FAST MODE DECISION ALGORITHM FOR H.264 BASED ON MOTION COST

Geun-Yong Kim, Seung-Hwan Kim, Hee-Soon Kim, and Yo-Sung Ho

Gwangju Institute of Science and Technology (GIST)
1 Oryong-Dong, Buk-Gu, Gwangju, 500-712, Korea
Phone: +82-062-970-2258, FAX: +82-062-970-2247
Email: {gykim, kshkim, hskim03, hoyo}@gist.ac.kr

ABSTRACT

The H.264 standard achieves higher compression efficiency than previous video coding standards with the rate-distortion optimized (RDO) method for mode decision; however, its high complexity causes a large amount of encoding time. In this paper, we propose a fast mode decision algorithm using early SKIP mode decision and motion cost to reduce the candidate modes. Experimental results show that the proposed algorithm reduces the encoding time by 88% on average compared to the H.264 standard.

1. INTRODUCTION

H.264 provides high compression efficiency compared to previous video coding standards, such as MPEG-4 and H.263, mainly due to variable block-size macroblock modes. In H.264, there are seven different block sizes (16×16, 16×8, 8×16, 8×8, 8×4, 4×8 and 4×4 blocks) that are used in the macroblock for inter modes, as shown in Fig. 1. The available coding modes for the macroblock in the I-slice include: Intra4×4 prediction and Intra16×16 prediction for luminance samples, and Intra8×8 prediction for chrominance samples [1].

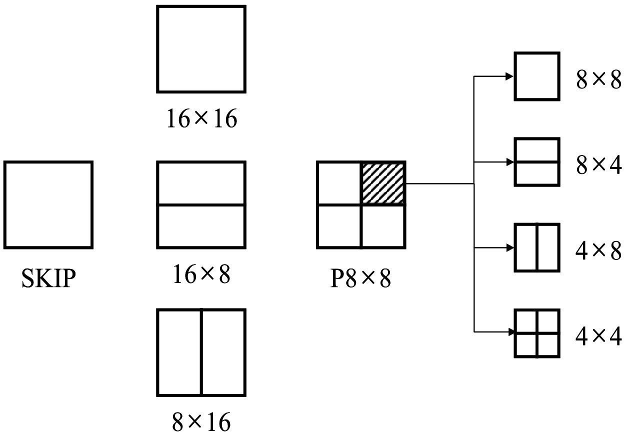


Fig. 1. Inter modes in H.264

The SKIP mode represents the case where the block size is 16×16, but no motion or residual information is coded. In the P8×8 mode, each 8×8 block can be divided into smaller blocks, such as 8×8, 8×4, 4×8, and 4×4 blocks. Except for the SKIP, Intra4×4, and Intra16×16 modes, the other inter modes require the motion estimation operation. For each macroblcok, the H.264 reference software tries all possible inter modes [2].

Motion estimation and the rate-distortion optimized (RDO) calculation are performed to find the best mode in each macroblock. The motion vector (MV) and the reference frame are selected to minimize the motion cost:

$$J_{motion}(MV, REF \mid \lambda_{motion}) = SAD(s, r(MV, REF)) + \lambda_{motion} \cdot R(MV, REF) \qquad (1)$$

where *MV* is the motion vector, *REF* denotes the reference picture, and λ_{motion} is the Lagrangian multiplier which equals to $0.85 \times 2^{(QP/3)}$. *R(MV, REF)* represents the bits used for coding motion vectors and the reference picture. *s* and *r* indicate the current and reference blocks, respectively. *R(MV, REF)* value is computed by the lookup table. *SAD* represents the sum of absolute differences defined by

$$SAD(s, r(MV, REF)) = \sum_{x \in H, y \in V} \left| s(x, y) - r(x - m_x, y - m_y) \right| \qquad (2)$$

with the motion vector (m_x, m_y), the horizontal block size *H*, and the vertical block siVe .

