

# EFFICIENT CODING MODE DECISION IN MPEG-4 PART-10 AVC/H.264 MAIN PROFILE

*Inchoon Choi, Jeyun Lee, and Byeungwoo Jeon*

School of Information and Computer Engineering, Sungkyunkwan University  
300 Chunchun-dong, Jangan-gu, Suwon, 440-746, KOREA  
sonne@ece.skku.ac.kr, jeyun22@ece.skku.ac.kr and bjeon@yurim.skku.ac.kr

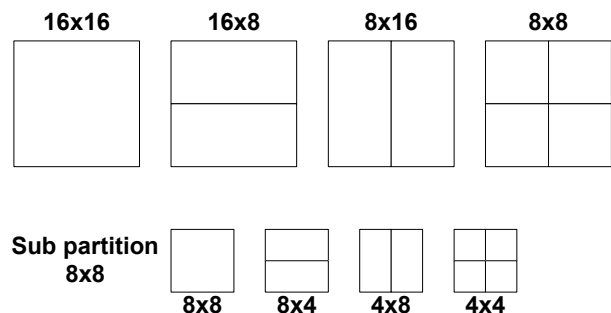
## ABSTRACT

MPEG-4 Part-10 AVC/H.264 provides much improved coding efficiency. The powerful rate-distortion optimization for adaptive motion compensation employing variable block sizes investigates all possible coding modes of a block to decide the best coding mode. This achieves highest possible coding efficiency, but it only comes at much higher computational complexity. In this paper, we propose two new techniques – ‘early SKIP mode decision in P- and B-slices’ and ‘selective intra mode decision’ - which can significantly reduce the computational complexity. Simulation results verify that the proposed method makes it possible to reduce the encoding time by about 66% without any perceivable loss in coding rate and quality.

## 1. INTRODUCTION

The newest video coding standard, MPEG-4 Part-10 AVC/H.264 [1], is developed by the joint video team of ISO/IEC MPEG and ITU-T VCEG. It employs several powerful coding techniques such as variable block-size motion compensation, motion vector of quarter-pel accuracy, multiple reference pictures, intra coding with several different spatial prediction directions, and so on. These powerful coding tools altogether provide up to more than twice of compression compared to the MPEG-4 simple profile at the same subjective quality. However, its encoding process is about 16 times more complex than the MPEG-4 simple profile [2]. Under the variable block-size motion compensation scheme, a macroblock of size 16X16 is scrutinized whether its finer segmentation into MB16x8, MB8x16, MB8X8, and SUB8x8 is better in terms of coding efficiency. In the SUB8x8 mode, each 8x8 can be further divided into four sub-blocks of sizes 8x8, 8x4, 4x8, or 4x4. Figure 1 shows all the available segmentation of a macroblock. H.264 employs the rate-distortion optimization method to find out the best coding mode of a macroblock, which requires a lot of

computation since it tests encoding with all possible coding modes of inter and intra coding, and calculates their RD costs to choose the mode having the minimum cost. Calculation of the RD costs in H.264 needs to execute both forward and backward process of integer transform, quantization, and variable length coding since the value of the rate is available only after the integer transform, quantization, and variable length coding, and the value of distortion is also obtained only through the backward process of inverse quantization and inverse transform. The maximum number of computing RD cost values for mode decision is 768[3].



**Figure 1. Variable block size for motion compensation**

H.264 has baseline, main, and extended profiles. The main profile supports interlaced coding, B slices, weighted prediction, context-based arithmetic coding (CABAC) in addition to the coding methods of baseline profile except the arbitrary slice order (ASO), the flexible macroblock ordering (FMO), and the redundant slices (RS). The B slice coding technique uses two motion vectors of list 0 and list 1, and two reference pictures at multiple reference frame memory. Therefore, the main profile employing the B-slice coding requires even more computation in its motion search and also in the RD optimization than the baseline profile having only P- and I-slice.

## 2. EARLY DETECTION OF SKIP MODE

In most video sequences, many macroblocks finally end up with being decided as skipped after all the computationally expensive RD optimization since they belong to background or motionless object. Furthermore, many real world video indeed contain substantial amount of background. Therefore it will be very efficient if one can decide in advance whether the given macroblock belongs to this category without going through the full expensive RD optimization process. Therefore, we propose to check the SKIP conditions first, and when it satisfies the conditions, then to decide the SKIP mode as the selected macroblock coding mode. This decision of SKIP mode at early stage makes it possible to omit all the remaining RD calculation, motion-compensation, and intra predictions. The condition of a macroblock to be a skip mode depends on the slice type. In P-slice, the skipped macroblock should satisfy following conditions all together [1]:

- 1) the best motion compensation block size is 16x16
- 2) the reference frame is the one closest in the reference frame memory to the current picture
- 3) the motion vector is the same as its predicted motion vector (PMV)
- 4) its transform coefficients are all quantized to zero

