Fast Intermode Decision via Statistical Learning for H.264 Video Coding

Wei-Hau Pan, Chen-Kuo Chiang, Shang-Hong Lai

Department of Computer Science, National Tsing Hua University, Hsinchu, Taiwan

{whpan, ckchiang, lai}@cs.nthu.edu.tw

Abstract. Although the variable-block-size motion compensation scheme significantly reduces the compensation error, the computational complexity of motion estimation (ME) is tremendously increased at the same time. To reduce the complexity of the variable-block-size ME algorithm, we propose a statistical learning approach to simplify the computation involved in the sub-MB mode selection. Some representative features are extracted during ME with fixed sizes. Then, an off-line pre-classification approach is used to predict the most probable sub-MB modes according to the run-time features. It turns out that only possible sub-MB modes need to perform ME. Experimental results show that the computation complexity is significantly reduced while the video quality degradation and bitrate increment is negligible.

Keywords: Variable block-size, H.264, motion estimation, statistical learning

1 Introduction

H.264/AVC, the latest video coding standard of Joint Video Team (JVT), outperforms previous standards such as MPEG-4 and H.263 in terms of coding efficiency and video quality. This is due to the fact that many new techniques are adopted in this standard. Variable-block-size ME is one of the most important features in H.264. There are seven kinds of block sizes, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4. An MB (16×16) can be partitioned to 16×8, 8×16 and 8×8. A sub-MB (8×8)

can be further partitioned to 8×4, 4×8 and 4×4. Different types of partitions are shown in Fig. 1.

In [1], the degrees of homogeneous and stationary regions are determined as the criteria for block partition. This is based on the assumption that the higher the degree of homogeneous and stationary blocks, the larger the block partition is used. However, the thresholds to determine the degree of homogeneity are empirically selected, and the resulting criteria can not provide very accurate block partitioning. Another approach [2] analyzes the likelihood and the correlation of motion fields for a suitable block mode selection.

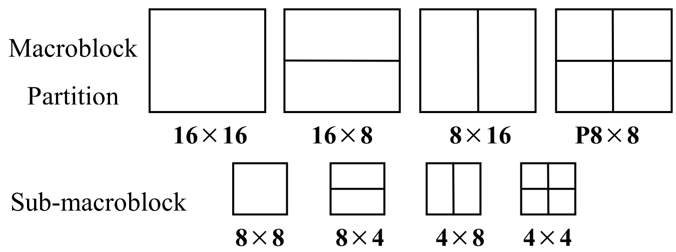

Fig. 1. MB partitions and sub-MB partitions in H.264

In this paper, we propose an algorithm by using statistical learning to analyze representative features for variable-block-size ME and determined the most probable modes. More accurate mode decision can be obtained by statistical learning results rather than heuristic thresholding on simple features. On the other hand, it eliminates all unnecessary computation involved in variable-block-size ME.

2 Feature Selection

In this section, we will introduce some representative features for variable-block-size mode selection. These features can be used to predict the most probable partition type of the current MB. All of them are well chosen and examined carefully in our experiments. The results show that they are helpful to the variable-size mode selection.

2.1 Inter Best SAD

For an Inter prediction MB, the ME procedure determines the best matching reference MB. The distortion measure used in this ME procedure in H.264 is the sum of absolute difference (SAD). The Inter Best SAD is the lowest SAD value of the ME results to the current MB. This value may indicate not only the accuracy of ME procedure but also the possibility of being a background MB. The lower this value is, the higher probability the current MB contains still background. Thus, small Inter Best SAD value means it is very unlikely that this MB will be split up into sub-MBs. This is due to the fact that a background MB is stationary and can be matched well by a large MB. On the other hand, we need to consider the bitrate overhead caused by more motion vectors if we split up a MB into sub-MBs. Fig. 2(A) shows that lower SAD value indicates higher probability of being a 16×16 or an 8×8 partition mode.