The following Lagrangian function is used (a) to determine the sub-block mode for the P8×8 mode, (b) to decide the prediction mode in the intra mode decision, and (c) to select the best mode.

$$J_{mode}(s, r, MODE \mid \lambda_{mode}) = SSD(s, r, MODE) + \lambda_{mode} \cdot R(s, r, MODE) \qquad (3)$$

where λ_{mode} is λ_{motion}^{2} , and *MODE* can be a sub-block mode, a prediction mode, or the best macroblock mode. *MODE* indicates a mode out of ten possible macroblock modes: {SKIP, 16×16, 16×8, 8×16, 8×8, 8×4, 4×8, 4×4, Intra4×4 and Intra16×16}. *R(s, r, MODE)* is the number of bits associated with choosing*MODE* including the bits for the macroblock header, motion vectors, and laDCT coefficients. In the H.264 reference software, this value is calculated for all possible modes, which means all encoding processes are performed. *SSD* denotes the sum of square differences between the original and reference blocks.

$$SSD(s, r, MODE) = \sum_{x \in H, y \in V} \left(s(x, y) - r(x - m_x, y - m_y) \right)^2 \qquad (4)$$

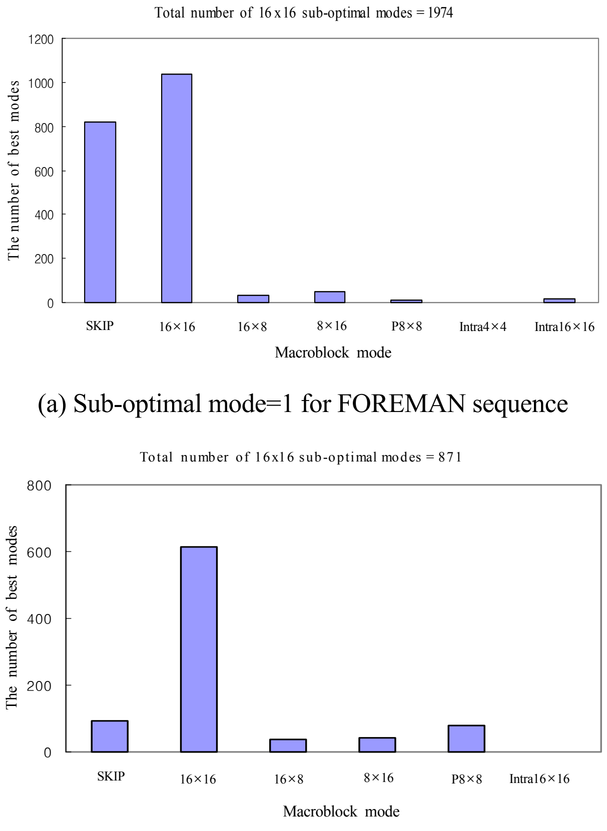
with the motion vector (m_x, m_y), the horizontal block size *H*, and the vertical block siVe .

2. OBSERVATION OF J_{motion} AND J_{mode}

For the fast mode decision, recently proposed schemes compare the rate-distortion cost between modes using certain thresholds or classified possible modes based on J_{mode} [3][4][5]. However, they does not reduce the encoding time since the motion estimation process is performed at first for every mode and motion cost is used for the mode decision.

The H.264 reference software includes the sub-optimal mode decision method, where the best mode is the mode of minimum J_{motion}. The result of the sub-optimal mode decision is worse than that of the rate-distortion method in terms of PSNR value and bit rates, since it does not count the actual coding bits.

However, this observation provides a clue for the fast mode decision. We are able to use J_{motion} for choosing candidate modes. When the sub-optimal mode is 16×16, the best mode is determined as shown in Fig. 2. The most popular best macroblcok mode is 16×16 for the FOREMAN sequence, and P8×8 for the MOBILE sequence. In our experiment, we have taken 50 frames for each test sequence.



(a) Sub-optimal mode=1 for FOREMAN sequence

(b) Sub-optimal mode=1 for MOBILE sequence

Fig. 2. Relation between J_{motion} and J_{mode} for 16×16 mode

It is found that the SKIP mode or 16×16 mode is the best mode for macroblocks in the background or smooth regions of the image. Hence, if 16×16 is the best mode among 16×16, 16×8 and 8×16 in sub-optimal mode decision, we do not need to calculate J_{motion} and J_{mode} for P8×8, since the macroblock is likely to be in the background or smooth regions. Therefore, we have proposed the fast mode decision algorithm using J_{motion} based on this experiment.

3. FAST MODE DECISION ALGORITHM

3.1 Early SKIP Mode Decision

The SKIP mode refers to the 16x16 mode where neither motion nor residual information is encoded. It has the lowest complexity in the mode decision process since no motion search is required. Table 1 shows the frequency of the SKIP mode in various sequences with 100 frames. If we determine the SKIP mode at an early stage, we can efficiently reduce the encoding time by skipping the other inter modes.

