Fast Mode Selection Based on Texture Analysis and Local Motion Activity in H.264/JVT

Yanfei shen   Dongming Zhang   Chao Huang   Jintao Li
Institute of Computing Technology of
Chinese Academy of Sciences
Email: {syf,dmzhang,chuang,jtli}@ict.ac.cn

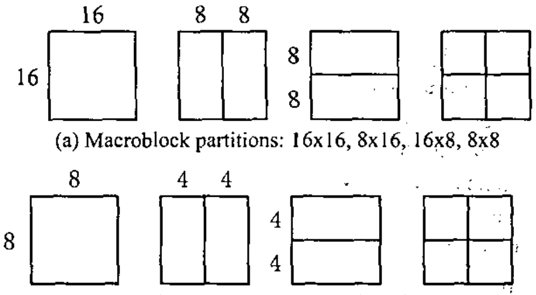
Abstract – ***The H.264/JVT video coding standard can achieve considerably higher coding efficiency than previous standards. Unfortunately this comes at the cost of greatly increased complexity at the encoder mainly due to multi-reference frames and multi-block type motion estimation. In order to speed up the process of multi-block type motion estimation, in this paper a method is proposed to eliminate some redundant coding modes which contribute very little coding gain based on the texture analysis and local motion activity. Simulation results show that the proposed method can eliminate about half of all coding modes while keeping similar coding efficiency, visual quality and bitrates. Furthermore this method can be used with any fast motion estimation algorithm.***

Keywords: mode selection   video coding   H.264

I. INTRODUCTION

H.264[1] is a new international video coding standard with superior objective and subjective video quality. The basic encoding algorithm of H.264 is similar to H.26x[2][3] or MPEG series[4][5] except that integer 4x4 discrete cosine transform (DCT) is used instead of the traditional 8x8 real DCT. Additional features include directional intra prediction for intra coding, multi-block types for motion estimation, multiple references frame selection for higher coding efficiency[6], etc.

In particular, H.264 supports motion estimation and compensation using different block sizes ranging from 16x16 to 4x4 luminance samples with many options between the two. The luminance component of each macroblock can be split by four ways as shown in Fig.1 (a): 16x16, 16x8, 8x16 and 8x8, each of the sub-macroblock partitions is called a macroblock partition. If the 8x8 mode is chosen, each of 8x8 macroblock partitions within the macroblock can be further split by four ways as shown in Fig.1 (b): 8x8, 8x4, 4x8 or 4x4, which are called macroblock sub-partitions. These partitions and sub-partitions give rise to a lager number of possible combinations within each macroblock. This method of splitting macroblocks into partitions or sub-partitions of varying size is known as tree structured motion estimation. Simulation results show that this tree structured motion estimation can save more than 15% of the bit rates when compared with using only 16x16 block type.



(a) Macroblock partitions: 16x16, 8x16, 16x8, 8x8

(b) Macroblock sub-partitions: 8x8, 4x8, 8x4, 4x4

Fig.1. Multiple Block Types in H.264

However, the computational complexity required by motion estimation increases linearly with the number of used block types because block matching needs to be performed for each of them. In JVT reference software JM75C[7], it adopts full search method for each block type and selects the optimal block type as the final coding mode based on the RD cost function. It provides the best coding efficiency, but its computational complexity is very high. In order to reduce the intensive computational requirement, Andy Chang etc. proposed fast multi-block motion estimation[8]. They use early termination to skip the searching for mode 16x8 and mode 8x16 if the performance of mode 16x16 is "good enough", otherwise all coding modes will be performed. They only consider three coding modes which are 16x16, 16x8 and 8x16 inter coding modes; In[9], Peng Yin etc assume that the error surface is monotonic which is built by initial three coding modes: 16x16, 8x8 and 4x4, then select partial coding modes to do motion estimation based error surface tendency, so at least five coding modes need to be tested. In this paper, we propose a method to eliminate some redundant coding modes based on the texture analysis and local motion activity, accordingly computational complexity caused by them can also be eliminated.

The paper will be organized as follows. Some observations in multi-block type motion estimation will be shown in section 2. Section 3 describes the proposed fast mode selection method based on texture analysis and local

0-7803-8647-7/04/$20.00©2004 IEEE.

motion activity. Simulation results will be presented in section 4. A conclusion will be given in section 5.