2.2 Motion Vector Difference and Motion Vector Magnitude

Motion Vector Difference (MVD) is the sum of absolute value of the difference between the predicted MV and the motion vector after ME in horizontal and vertical directions. In the H.264 standard, the predicted MV is defined as the median MVs of the adjacent blocks in both x and y directions. MVD may represent the motion smoothness between current MB and adjacent MBs. Fig. 2(B) indicates that if MVD is small, the current MB has more chance to be a background block. In this case, it is not necessary to partition this MB into sub-MBs.

Motion Vector Magnitude (MVM) is the sums of absolute value of the motion vector itself regardless of the MV prediction. It indicates that whether this MB is stationary or not. Stationary MBs are still temporally. That is, little change between the current block and the collocated block in the previous frame can be detected. If the MB is stationary, it can be matched well by large MBs. Another reason to consider MVM is that it may show more precise MB activity than MVD when the camera is fixed. Fig. 2(C) shows that MVM is helpful to decide the mode when the MB is stationary.

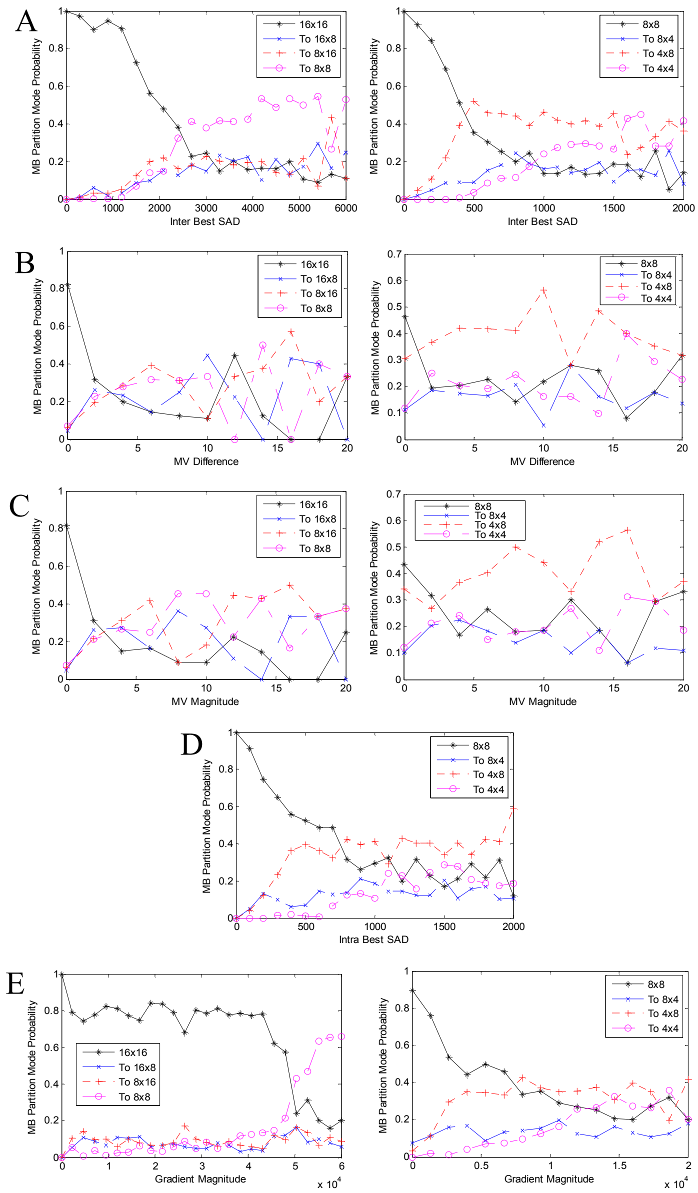

Fig. 2. The probability of partition mode for 16×16 and 8×8 MBs under different (A) Inter Best SAD, (B) MVD, (C) MVM, (D) Intra Best SAD and (E) Gradient Magnitude in News sequence.