To check these conditions in P-slice, first of all, one needs to estimate the motion vector and reference frame for the given macroblock. Subsequently, RD cost of 16x16 macroblock is computed. If the conditions are all satisfied, the other coding modes are simply put aside. In case of B-slice, by the way, the condition is different as follows [1]:

- 1) the best motion compensation block size is 8x8
- 2) the reference frames and the motion vectors are the same as the ones decided under the direct mode [1]
- 3) its transform coefficients are all quantized to zero

In the same way as P-slice, the proposed method checks whether the mode of current macroblock can be decided as SKIP in B-slice. To do this, the macroblock is first encoded in the direct mode, and subsequently, RD cost of the macroblock is computed and checked for the B-skip conditions. If the conditions are satisfied, the mode of this block is decided as the SKIP.

## 3. SELECTIVE INTRA CODING

As other video coding standards, H.264 can also have intra macroblock in P or B slices. If the given macroblock does not satisfy P or B SKIP conditions, H.264 should compute RD costs of all intra modes (intra 16x16 and intra 4x4) that involve testing all intra prediction directions. This process is very complex and the number of computing RD cost values is about 5 times higher than the case of inter modes [3]. However, as statistical data of

intra mode indicates, the probability for a macroblock to have an intra mode in P slice is maximum 0.09 and on average 0.03, although the exact figure depends on specific input video characteristics [5]. The small probability of intra mode in real video sequences suggests that the current practice of deciding the best intra mode first and subsequent decision of a inter mode in the reference software of H.264 [1] may have a certain limit in reducing the computational complexity. Therefore, we propose a more efficient mode decision method that investigates intra modes *after* deciding the best inter mode. Furthermore, it investigates various intra modes only when such investigation is believed to be certainly worthwhile. In this way, the encoder can avoid quite a number of unfruitful calculation of RD cost values. The underlying principle of inter mode is to make use of the temporal correlation between current and reference pictures, and the intra mode is to utilize spatial correlation between current and its adjacent blocks. Therefore, if spatial correlation of current block is higher than the temporal correlation, the block has higher probability of being an intra block. A proper decision between the intra and the inter modes need some objective measure of spatial and temporal correlation. To avoid additional computational requirement, the measure will be desirable if it can utilize already available intermediate result. We propose here to use *the average boundary error (ABE)* between pixels at boundary of the current and its adjacent encoded blocks under the best inter mode as an indicative of degree of spatial correlation. Also we propose to use *the average rate (AR)*, i.e., the average number of bits consumed to encode the motion-compensated residual data under the best inter mode as an indicative of degree of temporal correlation. So, we compare the average rate for the best inter mode and the average boundary error for the current block [3]. Figure 2 shows pixels involved in computing the average boundary error.

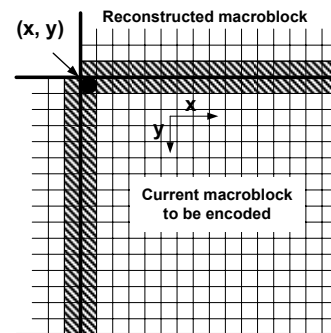


Figure 2. Illustration of boundary error

AR is the average number of bits needed to encode residual data of inter macroblock. We can compute AR as following:

$$AR = \frac{1}{384} (\# \text{ bits of texture data}) \times \lambda \quad (1)$$

where  $\lambda = 0.85 \times 2^{Qp/3}$ ,  $384 = (16 \times 16)_{Luma} + (8 \times 8)_{Chroma} \times 2$

If AR is larger than ABE, we do not need to calculate RD cost of intra modes. Otherwise, H.264 encoder calculates RD cost of intra modes, and compare its value with that of the best inter mode. To compute the ABE value, we have to check adjacent block whether it is available or not. Therefore, there can be three possible cases. In case 1, top and left blocks are available. In case 2 or 3, only top or left block is available. In the equations below,  $Y_{Orig}, Cb_{Orig}, Cr_{Orig}$ , and  $Y_{Rec}, Cb_{Rec}, Cr_{Rec}$  are respectively the original pixel data of current macroblock to be encoded, and the reconstructed pixel data of adjacent blocks.

**Case 1 : [Left: not available, Upper: not available]**

If AR=0, skip testing intra modes. Otherwise, set ABE=0.

