we only exploit the spatial correlation within one macroblock itself (*Fig.2 (a)*). We choose $MV(0,0)$, left MV ($MV1$) and top MV ($MV2$) as candidate predictors. It should be mentioned that the top-left block ($B1$) only can use one predictor, the left blocks ($B2$) or the top blocks ($B3$) only can use two predictors and other blocks ($B4$) use three predictors (*Fig.2 (b)*). We calculate the SAD value of these predictors and choose the one which obtain the minimal SAD as search center.



(a) (b)

Fig. 2. MV prediction (a) block classification (b) predictors selection

After MV prediction, perform diamond search (DS)[3] (*Fig.3*) ME for these 4×4 blocks. During DS process, if we can predetermine the quantized coefficients of 4×4 block are all-zero, the search can be early-terminated. According to[10], the condition of all-zero quantized coefficients is given by *equation 1*. Therefore, we can set a threshold $T3$, which is a function of QP. If the current SAD is smaller than $T3$, the search can be terminated. $T3$ also can be used as a threshold to determine stationary 4×4 blocks before the MV search.

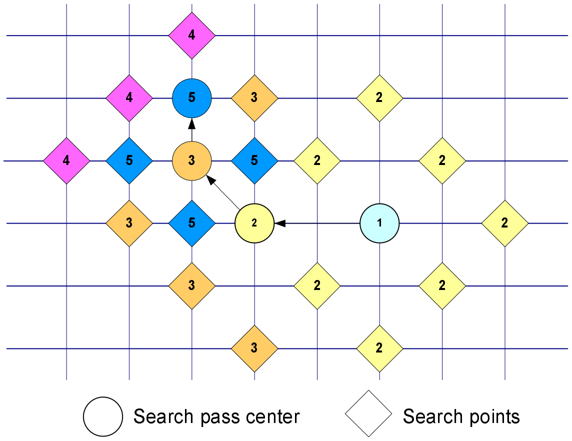


Fig. 3. Diamond search (DS) process

$$SAD < (2^{qbits} - f)/QE[q_{rem}][0][0] = T3 \qquad (1)$$

Where $qbits = 15 + QP/16, q_{rem} = QP\%6, f = (1 << qbits)/6$, QE is the predefined quantization coefficient table and QP is the input quantization parameter.

2.3. Merging strategy with predictive multi-pattern refinement search

After MV search of all non-stationary 4×4 blocks, a bottom-up merging procedure will be taken. In order to reduce the complexity, we propose MV reusing technique which makes use of the MVs of small blocks to predict the MV of large blocks (*Fig.4*). The MV prediction of merged blocks is performed by *equation 2*. The predictive MV will act as the search center for up-layer refinement search.

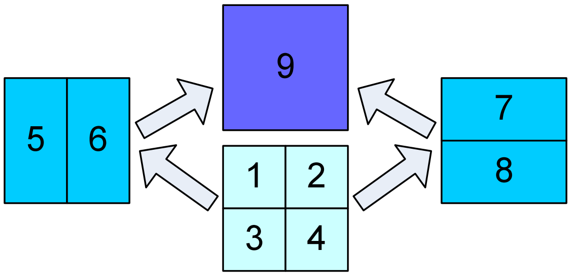


Fig. 4. Bottom-up merging prediction

$$PMV_x = \begin{cases} (MV_1 + MV_3)/2 & \text{if } x = 5 \\ (MV_2 + MV_4)/2 & \text{if } x = 6 \\ (MV_1 + MV_2)/2 & \text{if } x = 7 \\ (MV_3 + MV_4)/2 & \text{if } x = 8 \\ (MV_{5-6} + MV_{7-8})/2 & \text{if } x = 9 \end{cases} \qquad (2)$$

Where $(MV_{5-6} = (MV_5 + MV_6)/2)$, $(MV_{7-8} = (MV_7 + MV_8)/2)$ and x denotes the block according to *Fig. 4*

After MV prediction, we adaptively apply DS (*Fig.3*) and small diamond search (SDS) (*Fig.5*) to achieve the refined MV field.