Table 1. Frequency of the SKIP mode

Sequences	Percentage (%)
NEWS	73
CARPHONE	34
FOREMAN	24
COASTGUARD	13
MOBILE	3

In order to determine whether the best macroblock mode is SKIP or not, we compare J_{mode} of 16×16 with that of SKIP.

$$J_{\mathrm{mod}e}(SKIP) \leq J_{\mathrm{mod}e}(16\times16) \tag{5}$$

If J_{mode} of SKIP is smaller than that of 16×16, the mode decision procedure is terminated and the best mode is determined as the SKIP mode.

3.2 Procedure of the Proposed Algorithm

We propose a fast mode decision algorithm using the early SKIP mode decision and J_{motion}. The proposed algorithm consists of the following steps:

Step 1: Calculate $J_{motion}(16\times16)$, $J_{mode}(16\times16)$ and $J_{mode}(SKIP)$.

Step 2: If $J_{mode}(SKIP)$ is equal to or less than $J_{mode}(16\times16)$, the best mode is determined as SKIP and mode decision procedure stops. On the other hand, go to the next step.

Step 3: Inactivate the SKIP mode since $J_{mode}(SKIP)$ is larger than $J_{mode}(16\times16)$.

Step 4: Calculate $J_{motion}(16\times8)$ and $J_{motion}(8\times16)$. Among $J_{motion}(16\times16)$, $J_{motion}(16\times8)$ and $J_{motion}(8\times16)$, the mode with minimum J_{motion} cost is the sub-optimal mode so far.

Step 5: If the sub-optimal mode is 16×16, Inactivate 16×8, 8×16 and P8×8 modes. Otherwise, go to the next step.

Step 6: Calculate J_{motion} and J_{mode} for the activated modes. The best mode is determined.

Fig. 3 shows the flowchart of the proposed fast mode decision algorithm.

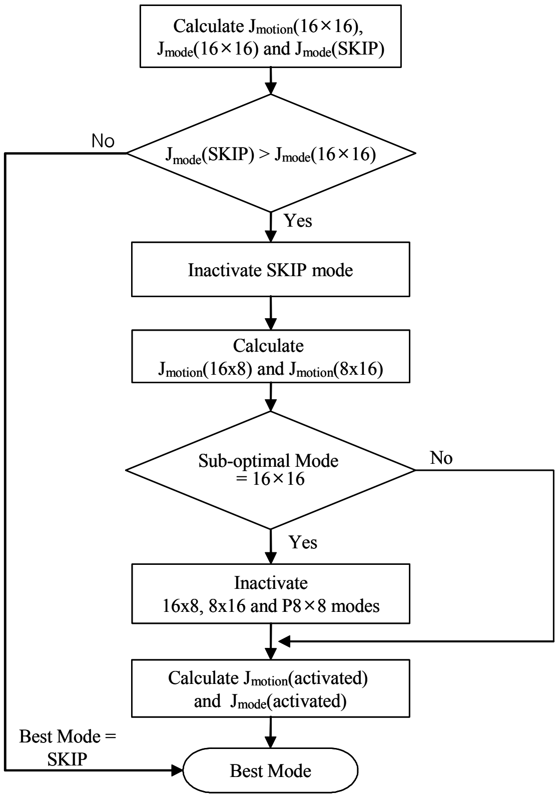

Calculate $J_{motion}(16\times16)$, $J_{mode}(16\times16)$ and $J_{mode}(SKIP)$
$J_{mode}(SKIP) > J_{mode}(16\times16)$
No
Yes
Inactivate SKIP mode
Calculate $J_{motion}(16x8)$ and $J_{motion}(8x16)$
Sub-optimal Mode = 16×16
No
Yes
Inactivate 16x8, 8x16 and P8×8 modes
Calculate $J_{motion}(activated)$ and $J_{mode}(activated)$
Best Mode = SKIP
Best Mode

Fig. 3. Flow chart of the proposed algorithm

3.3 Integration with Fast Motion Estimation

In the proposed algorithm, we can consider a fast motion estimation (FME) algorithm to obtain a speedup for the fast mode decision. If we use a good FME algorithm, we can achieve further speedup. Thus, we have used the hybrid asymmetrical-cross multi-hexagon-grid search (UMHexagonS) algorithm [6] for integer-pel motion estimation and the center-biased fractional-pel search algorithm (CBFPS) [7] for sub-pel motion estimation which are adopted in JM reference software. In FME, initial search point prediction is used hierarchically from 16×16 mode to 4×4 mode. Since the proposed fast mode decision algorithm does not break this structure, it works well with this FME.