**Case 2 : [Left: not available, Upper: available]**

$SBE(\text{Sum of Boundary Error}) =$

$$\begin{aligned} & \sum_{i=0}^{15} \left[ \left| Y_{Orig}(x+i, y) - Y_{Rec}(x+i, y-1) \right| \right] \\ & + \sum_{i=0}^7 \left[ \left| Cb_{Orig}(cx+i, cy) - Cb_{Rec}(cx+i, cy-1) \right| \right] \\ & + \sum_{i=0}^7 \left[ \left| Cr_{Orig}(cx+i, cy) - Cr_{Rec}(cx+i, cy-1) \right| \right] \end{aligned} \quad (2)$$

$$\therefore ABE = \frac{1}{32} SBE, \text{ where } 32 = 16_{Luma} + 8_{Chroma} \times 2$$

**Case 3 : [Left: available, Upper: not available]**

$SBE(\text{Sum of Boundary Error}) =$

$$\begin{aligned} & \sum_{i=0}^{15} \left[ \left| Y_{Orig}(x, y+i) - Y_{Rec}(x-1, y+i) \right| \right] \\ & + \sum_{i=0}^7 \left[ \left| Cb_{Orig}(cx, cy+i) - Cb_{Rec}(cx-1, cy+i) \right| \right] \\ & + \sum_{i=0}^7 \left[ \left| Cr_{Orig}(cx, cy+i) - Cr_{Rec}(cx-1, cy+i) \right| \right] \end{aligned} \quad (3)$$

$$\therefore ABE = \frac{1}{32} SBE, \text{ where } 32 = 16_{Luma} + 8_{Chroma} \times 2$$

**Case 4 : [Left: available, Upper: available]**

$SBE(\text{Sum of Boundary Error}) =$

$$\begin{aligned} & \sum_{i=0}^{15} \left[ \left| Y_{Orig}(x, y+i) - Y_{Rec}(x-1, y+i) \right| \right] \\ & + \left| Y_{Orig}(x+i, y) - Y_{Rec}(x+i, y-1) \right| \\ & + \sum_{i=0}^7 \left[ \left| Cb_{Orig}(cx, cy+i) - Cb_{Rec}(cx-1, cy+i) \right| \right] \\ & + \left| Cb_{Orig}(cx+i, cy) - Cb_{Rec}(cx+i, cy-1) \right| \\ & + \sum_{i=0}^7 \left[ \left| Cr_{Orig}(cx, cy+i) - Cr_{Rec}(cx-1, cy+i) \right| \right] \\ & + \left| Cr_{Orig}(cx+i, cy) - Cr_{Rec}(cx+i, cy-1) \right| \end{aligned} \quad (4)$$

$$\therefore ABE = \frac{1}{64} SBE, \text{ where } 64 = (16+16)_{Luma} + (8+8)_{Chroma} \times 2$$

The proposed mode decision is summarized in Figure 3.

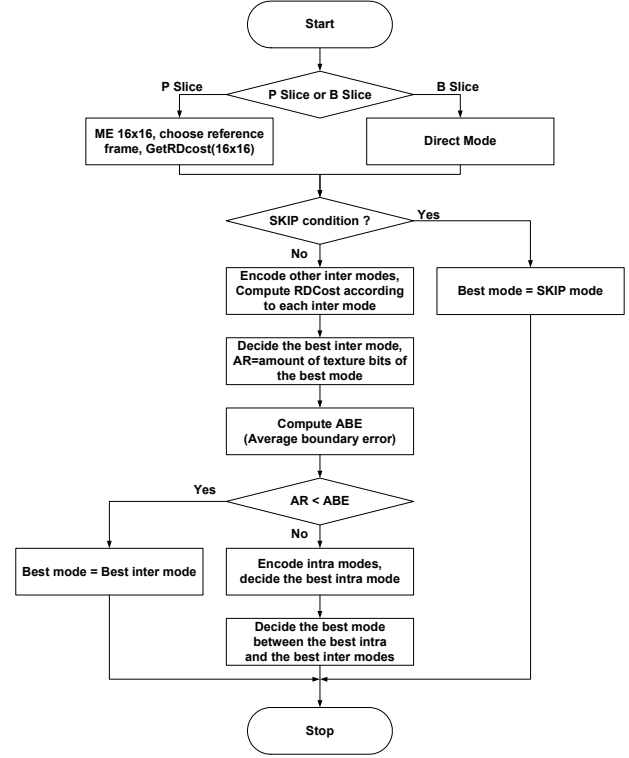


Figure 3. Flowchart of proposed fast mode decision

#### 4. EXPERIMENTAL RESULTS

The video sequences and coding conditions for the test are shown in Table 1. The proposed methods are tested using the JM 7.5c encoder with coding option chosen in the main profile as in Table 1.