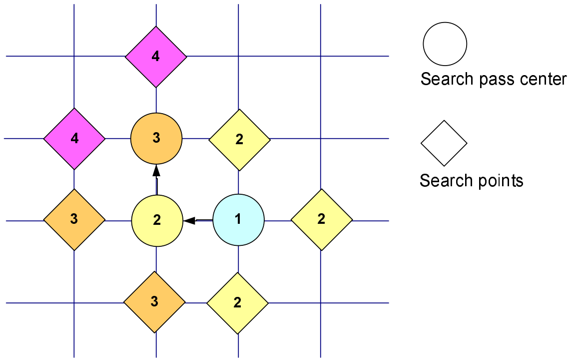


Fig. 5. Small diamond search (SDS) process

From *equations 2*, we can find that each prediction process use two predictors. Then we define the predictors' difference (PD) as *equation 3*.

$$PD = max\{|MV_{p1-x} - MV_{p2-x}|, |MV_{p1-y} - MV_{p2-y}|\} \quad (3)$$

Where MV_{p1-x} and MV_{p1-y} denote the x and y components of predictor 1; MV_{p2-x} and MV_{p2-y} denote the x and y components of predictor 2.

We also define the MV distance (MD) between predictive MV and true MV (obtained by full search) as *equations 4*

$$MD = max\{|PMV_x - TMV_x|, |PMV_y - TMV_y|\} \quad (4)$$

Where PMV_x and PMV_y denote the x and y components of predictive MV; TMV_x and TMV_y denote the x and y components of true MV.

Statistical analysis (*Table 2*) have shown that if $PD \leq 3$, MD is within 2 pixel with high probability. If $PD > 3$, MD is within 6 pixel reaches the high probability. Therefore, the predictive multi-pattern refinement search can be described as follows. If $PD = 0$, get $TMV = PMV$; else if $PD \leq 3$, do SDS ME and the search range is from -2 to +2; else if $PD > 3$, do DS ME and the search range is from -6 to +6.

2.4. Algorithm Summary

The fast predictive variable-block-size ME can be summarized as follows.

- **Step 1:** Calculate SAD for 16×16, 8×8 and 4×4 blocks in zero-motion position.
- **Step 2:** If $SAD_{16\times16} < T1$, then the 16×16 block is determined stationary and the search is done. Else go to step 3.
- **Step 3:** For four 8×8 blocks in non-stationary 16×16 block
  - If $SAD_{8\times8} < T2$, the 8×8 block is marked stationary, else marked non-stationary.
- **Step 4:** If four 8×8 blocks are all marked stationary, then the 16×16 block is determined stationary and the search is done. Else go to step 5.

Table 2. Cumulative probability distribution of MD for merged blocks by full search

sequence	$PD \leq 3$			
	$MD = 0$	$MD \leq 1$	$MD \leq 2$	$MD \leq 3$
news	89.38%	97.30%	98.58%	99.06%
M&D	69.73%	89.34%	93.21%	95.09%
foreman	63.71%	91.47%	95.78%	97.16%
M&C	77.87%	97.01%	98.57%	99.10%
coastguard	79.18%	97.01%	98.70%	99.14%
	$PD > 3$			
	$MD = 0$	$MD \leq 2$	$MD \leq 4$	$MD \leq 6$
news	1.46%	36.20%	79.73%	95.02%
M&D	1.24%	29.53%	68.84%	94.16%
foreman	1.40%	32.21%	73.73%	95.20%
M&C	1.34%	38.76%	87.76%	97.59%
coastguard	1.01%	34.90%	83.90%	97.11%

- **Step 5:** For each 4×4 blocks in non-stationary 8×8 blocks
  - If $SAD_{4\times4} < T3$, the search for the 4×4 block is bypassed. Else employ predictive DS search with early termination scheme (section 3.2) to find the best MV for the 4×4 block.
- **Step 6:** Bottom-up merge and employ predictive multi-pattern refinement search (PMRS) scheme (section 3.3) to get the refined MVs for merged blocks.
  - From 4×4 blocks, merge to 4×8 and 8×4 blocks and employ PMRS.
  - From 4×8 and 8×4 blocks, merge to 8×8 blocks and employ PMRS.
  - From 8×8 blocks, merge to 8×16 and 16×8 blocks and employ PMRS.
  - From 8×16 and 16×8 blocks, merge to 16×16 block and employ PMRS.

3. EXPERIMENTAL RESULTS

To evaluate the performance of our proposed fast predictive variable-block-size ME, we implemented the algorithm into the JVT reference software version 8.5 (JM8.5)[8]. We have tested our proposed algorithm with a set of CIF (352 × 288) video sequences which present different kinds of motion. In the experiments, we encoded the sequences with 30fps. The baseline profile encoder was used through all simulations. The quantization parameter (QP) has a range of 20, 24, 28, 32, 36 and 40. Search range is set from -16 to 15 and only 1 reference frame is under consideration. We have used PSNR gain and bit rate gain to compare ME efficiency of our proposed algorithm with that of FFSME algorithm. The speedup is also based on FFSME. *Table 3* shows the experimental results.

