

An Efficient Mode Selection Algorithm for H.264

Lu Lu¹, Wenhan Wu², and Zhou Wei³

1 South China University of Technology, Institute of Computer Science,
Guangzhou 510640, China
lul@scut.edu.cn

2 South China University of Technology, Institute of Computer Science,
Guangzhou 510640, China
wuwenhan_2005@yahoo.com.cn,

3 South China University of Technology, Institute of Computer Science,
Guangzhou 510640, China
vivito@126.com

Abstract. H. 264 video coding standard introduces motion estimation with multiple block sizes to achieve a considerably higher coding efficiency than other video coding algorithms. However, this comes at the greatly increased computing complexity at the encoder. In this paper, a method is proposed to eliminate some redundant coding modes that contribute very little coding gain based on analysis of macroblock detail-level and texture directions. The simulation results show that the algorithm can remarkably decrease the complexity at the encoder while keeping satisfying coding efficiency.

1 Introduction

The JVT[1] (Joint Video Team) introduced a number of advanced features in H.264 or MPEG-4 AVC. These improvements achieve significant gains in encoder and decoder performances. One of the new features is multi-mode selection, which is the subject of this paper. In the H.264 coding algorithm, blockmatching motion estimation is an essential part of the encoder to reduce the temporal redundancy between frames. H.264 supports motion estimation and compensation using different block sizes ranging from 16x16 to 4x4 luminance samples, which is shown in Fig1, with many options between the two. The luminance component of each macroblock can be split by four ways: 16x16, 16x8, 8x16 and 8x8. Each of the submacroblock 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: 8x8, 8x4, 4x8 or 4x4, which are called macroblock sub-partitions. These partitions and

subpartitions give rise to a larger number of possible combinations within each macroblock[4].

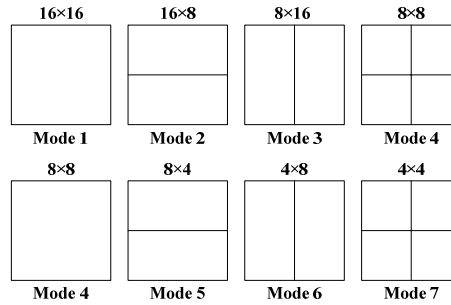


Fig. 1. Inter-prediction modes

H.264 standard uses computationally intensive Lagrangian rate-distortion (RD) optimization to choose the best block size for a macroblock[7]. The general equation of Lagrangian RD optimization is given as:

$$J_{mode} = D + \lambda_{mode} \cdot R \quad (1)$$

where J_{mode} is the rate-distortion cost (RD cost) and λ_{mode} is the Lagrangian multiplier; D is the distortion measurement between original macroblock and reconstructed macroblock located in the previous coded frame, and R reflects the number of bits associated with choosing the mode and macroblock quantizer value, Q_p , including the bits for the macroblock header, the motion vector(s) and all the DCT residue blocks.

The computational complexity required by motion estimation, however, 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[5], 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. Though it provides the best coding efficiency, the computational complexity is obviously much too high. In order to reduce the intensive computational requirement, Andy Cbang etc. proposed fast multi-block motion estimation[2]. They adopt an approach of early termination by skipping searching for mode 16x8 and mode 8x16, if the performance of mode 16x16 is “good enough”, otherwise all coding modes will be performed. This method only considers three coding modes which are 16x16, 16x8 and 8x16 inter coding modes. Another approach, proposed by Andy C. Yu, is based on estimating block detail complexity[6]. It is an effective way judging by his simulation results, but there is more a critical factor, texture direction, which he does not think about and is also can be useful to significantly improve coding efficiency.

In this paper, we propose a method to eliminate some redundant coding modes by estimating block detail and texture direction. The paper will be organized as follows. The proposed algorithm will be described in detail in section 2. Section 3 shows the simulation and the results. Finally, a conclusion will be given in section 4.

2 PROPOSED ALGORITHM

2.1 Estimating block details

Table 6. shows the observations on how selected modes relate sequence characteristics.

sequence	Skip	16×16	16×8	8×16	8×8	Intra16	Intra4
Container	75.8	10.4	3.5	2.7	7.3	0.3	0.0
Foreman	23.7	39.9	39.9	7.3	7.6	7.3	9.3
Bus	3.5	22.0	12.1	14.4	40.5	1.0	5.5
Mobile	4.5	31.3	7.1	6.1	49.7	0.0	0.3

IPPP, 5 reference frames, CABAC, CIF Format

The choice of partition size has a significant impact on compression performance. In general, according to Tab1, large partition sizes are appropriate for homogeneous areas of the frame and small partition sizes may be beneficial for detailed areas.