Table 1. Test condition

Codec		JM 7.5c
MV search range		$\pm 16$
QP		28, 34, 36, 40
Number of Reference		5
Coding option		Rate distortion optimization, Hadamard transform, CABAC, IBBP coding
Test sequence	CONTAINER	QCIF (300 frames)
	FOREMAN	QCIF (300 frames)
	NEWS	QCIF (300 frames)
	MOBILE	CIF (300 frames)
	SILENT	QCIF (300 frames)
	PARIS	CIF (300 frames)
	TEMPETE	CIF (260 frames)

Table 2~5 show the experimental results of the proposed methods where  $TS$  indicates the average time saving in encoding process:

$$TS = \frac{T_{ref} - T_{proposed}}{T_{ref}} \times 100[\%] \quad (5)$$

where  $T_{ref}$  and  $T_{proposed}$  are respectively the encoding times of JM 7.5c software, and its modified software according to the proposed method.

**Table 2. Test result with early detection of P-skip**

	TS [%]	BDBR[%]	BDPSNR[dB]
CONTAINER	6.39	0.10	-0.01
FOREMAN	2.20	0.22	-0.01
NEWS	6.58	0.35	-0.02
MOBILE	1.02	0.22	-0.01
SILENT	10.47	-0.04	0.00
PARIS	10.92	0.24	-0.01
TEMPETE	3.85	0.12	0.00
Average	5.92	0.17	-0.01

**Table 3. Test result with early detection of B-skip**

	TS [%]	BDBR[%]	BDPSNR[dB]
CONTAINER	68.20	-0.34	0.03
FOREMAN	50.75	1.11	-0.06
NEWS	62.13	-0.61	0.03
MOBILE	40.59	2.51	-0.10
SILENT	47.17	-0.50	0.02
PARIS	44.18	-0.48	0.02
TEMPETE	38.96	0.57	-0.02
Average	50.28	0.32	-0.01

**Table 4. Test result with selective intra coding**

	TS [%]	BDBR[%]	BDPSNR[dB]
CONTAINER	20.38	0.69	-0.05
FOREMAN	19.46	0.77	-0.04
NEWS	20.21	0.50	-0.03
MOBILE	23.99	0.21	-0.01
SILENT	21.64	0.60	-0.03
PARIS	24.56	0.36	-0.02
TEMPETE	23.69	0.41	-0.02
Average	21.99	0.51	-0.03

**Table 5. Experimental result with combined method**

	TS [%]	BDBR[%]	BDPSNR[dB]
CONTAINER	80.80	0.07	0.00
FOREMAN	60.45	1.86	-0.09
NEWS	76.12	0.15	-0.01
MOBILE	56.17	2.55	-0.10
SILENT	66.83	-0.06	0.00
PARIS	65.05	0.26	-0.01
TEMPETE	53.65	0.95	-0.03
Average	65.58	0.83	-0.04

The results show that the proposed method is very effective in reducing the encoding time - the total encoding time is reduced by about 66%. This means that the modified H.264 encoder is about 3 times faster than the reference software JM 7.5c. However its loss in terms of BDBR and BDPSNR [7] is very negligible – it is only 0.83% and 0.04dB, respectively.

## 5. CONCLUSION

The current practice of H.264 encoding relies on the rate distortion optimization method for the coding mode decision which demands huge computational complexity. In order to attain highest possible coding efficiency, the RD optimization method simply tries encoding all coding modes at every macroblock. For this reason, the complexity of H.264 encoder is dramatically increased. In this paper, we proposed two efficient mode decision techniques – ‘early SKIP mode decision’ and ‘selective intra mode decision’. The proposed method provides considerable reduction in computational complexity with only negligibly small coding loss.

## REFERENCES

- [1] Joint Video Team of ISO/IEC MPEG & ITU-T VCEG, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification”.
- [2] Minhua Zhou, “Evaluation and Simplification of H.26L Baseline Coding Tools,” ITU-T Q.6/16, Doc. #JVT-B030, 2002.
- [3] B. Jeon and J. Lee, “Fast mode decision for H.264,” ITU-T Q.6/16, Doc. #JVT-J033, 2003.
- [4] J. Lee and B. Jeon, “Fast Mode Decision for H.264 with Variable Motion Block Sizes,” Lecture Notes in Computer Science, vol.2869, pp. 723~730, Nov. 2003.
- [5] J. Lee and B. Jeon, “Pruned Mode Decision based on Variable Block Sizes Motion Compensation for H.264,” Lecture Notes in Computer Science, vol.2899, pp. 410~418, Nov. 2003.
- [6] G. Sullivan and T. Wiegand, “Rate-Distortion Optimization for Video Compression,” IEEE Signal Proc. Magazine, Vol.15, pp.74~90, 1998.
- [7] G. Bjontegaard, “Calculation of Average PSNR Differences between RD-curves,” Doc. VCEG-M33, Apr. 2001.
- [8] Iain E.G. Richardson, H.264 and MPEG-4 Video Compression, Wiley, pp. 160~223, 2003.
- [9] T. Wiegand and G. Sullivan, “Overview of the MPEG-4 PART-10 AVC/H.264 Video Coding Standard,” IEEE Trans-CAS-VT, Vol.13, pp.560~576, July 2003.