4. EXPERIMENTAL RESULTS

In an experiment, we have used nine video sequences. Each sequence has 100 frames and the simulation conditions are shown in Table 2 [8].

Table 2. Simulation conditions

Reference Software	JM 8.2
Number of Frames	100
Profile	Baseline Profile (IPPP, RDO, CAVLC)
Search Range	±32
Reference Frames	5
Quantization Parameters	28, 32, 36, 40
Fast Motion Estimation Algorithm	UMHexagonS, CBFPS

For performance comparison, we use the Bjonteggard delta bit rates and Bjonteggard delta PSNR [9]. Time saving is defined as

$$\text{Time Saving} = \frac{\text{Time}(\text{reference}) - \text{Time}(\text{proposed})}{\text{Time}(\text{reference})} \times 100\ (\%) \quad (6)$$

Table 3 shows the efficiency of the early SKIP mode decision technique. The comparison is performed between the mode decision in the reference software and the early SKIP mode decision without FME. With Eq. (5), the best mode is likely to be SKIP for the smooth or background regions.

Table 3. Early SKIP mode decision

Sequences	△ Bit rates (%)	△ PSNR (dB)	Time Saving(%)
NEWS	- 0.44	+ 0.03	25.0
CONTAINER	- 2.02	+ 0.11	26.4
SILENT	- 0.92	+ 0.05	23.1
FOREMAN	- 0.16	0	15.0
CARPHONE	+ 0.09	0	15.9
COASTGUARD	- 0.29	+ 0.01	14.2
MOBILE	- 0.37	+ 0.02	9.0
STEFAN	+ 0.27	- 0.01	12.4
MOTHER & DAUGHTER	- 0.74	+ 0.03	20.6
Average	- 0.51	+ 0.03	17.9

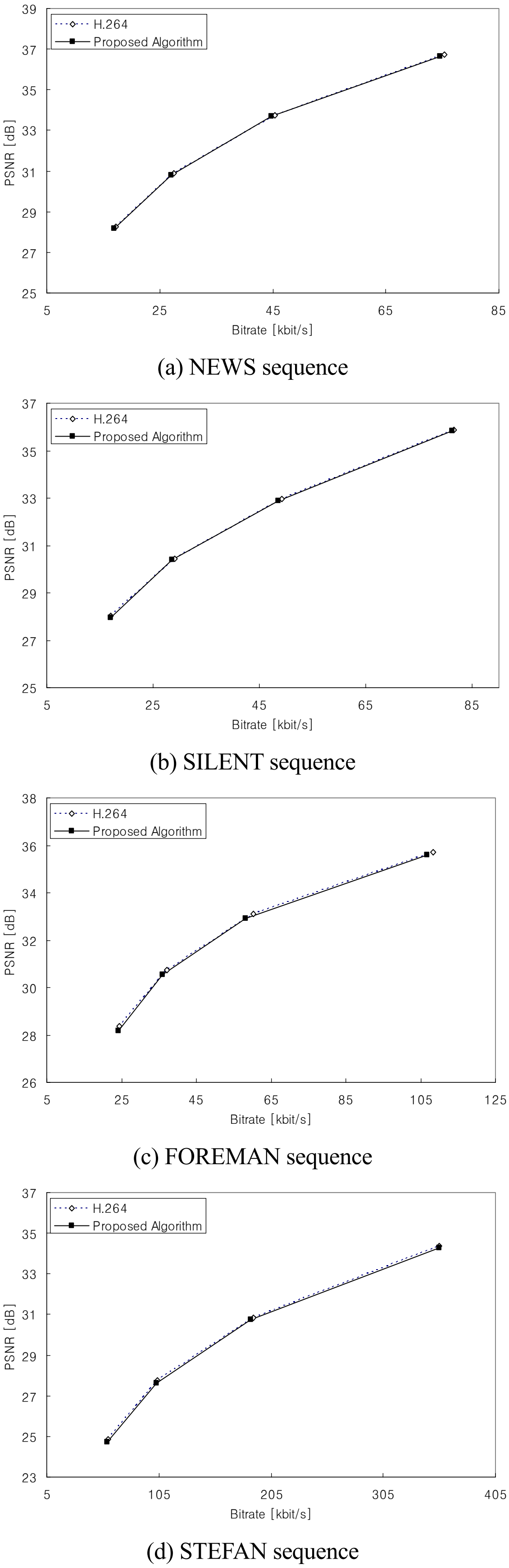
The early SKIP mode decision scheme increases PSNR by 0.03 dB and decreases bit rates by 0.51% on average. Thus, we can reduce the total coding bits efficiently without any significant PSNR loss. This scheme provides better performance than the mode decision of the reference software in terms of PSNR values and resulting bit rates.