It can be seen that: for "news" and "M&D" sequences with motion limited in the center region, our proposed algorithm achieves averagely 45 speedup of FFSME algorithm, with 0.076 PSNR loss and 2.35% bit rate increase; for "foreman" sequence with disordered large motion, our proposed algorithm achieves about 25

Table 3. Performance of the proposed algorithm compared with FFSME

sequence	QP	PSNR gain (dB)	bit rate gain (%)	speedup
news	20	-0.18	3.70%	40.5
	24	-0.11	3.65%	44.0
	28	-0.08	1.88%	46.7
	32	-0.03	2.34%	47.3
	36	-0.06	5.19%	48.9
	40	-0.03	4.25%	49.8
M&D	20	-0.17	0.61%	38.4
	24	-0.10	1.14%	41.3
	28	-0.06	2.11%	44.1
	32	-0.04	1.23%	46.9
	36	-0.03	2.17%	48.9
	40	-0.02	0.00%	50.1
foreman	20	-0.10	2.62%	25.2
	24	-0.07	3.27%	25.4
	28	-0.06	4.40%	25.7
	32	-0.07	6.04%	26.4
	36	-0.08	6.92%	26.8
	40	-0.07	8.71%	27.1
M&C	20	-0.07	0.18%	29.9
	24	-0.05	0.28%	30.1
	28	-0.05	0.24%	30.2
	32	-0.05	0.30%	30.4
	36	-0.04	0.47%	30.7
	40	-0.05	1.00%	30.8
coastguard	20	-0.04	0.23%	31.1
	24	-0.03	0.30%	31.2
	28	-0.03	0.72%	31.3
	32	-0.05	1.16%	31.6
	36	-0.04	2.30%	31.7
	40	-0.04	3.67%	31.9
average		**-0.065**	**2.43%**	**35.8**

speedup of FFSME algorithm, with 0.075 PSNR loss and 5.32% bit rate increase; and for “M&C”and “coastguard” sequences with global motion, our proposed algorithm also achieve 30 speedup of FFSME algorithm with negligible performance degradation. Therefore, it can be concluded that our proposed algorithm can adapt to all kinds of motion scenarios. Especially, for some specific scenarios, such as video phone and video conference (just like “news” and “M&D”sequences) with low bit rate video coding, the stationary block prediction scheme can take more effect. If there is strong spatial correlation in video frame, such as capture moving scenarios (just like ”coastguard” and ”M&C” sequences), the MV predictive schemes can take more effect.

4. CONCLUSIONS AND FUTURE WORKS

In this paper, we proposed a novel fast predictive variable-block-size ME algorithm for H.264 advanced video coding. Through statistical analysis, three predictive schemes are introduced into the ME process. Experimental results have shown that the proposed algorithm can achieve a good tradeoff between ME complexity and coding efficiency. The ME complexity can be reduced up to about 2% of FFSME algorithm. 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 with adaptive thresholds. In our future work, we will consider how to extend our algorithm to support multiple references search without much complexity increasing.

5. ACKNOWLEDGEMENT

The authors thank Linjian Mo and Shuiyong Lou for giving some suggestions about the paper.

6. REFERENCES

[1] T. Wiegand; G.J. Sullivan; G. Bjntegaard; A. Luthra, “Overview of the h.264/avc video coding standard,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 13, pp. 560–576, July 2003.

[2] 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, pp. 7–28, 2004.

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

[4] C. Zhu; X. Lin; L.P. Chau, “Hexagon-based search pattern for fast block motion estimation,” *IEEE Transaction on Circuits and Systems for Video Technology*, vol. 12, pp. 349–355, May 2002.

[5] H. S. Yoon; G. S. Lee, “Motion estimation based on spatio-temporal correlations,” *Proceedings of the International Conference on ICIP ’03*, vol. 2, pp. 359–362, Sept. 2003.

[6] J.F. Yang; S.C. Chang; C.Y. Chen, “Computation reduction for motion search in low rate video coders,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, pp. 948–951, Oct. 2002.

[7] K. Lengwehasarit; A. Ortega, “Probabilistic partial-distance fast matching algorithms for motion estimation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 11, pp. 139–152, Feb. 2001.

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

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

[10] Z.B. Chen; P. Zhou; Y. He, “Fast integer pel and fractional pel motion estimation for jvt,” *JVT-F017, 6th JVT Meeting, Awaji Island, Japan*, Dec. 2002.