2.3 Intra Best SAD and Gradient Magnitude

Intra Best SAD is the minimum SAD value after Intra prediction of the current MB. An MB with a large SAD value after Intra prediction usually contains complicated texture or object boundaries. Therefore, such kinds of MBs tend to be partitioned into smaller sub-MBs. It is obvious especially for 8×8 MBs, so we take this feature for 8×8 MBs. In Fig. 2(D), the 8×8 MBs with higher Intra SAD value have larger probability to be partitioned into smaller sub-MBs.

The gradient magnitude of the current $M\times N$ MB is defined as the summation of the gradient magnitudes of all pixels inside the MB obtained by applying the Sobel operator.

The gradient magnitude is low in homogeneous regions. An MB with low gradient magnitude tends to be a background block, which is unlikely to be partitioned into sub-MBs. It is obvious in Fig. 2(E) that the gradient magnitude provides useful information for our purpose.

3 Problem Formulation and Solution

We will formulate a sub-MB mode decision problem into a classification problem and provide a solution in this section. Firstly, ME is performed on 16×16 MBs. Then, the most possible sub-MB mode is predicted based on the selected features.

3.1 Problem Formulation for a 16×16 MB and an 8×8 MB

There are four possible partition modes for a 16×16 MB, including 16×16, 16×8, 8×16 and P8×8. Since we have performed ME for 16×16 MBs, we need to decide which mode to perform ME in the next step. Our experiments indicate that 16×16, 16×8, and 8×16 modes usually have similar features. Thus, partition mode 16×16, 16×8 and 8×16 are grouped to class one (C_1) and P8×8 is set as class two (C_2). Four features, mentioned in the Section 2 are used for the classification, including Inter

Best SAD (SAD_P), Motion Vector Difference (MV_D), Motion Vector Magnitude (MV_{mag}), and Gradient Magnitude (G). Thus, the binary classification results can be decided from the class conditional probabilities given the features, i.e.

$$\text{Decide } C_1 \text{ if}$$
$$P(C_1 \mid SAD_P,\ MV_D,\ MV_{mag},\ G) > P(C_2 \mid SAD_P,\ MV_D,\ MV_{mag},\ G)\ ; \qquad \textbf{(1)}$$
$$\text{otherwise decide } C_2.$$

For an 8×8 MB, there are also four possible partition modes including 8×8, 8×4, 4×8 and 4×4. Similar to the case for a 16×16 MB, we set partition mode 8×8 as class three (C_3) and the mode 8×4, 4×8 and 4×4 are grouped to class four (C_4). In addition to the four features used for 16×16 MB, the Intra Best SAD (SAD_I) is included in the binary classification for an 8×8 MB. The decision is still based on the class conditional probability given as follows

$$\text{Decide } C_3 \text{ if}$$
$$P(C_3 \mid SAD_P,\ MV_D,\ MV_{mag},\ G,\ SAD_I) > P(C_4 \mid SAD_P,\ MV_D,\ MV_{mag},\ G,\ SAD_I)\ ; \qquad \textbf{(2)}$$
$$\text{otherwise decide } C_4.$$

3.3 Training and Off-Line Pre-Classification

In the above, two decision rules are defined for our classification problem. However, it is difficult to model the joint probability of those features well from limited training samples. The Support Vector Machine (SVM) is used here to solve the classification problem.

The training data is obtained by applying the H.264 reference code JM 11.0 to four video sequences; namely, News, Akiyo, Foreman and Coastguard. For the 16×16 MBs, the required features and the MB partition mode results (C_1 or C_2) are collected, similarly for the case of the 8×8 MBs (C_3 or C_4). The collected data is regarded as the input training samples for SVM. Experimental results show that the accuracy of cross validation is very high by using SVM on these two classification problems.