We derive an approach based on summing the total energy of the AC coefficients to estimate the block detail. The AC coefficients can be obtained from the DCT coefficients of each block. The definition is:

$$E_{AC} = \sum_{u=1}^{M-1} \sum_{v=1}^{N-1} (F(u, v))^2 \tag{2}$$

From (2), EAC, the total energy of the AC components of an M×N block is the sum of all the DCT coefficients, F(u,v), except for the DC component, u = 0 and v = 0.

$$F(u, v) = c(u)c(v) \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \cos\left[\frac{(2x+1)u\pi}{16}\right] \cos\left[\frac{(2y+1)v\pi}{16}\right] \tag{3}$$

where,

$$c(u), c(v) = \begin{cases} \sqrt{\frac{1}{M}}, \sqrt{\frac{1}{N}} & \text{for } u, v=0 \\ \sqrt{\frac{2}{M}}, \sqrt{\frac{2}{N}} & \text{for } u, v \neq 0 \end{cases} \tag{4}$$

According to the energy conservation principle, the total energy of an M×N block is equal to the accumulated energy of its DCT coefficients. Thus, (3) can be further simplified as

$$E_{AC} = \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} (f^2(x, y)) - \frac{1}{MN} \left[\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} f(x, y) \right]^2 \quad (5)$$

where the first term is the total energy of the image intensities within an $M \times N$ block, and the second term represents the mean square intensity. Equation (6) clearly shows that the energy of the AC components of a macroblock can be represented by the variance.

Evaluating the maximum sum of the AC components is the next target. By definition, the largest variance is obtained from the block comprising checkerboard pattern in which every adjacent pixel is the permissible maximum and minimum value. Thus, E_{\max} , the maximum sum of AC components of an $M \times N$ block is

$$E_{\max} = MN \frac{f_{\max}^2(x, y) + f_{\min}^2(x, y)}{2} - \frac{MN}{4} [f_{\max}(x, y) + f_{\min}(x, y)]^2 \quad (6)$$

Note that E_{\max} can be calculated in advance. Then the criterion to assess the complexity of a macroblock detail is

$$r_d = \frac{\ln(E_{AC})}{\ln(E_{\max})} \quad (7)$$

In total, 7 different block sizes are recommended by H.264 for P-frames, namely, 16×16 , 16×8 , 8×16 , 8×8 , 8×4 , 4×8 , 4×4 as well as SKIP, and other two INTRA prediction modes, I4MB and I16MB. However, in our complexity measurement, there are only 3 categories, which are denoted as MD16 category, MD8 category, and MD4 category, respectively.

The proposed algorithm provides a recursive way to decide the complexity of each macroblock. Firstly, a macroblock of 16×16 pixels is examined with the first piecewise equation in (7). An LDB category is given if it is recognized as being a homogenous macroblock. Otherwise, the macroblock is decomposed into 4 blocks of 8×8 pixels. Note that an 8×8 block is recognized as high-detailed if it satisfies two conditions: (a) the RB in (9) is greater than 0.7, and it is decomposed into four 4×4 block, and (b) one of its four decomposed 4×4 blocks is highdetailed as well. If an 8×8 block satisfies the first condition but not the second, it is still recognized as low-detailed. After checking all the 8×8 blocks, an MD8 category is given to a macroblock which possesses more than two high-detailed blocks, otherwise the MD4 category is assigned. Table 1 displays the relationship between the three categories in the proposed algorithm and the 9 inter-frame prediction modes. It is observed that the MD16 category covers the least number of prediction modes, whereas the MD4 category contains all the available modes. The table further indicates that the higher detailed the macroblocks are, the more prediction modes the proposed algorithm has to check.

The function of the natural logarithm is to linearize both E_{\max} and E_{AC} such that the range of r_d can be uniformly split into 10 subgroups. In our evaluation, a macroblock that has the $r_d > 0.7$, is considered to be a high-detailed block.

Table 2. Block categories and corresponding modes

Detail Level	Enabled Modes
LDB	16×16
MDB	16×16, 16×8, 8×16, 8×8
HDB	8×8, 8×4, 4×8, 4×4

2.2 Detecting block texture direction

It is obvious that 16×8 and 8×4 are appropriate for blocks with horizontal texture characteristics, and 8×16, 4×8 could be selected when there is a block with vertical texture direction. Texture characteristics can be estimated based on the following equations:

$$D_h = \sum_{x=0}^{M-1} \sum_{y=1}^{N-1} [f(x, y) - f(x, y-1)]^2 \tag{8}$$

$$D_v = \sum_{x=1}^{M-1} \sum_{y=0}^{N-1} [f(x, y) - f(x-1, y)]^2 \tag{9}$$

If $D_h > D_v$, the block tends to have horizontal texture direction. When $D_v > D_h$, the block tends to be with vertical texture direction. The maximum values of D_h and D_v are expressed as following respectively:

$$D_{hmax} = M(N-1)d^2 \tag{10}$$

$$D_{vmax} = N(M-1)d^2 \tag{11}$$

where d is the difference between maximum and minimum value of $f(x, y)$. In order to make a further step to eliminate redundant modes, detection of block texture direction should be performed, which uses the following equation:

$$r_h = \frac{\ln(D_h)}{\ln(D_{hmax})} \tag{12}$$

$$r_v = \frac{\ln(D_v)}{\ln(D_{vmax})} \tag{13}$$

If $r_h > r$, 8×16 and 4×8 will be eliminated. If $r_v > r_h$, 16×8 and 8×4 will be eliminated.