II. OBSERVATIONS ON MULTI-BLOCK TYPE MOTION ESTIMATION

Accurate prediction can efficiently reduce the degree of error between the original image and the predicted image. For this reason, multi-block type motion estimation is adopted in H.264 for better compression performance. For a macroblock, it is possible to contain more than one object which may move in different directions. Therefore using only one motion vector per macroblock may be not enough to precisely describe the motion of all objects contained in it. In this case, only part of the macroblock can have good motion compensation and the resulting residue energy can be large due to the mismatch in the remaining part of the macroblock.

In JVT reference software JM75C, motion estimation is performed mode by mode with full search scheme, that is, we need to perform motion estimation for each coding mode in every previous reference frame. The allowed modes are inter16x16, inter16x8, inter8x16, inter8x8, intra4x4 and intra16x16. Note that the inter8x8 block can be further partitioned into smaller blocks. The best mode is selected by minimizing the following Lagrange cost function:

$$J(m,\lambda_{motion}) = SAD(s,c(m)) + \lambda_{motion} \cdot R(m-p) \qquad (1)$$

where $m=(m_x,m_y)^T$ is the motion vector, $p=(p_x,p_y)^T$ is the prediction for motion vector and λ_{motion} is the Lagrange multiplier, $R(m-p)$ represents the bits used to encode the motion information, s is the coded video signal, c is the prediction video signal. We can see that the processing time increases linearly with the number of coding modes. If we can eliminate some redundant coding modes which contribute very little coding gain, it will reduce the intensive computational requirement needed by motion estimation for them.

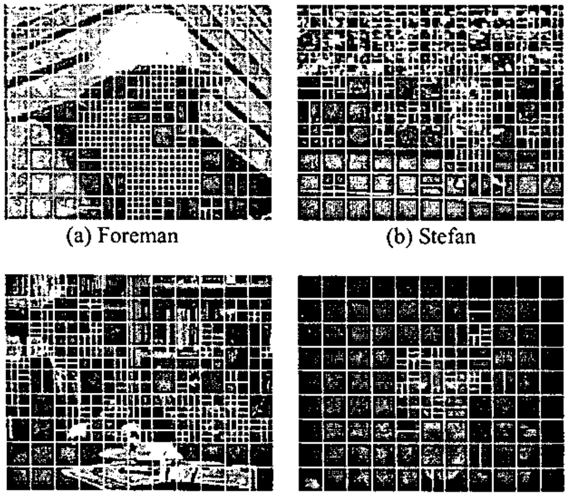
In addition, all kinds of coding modes are not averagely distributed in RD optimization. In literature [10], Yu-Wen Huang etc make statistics for selected modes after motion estimation, intra prediction and RD optimal, results are shown in Table.1. We can see that average 73%, 4%, 4%, 17% and 2% of the MBs respectively select 16x16, 16x8, 8x16, 8x8 and intra coding mode. Therefore we should analyze scenes that are suitable for different coding modes, especially for 16x16 inter coding mode. For an encoding macroblock, if we can accurately select one or several coding modes or eliminate some redundant coding modes which is not suitable for this macroblock based some analyses of this macroblock, a lot of unnecessary computation can be saved. In this paper, we mainly concentrate on the analyses of the relations among coding modes, texture characteristic and local motion activity.

Table.1. statistics of selected modes

Sequences	16x16	16x8	8x16	8x8	Intra
Coastguard	57	06	06	30	01
Container	92	01	01	04	02
Foreman	64	08	08	17	03
Hall Monitor	90	01	01	07	01
Mobile	49	06	07	37	01
Mother	89	03	03	04	01
Silent	83	03	03	10	01
Stenfan	47	06	05	38	04
Table Tennis	76	04	04	13	03
Weather	87	01	04	09	01
Average	73	04	04	17	02