For the consideration of real-time encoding, it takes too much time for run-time classification for SVM. Thus, an off-line pre-classification approach is proposed to minimize the computation time involved in the classification procedure. The idea is to generate all possible combinations of the feature vectors and pre-classify them with SVM. However, the total number of possible combinations is too large for real applications. A useful method is to quantize each feature based on the feature distribution. Instead of using the uniform quantizer, Llyod-Max quantizer [3] which has adaptive step size is applied on the training samples since it can approximate a distribution much better than the uniform quantizer. Each feature is quantized into 20 bins in our implementation.

From the trained SVM classifiers, we can decide the class for each possible input sample. The classification results for all possible input samples are stored. During the encoding, we only need to collect the necessary features, quantize them and search the look-up table for classification. Hence, the computation time in the classification can be significantly reduced by using this off-line pre-classification approach.

4 Proposed Intermode Decision Algorithm

The procedure of the proposed Intermode decision algorithm is given as follows:.

Step 1) Perform 16×16 ME.

Step 2) Collect four features: SAD_P, MV_D, MV_{mag}, and G_{mag} for 16×16 MB.

Step 3) Quantize these features from the Lloyd-Max quantizer.

Step 4) Obtain the classification result from the table look-up. If the results is *C1*, perform 16×8, 8×16 ME and go to Step 12.

Step 5) Perform 8×8 ME.

Step 6) Collect five features: SAD_P, MV_D, MV_{mag}, G_{mag} and SAD_I for 8×8 MB.

Step 7) Quantize these features from the Lloyd-Max quantizer.

Step 8) Classify via table look-up procedures. If *C3* is preferable, go to Step 11.

Step 9) Perform 8×4, 4×8, and 4×4 ME.

Step 10) Determine the best P8×8 MB partition mode.

Step 11) Repeat from Step 5 to Step 10 until all the 8×8 MBs are performed.

Step 12) Select the best MB partition mode as the Intermode. Go to Step 1 and proceed to the next MB.

Notice that the feature quantization and the classification step are simply via table look-up. Thus, they have insignificant computation overhead for the encoding. The 8×8 Intra prediction is performed before Step 6 since the four features, SAD_P, MV_D, MV_{mag} and SAD_I, are required for encoding. The main overhead of the proposed algorithm is on the calculation of G_{mag} for each MB.

5 Experimental Results

The proposed algorithm and the algorithm by Wu et al. [1] are implemented in JM11.0 using the fast ME, EPZS [4]. The motion search range is set to 32 and the number of reference frame is set to 1. The RD optimization is disabled and the CABAC entropy encoding is enabled in our experiments. All test sequences are in QCIF format and tested on an Intel Pentium M processor of 1.73GHz. All frames except the fist frame are encoded as P-frames. The training data for our algorithm is collected from News, Akiyo, Foreman, Coastguard sequences. The QP is set to 28.

We applied Full Search (FS), EPZS, our proposed algorithm and the method by Wu et al. [1] to six test sequences. The PSNR increase and bitrate increase compared with Full Search are listed in Table 1 when QP is set to 24, 28 and 32, respectively. The speedup ratios are calculated for three methods compared with FS in terms of execution time and number of search points per MB when QP is set to 24, 28 and 32. Note that SP stands for the number of search points per MB, which is the number of search points per mode multiplied by a weight proportional to the size of the partition mode. Fig. 3 shows the average speedup ratios of the results when QP is set to 24, 28 and 32 for different sequences. It indicates that the number of search points in the proposed algorithm is about half of those of the other two methods in JM while the execution time is about three times faster than the other methods.

Table 1. PSNR and Bitrate decrease compared with the Full Search for the three algorithms when QP is set to 24, 28 and 32.