2.3 Algorithm

- Step 1:** Calculate r_d in the current block. If $r_d < 0.3$, go to step 2. If $0.3 < r_d < 0.7$, go to If $r_d > 0.7$, and go to step 4.
- Step 2:** Select 16×16 as the only enabled mode, go to step 5.
- Step 3:** Disable 8×8 , 4×8 , 4×4 , and go to step 5.
- Step 4:** Enable all of the modes, and go to step 5.
- Step 5:** Calculate r_h and r_v in the current block. If $r_h > r_v$, disable 8×16 and 4×8 . If $r_v > r_h$, disable 16×8 and 8×4 . Go to step 6.
- Step 6:** Calculate Jmode with every enable mode, plus SKIP and INTRA. And then select the best mode.

3 SIMULATION RESULTS

To test the efficiency of our proposed method, the proposed method was integrated within JVT reference software JM75C. Five sequences were selected for test, which were Container, Foreman, News, Mobile. Their texture characteristics are different and representative. Major parameters setting for the test is shown in Table 3.

Table 3. Major parameters setting in simulation

GOP structure	IPPP
Use Hadmard	Used
Frame Rate(HZ)	15
NumberReferenceFrames	1
InterSearch AxB	All are enable
SymbolMode	CABAC
PartitionMode	No DP
QP	30
UseConstrainedIntraPred	0
MVResolution	1/4
SearchRange	16
RateControl	Disable

Table 4 is a summary of the performance of the proposed mode selection algorithm. It is demonstrated that the proposed algorithm can save up to 28.2% encoding time as compared to JM75c. On average, there is a degradation of 0.08dB. As to compression ratio, the proposed algorithm produces an average slightly higher bit rates than H.264. The degradations and the bit differences are due to less accurate prediction in the proposed algorithm. Nevertheless, the degradations are still within an acceptable range because human visual perception is unable to distinguish the PSNR difference of less than 0.2dB.

Table 4. Simulation results of the proposed algorithm compared to JM75 in terms of PSNR, bit rate and time saved

Sequence	PSNR(db)			BITS(bit)			TIME(%)		
	JM	Proposed Algorithm	Δ PSNR	JM	Proposed Algorithm	Δ BITS	JM	Proposed Algorithm	Δ Time
Container	35.97	35.92	-0.08	189944	188712	-0.65%	279.056	226.036	-19.0%
Foreman	35.53	35.48	-0.05	577120	575432	-0.03%	281.221	211.354	-24.8%
News	36.72	36.65	-0.09	364680	367624	0.08%	280.968	201.620	-28.2%
Mobile	33.07	32.93	-0.10	4227024	4221848	-0.12%	340.858	305.639	-10.3%

4 CONCLUSION

In this paper, we propose a method to eliminate some redundant coding modes based on analysis of macro block detail-level and texture directions., which speeds up the process of multi-mode selection. The simulation results show that the algorithm can remarkably decrease the complexity at the encoder while keeping satisfying coding efficiency.

REFERENCES

1. Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, Final committee draft of Joint Video Specification (ITU-T Rec. H.264 and ISO/IEC 14496-10 AVC)[S] (2002).
2. A. Chang, O.C. Au, Y.M. Yeung, "A novel approach to fast multi-block motion estimation for H.264 video coding[A]", Multimedia and Expo, 2003. *ICME'03 Proceedings 2003 International Conference.* (2003). Volume 1, 6-9.
3. A.C. Yu, "Efficient Block-size Selection Algorithm for Inter-Frame Coding", H.264/MPEG-4 AVC[A]. ICASSP, 2004. *IEEE. International Conference on Volume 3*, 17-21 (May 2004) Page(s):iii - 169-72 vol.3.
4. F.S. Yan, "Fast mode selection based on texture analysis and local motion activity", H.264/AVC [A] ICCAS 2004. *2004 International Conference on Volume 1*, 27-29 (June 2004) Page(s):539 - 542 Vol.1
5. JVT reference software JM75c [CP]. <http://bs.hhi.de/~suehring/tml>.
6. A.C. Yu, Efficient Block-size Selection Algorithm for Inter-Frame Coding, H.264/MPEG-4 AVC [C]. ICASSP, 2004, *International Conference on Volume 3*, 17-21: 69-72.
7. T. Wiegand, H. Schwarz,, A. Joch, R. Kossentini, J.G. Sullivan, Rate-constrained Coder Control and Comparison of Video Coding Standards , *IEEE Trans. Circuits Syst. Video Technol*, Vol13 (July 2003) , pp.688-703