In order to present the relations among coding modes, texture characteristic and local motion activity, four reconstructed frames produced by H.264 reference software JM75c are shown in Figure.2. For test sequences foreman and Stefan, they not only contain global motion caused by camera, but also contain local motion caused by several objects in the frame. We can see that it is more possible to select 16x16 coding mode for "smooth region" macroblock regardless of its motion activity. That is, this macroblock may belong to one object because it does not contain edge information. If more than one objects are contained in a macroblock and moving in different directions, it is suitable that this macroblock should be split into multiple sub-blocks for motion estimation and compensation. Reversely if this macroblock contains edge information, that is, its texture is complex, it is not always split into smaller sub-blocks. For example, in test sequence paris, there are complex textures in its background, but they mostly select larger block size coding mode because they belong to still background region and there is no motion. On the other hand, if the motion vectors of neighbor macroblocks are consistent, the current macroblock may belong to the same object as which contains its neighbor macroblocks, and all sub-blocks of current macroblock may contain similarly movement, so it is not necessary to perform motion estimation and compensation for smaller sub-block types, such as 8x4, 4x8 and 4x4 coding modes.

For 16x8 and 8x16 coding modes, they correlate with the direction of texture, such as, if the texture direction is horizontal, they can contain horizontal edge, so it is possible to select 16x8 coding mode and it is not necessary to perform search for 8x16 coding mode, whereas 8x16 coding mode should be used for macroblock containing vertical edge information.



(a) Foreman (b) Stefan

(c) Paris (d) Miss American

Fig.2. Reconstructed Frames Produced by JM75C

So we can conclude that

(1) If the texture of macroblock is unitary, it trends to select 16x16 inter coding mode and there is no necessary to perform motion estimation for other coding modes;

(2) If the macroblock contain horizontal edge information, we should set 8x16 coding disable, reversely if the macroblock contain vertical edge information, we should set 8x16 coding mode disable.

(3) If the macroblock belongs to static region or the motion vectors of neighbor macroblocks are consistent, it trends to select larger block coding modes, for example 16x16, 16x8, 8x16 or 8x8 coding modes, and there is no necessary to perform motion estimation for smaller sub-block coding modes whose size is less than 8 pixels.

III. Proposed Fast Mode Selection Method

The proposed fast mode selection method is based on above observed results. First we analyze current macroblock texture characteristics using the following equation:

$$Var = \sum_{i=0}^{m}\sum_{j=0}^{n}|p(i,j) - Avg| \qquad (2)$$

Where Avg is the average of current macroblock luminance pixel intensity $p(i,j)$, m and n are the block size used to calculate variance Var, which present texture complexity of current macroblock. For "smooth" macroblock, Var trends to very small. If $Var < TH_{(MB)}$, we can deduce that this macroblock should belong to one object, only 16x16 inter coding mode is selected to perform motion estimation and compensation, all other coding modes will be skipped. Note that $TH_{(MB)}$ is predefined threshold decided experimentally. If $Var > TH_{(MB)}$, we then detect its texture direction based on the following equation:

$$TDH = \sum_{i=0}^{m}\sum_{j=1}^{n}|p(i,j) - p(i,j-1)| \qquad (3)$$

$$TDV = \sum_{i=1}^{m}\sum_{j=0}^{n}|p(i,j) - p(i-1,j)| \qquad (4)$$

Where $p(i,j)$, m and n have the same meaning as equation (2). TDH and TDV represent horizontal and vertical texture characteristics respectively. If horizontal texture dominates, that is, $TDH < TDV$, we select 16x8 coding mode and set coding mode 8x16 disable, otherwise we will select 8x16 coding mode and set coding mode 16x8 disable. So we only select one coding mode to motion estimation between 16x8 and 8x16 coding modes.

For complex texture macroblock, for example the case of $Var > TH_{(MB)}$, we need to detect its local motion activity which is defined as:

$$l_i = |x_i - \bar{x}| + |y_i - \bar{y}| \qquad (5)$$

$$L = \max\{l_i\} \qquad (6)$$

Where (x_i, y_i) is motion vector of 4x4 blocks surrounding the current macroblock, $(\bar{x}, \bar{y})$ is the average of motion vectors (x_i, y_i), subscript i is the index of 4x4 block. L is the prediction of local motion activity for current macroblock. If L is less than threshold TH_{LMA}, all sub-blocks of current macroblock may locate in one object, and only larger block coding modes such as 16x16, 16x8, 8x16 and 8x8 are selected to perform motion estimation and compensation, remaining smaller coding modes are entirely set disable.