Sequence	PSNR increase (dB)			Bitrate increase (%)		
(QP 24)	EZPS	Proposed	Wu's	EZPS	Proposed	Wu's
Mobile	0.01	0.01	0.04	-0.64	-0.25	-0.65
HallMonitor	0.02	0.01	0.02	0.07	1.35	0.02
Container	0.05	0.02	0.04	-0.55	-0.64	-0.59
Akiyo	0.01	0.00	0.01	-0.62	0.21	-0.62
News	0.02	0.01	0.02	-1.37	0.09	-1.25
Foreman	0.01	-0.00	0.01	-0.12	1.12	-0.05
(QP 28)	EZPS	Proposed	Wu's	EZPS	Proposed	Wu's
Mobile	0.03	0.02	0.03	-0.97	-0.56	-1.01
HallMonitor	0.02	0.04	0.07	0.79	1.41	1.07
Container	0.08	0.08	0.08	-0.75	-0.46	-0.40
Akiyo	0.04	0.05	0.04	-1.23	-0.51	-1.23
News	0.04	0.04	0.03	-0.94	0.30	-0.88
Foreman	0.02	0.01	0.00	-0.96	1.32	-0.83
(QP 32)	EZPS	Proposed	Wu's	EZPS	Proposed	Wu's
Mobile	0.07	0.06	0.07	-2.34	-1.74	-2.32
HallMonitor	0.00	0.02	0.03	-0.17	1.35	-0.55
Container	0.21	0.21	0.21	-0.11	-0.10	-0.11
Akiyo	0.10	0.14	0.10	-3.38	-2.46	-3.38
News	0.22	0.16	0.17	-2.01	-0.53	-1.30
Foreman	0.03	0.04	0.03	-2.39	-1.18	-2.85

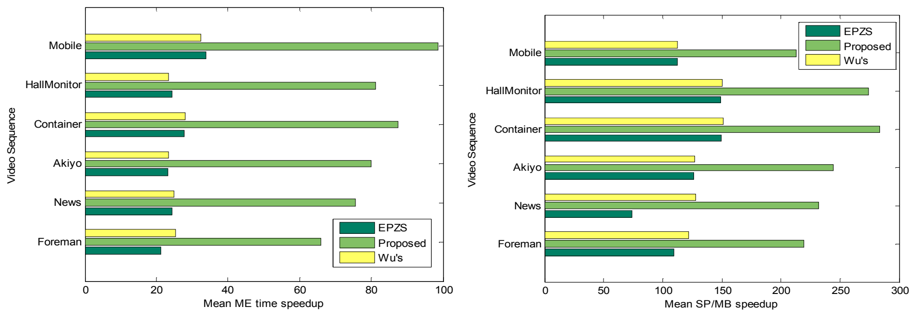

Fig. 3. Average speedup ratios of ME time and SP/MB compared with the Full Search for the three algorithms, including EPZS, Wu's and the proposed method.

From these results, it is obvious that our algorithm has similar performance compared with FS. Both the average PSNR and bitrate variations are negligible, but the average ME time speedup (81.48) and average SP/MB speedup (244.47) factors of our proposed algorithm are significantly greater than those of the other two algorithms.

6 Conclusions

In this paper, we presented a variable-block-size mode decision algorithm based on statistical learning. In this work, several representative features are investigated to help decide the best Intermode from variable block sizes. Experimental results show that they can provide good discriminating features for the Intermode classification problem. To our knowledge, this is the first work that introduces the statistical learning technique into the variable-block-size Intermode decision problem. Experimental results show that the number of search points required in the proposed algorithm is about half of the previous methods, including the EPZS method used in JM. The execution speed of our algorithm is about three times faster than these existing methods while achieving nearly the same compression quality in terms of PSNR and bitrate.

References

1. 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 Trans. Circuits & Systems For Video Technology, Vol. 15, No. 6 (2005) 953-958
2. T.-Y. Kuo and C.-H. Chan, Fast Variable Block Size Motion Estimation for H.264 Using Likelihood and Correlation of Motion Field, IEEE Trans. Circuits & Systems for Video Technology, Vol. 16, No. 10 (2006) 1185-1195
3. A.K. Jain, Fundamentals of digital image processing. Prentice Hall. chapter 4 (1989).
4. A.M. Tourapis, Enhanced Predictive Zonal Search for Single and Multiple Frame Motion Estimation, in: SPIE Proceedings of the Visual Comm. Image Proc., 2002.