Table 4 shows the performance of the proposed algorithm. We lose 0.02 dB PSNR with 0.45% bit rate increase; however, we achieve 88.4% time saving on average. Compared to a mode classification method [5], the proposed algorithm is faster with a slight bit rate increase and a small PSNR drop.

Table 4. Performance of the proposed algorithm

Sequences	△ Bit rate (%)	△ PSNR (dB)	Time Saving(%)
NEWS	- 0.02	0	93.5
CONTAINER	- 1.62	+ 0.09	95.3
SILENT	+ 0.03	0	92.1
FOREMAN	+ 0.95	- 0.05	87.3
CARPHONE	+ 1.34	- 0.07	88.8
COASTGUARD	+ 1.09	- 0.03	85.5
MOBILE	+ 0.71	- 0.03	79.3
STEFAN	+ 1.34	- 0.07	82.3
MOTHER & DAUGHTER	+ 0.21	0	91.1
Average	+ 0.45	- 0.02	88.3

For NEWS, SILENT, FOREMAN, and STEFAN sequences, the rate-distortion curves are shown in Fig. 4. We notice that the curves are almost identical to those of the H.264 reference software.



(a) NEWS sequence

(b) SILENT sequence

(c) FOREMAN sequence

(d) STEFAN sequence

Fig. 4. Rate-distortion curves

5. CONCLUSIONS

In this paper, we have proposed a fast mode decision algorithm using the early SKIP mode decision, and motion cost. The early SKIP mode decision method has reduced the encoding time without any significant PSNR loss and bitrate increase. The proposed fast mode decision algorithm was integrated with the fast motion estimation algorithm to achieve further speedup. Experimental results show that the proposed fast mode decision method reduces the encoding time by 88% on average with a negligible increase in bit rates and a slight PSNR drop. In this paper, we have only considered the fast inter mode decision; however, we can also develop a fast intra mode decision technique merged with the proposed fast inter mode decision algorithm.

ACKNOWLEDGEMENTS

This work was supported in part by MIC through RBRC at GIST, in part by MOE through BK21, and in part by DCRC.

REFERENCES

[1] ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC, “Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification,” *JVT Doc. JVT-G050*, 2003.

[2] JVT reference software version 8.2, available online at: http://iphome.hhi.de/suehring/tml/download/old_jm/

[3] K.P. Lim, S. Wu, D.J. Wu, S. Rahardja, X. Lin, F. Pan, and Z.G. Li, "Fast Inter Mode Selection," *JVT Doc. JVT-I020*, Sept. 2003.

[4] P. Yin, H.Y. Cheong Tourapis, A. Michael Tourapis, and J. Boyce, "Fast Mode Decision and Motion Estimation for JVT/H.264," *Proc. ICIP*, pp. 853-856, Sept. 2003.

[5] H.S. Kim, S.H. Kim, and Y.S. Ho, “Fast Mode Decision Algorithm Using Mode Classification for H.264,” *ITC,* Thiland, pp.51-58, Nov. 2004.

[6] Z. Chen, P. Zhou, and Y. He, "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," *JVT Doc. JVT-F017*, Dec. 2002.

[7] Z. Chen, P. Zhou, and Y. He, "Fast Motion Estimation for JVT," *JVT Doc. JVT-G016*, March 2003.

[8] G. Sullivan, and G. Bjontegaard, “Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressive-scan Source Material,” *ITU-T Q.6/16, Doc. VCEG-N81*, Sept. 2001.

[9] G. Bjontegaard, “Calculation of Average PSNR Difference between RD-curve,” *ITU-T Q.6/16, Doc. VCEG-M33*, April 2001.