If 8x8 coding mode is set enable, the similar process as detailed above is used to select coding modes for each 8x8 block, except that the threshold is different. The proposed method is detailed as follows:

Step1: Initialization. Set all inter coding modes enable;

Step2: Calculate variance of current macroblock Var

based equation (2). if $Var < TH_{(MB)}$, all coding modes except 16x16 inter coding modes are set disable, then go to step6, otherwise go to step 3;

Setp3: Select one of 16x8 and 8x16 coding modes based on macroblock texture directions, then go to step4;

Step4: Calculate local motion activity L . if $L < TH_{LMA}$, then set 8x4, 4x8 and 4x4 coding modes disable, then go to step6, otherwise go to step 5;

Step5: For every 8x8 block, select suitable coding modes using above method.

Step6: Perform motion estimation and RD optimization for current macroblock using available coding modes, and then select the optimal coding modes as final coding modes.

IV. SIMULATION

To test the efficiency of our proposed method, our proposed method was integrated within JVT reference software JM75C. To emphasize on the stability of our method, we select five CIF resolution sequences for test which are Mobile, Stefan, Paris, Akiyo and Mother. The first two sequences are relatively difficult to compress because they contain not only local motion but also global motion. Other three sequences only consist of local motion objects and static backgrounds, but their texture characteristics are different and representative. In the simulation, we encoded the sequences at 30fps, the CAVLC entropy coder was used for all our tests, a search rang of 32, 5 reference frames, QP is set 36 and encode 30 frames for each test sequence. Simulation results are shown in Table.2. From these results we can observe that coding efficiency is only slightly decreased while about half of all coding modes are eliminated

Table.2. Performance of the proposed algorithm (dB)

Sequence	mobile	stefan	paris	akiyo	mother
JM75C	28.09	29.67	29.75	34.93	34.28
Proposed	28.01	29.59	29.69	34.90	34.25
Δ PSNR	-0.08	-0.08	-0.06	-0.03	-0.03

V. CONCLUSION

In this paper, we propose a method to eliminate some redundant coding modes based on the texture analysis and local motion activity, which speed up the process of multi-block type motion estimation. Simulation results show that the proposed method can reduce about half of coding modes while keep similarity coding efficiency. Furthermore this method can be used with any fast motion algorithm.

REFERENCE

[1] *Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC)*, in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T, JVTG050, 2003.

[2] *Video Codec for Audiovisual Services at p_64 kbit=s ITU-T Recommendation H.261*, Version 1, ITU-T, ITU-T Recommendation H.261 Version 1, 1990.

[3] *Video Coding for Low Bit Rate Communication*, ITU-T, ITU-T Recommendation H.263 version 1, 1995.

[4] *Generic Coding of Moving Pictures and Associated Audio Information - Part 2: Video*, ITU-T and ISO/IEC JTC 1, ITU-T Recommendation H.262 and ISO/IEC 13 818-2 (MPEG-2), 1994.

[5] *Coding of audio-visual objects—Part 2*: Visual, in ISO/IEC 14496-2 (MPEG-4 Visual Version 1), Apr. 1999.

[6] T. Wiegand, G. J. Sullivan, G. Bjontegaard, and A. Luthra, "*Overview of the H.264/AVC video coding standard,*" IEEE Trans. Circuits Syst. Video Technol., vol. 13, pp. 560–576, July 2003.

[7] JVT reference software JM75c http://bs.hhi.de/~suehring/tml/download/jm75c.zip

[8] Chang, A, Au, O.C, Yeung, Y.M," *A Novel Approach to Fast Multi-Block Motion Estimation for H.264 Video Coding*", Proceedings of 2003 International Conference on Multimedia and Expo (ICME), Volume: 1, Pages: 105 – 108, 6-9 July 2003.

[9] Peng Yin, Hye-Yeon Cheong Tourapis, Tourapis, A.M, Boyce, J, "*Fast mode decision and motion estimation for JVT/H.264*", Proceedings of 2003 International Conference on Image Processing, Volume: 3, Pages:853 – 856, Sept. 14-17, 2003

[10] Yu-Wen Huang, Bing-Yu Hsieh, Tu-Chih Wang, Shao-K Chien, Shyh-Yih Ma, Chun-Fu Shen, Liang-Gee Chen, "*Analysis and reduction of reference frames for motion estimation in MPEG-4 AVC/JVT/H.264*", Proceedings of 2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, Volume: 3 , Pages: 145 - 148. 6-10 April 2003.