Fast Mode Decision for Inter-prediction in H.264/AVC

Bin Zhan, Baochun Hou and Reza Sotudeh
School of Electronic, Communication and Electrical Engineering
University of Hertfordshire, Hatfield AL10 9AB United Kingdom
Tel: +44-(0)1707286279, Fax: +44-(0)1707284199
E-mail: {b.zhan, b.hou, r.sotudeh}@herts.ac.uk

***Abstract*— The computational complexity is one of the key challenges for variable block size based mode decision introduced by H.264/AVC. The significant time cost and power consumption lead difficulty for real-time and low power supplied applications such as video conference and mobile video communication. This paper presents a novel fast inter mode decision algorithm which removes low-frequency candidate inter modes by considering the statistical characteristics obtained from the early encoding stage. The adaptive adjustment function of the proposed algorithm can effectively avoid the obvious degradation of encoding performance. Experimental results show that the proposed algorithm reduces 20% encoding time on average with very limited decrease of encoding efficiency in terms of PSNR and bitrate.**

I. INTRODUCTION

The advanced video coding (AVC) standard H.264 is the latest video compression standard jointly developed by ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEG) [1]. Evolved from the former video compression standards, H.264 also introduces several advanced techniques to enhance the encoding efficiency [2]. Variable block size based motion estimation (ME) with rate distortion optimization (RDO) enabled mode decision (MD) is one important improvement. In contrast with the traditionally unique 16×16 macroblock, multiple block size (from 16×16 to 4×4 seven block size) more flexibly and accurately predicts the contents of the video frames especially some detailed information such like human face. However, the high encoding efficiency is achieved at the expense of sharply increased system cost. The significantly increased encoding time and power consumption are major challenges for the applications requiring real-time processing and low power. Several efforts [3, 4, 5, 6] have been made to simplify the MD complexity by either classifying or conjecturing the modes from the selected best modes of neighbouring blocks. This paper presents a fast inter MD algorithm, which is able to efficiently disable some candidate modes according to statistical characteristics obtained from the early encoding stage. The algorithm can also adaptively adjust the combination of candidate modes to minimize the performance degradation incurred by the change of objects and backgrounds in video scenes.

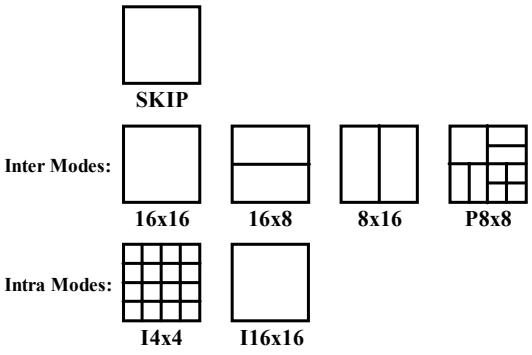


Fig. 1. Variable modes in H.264/AVC.

The rest of this paper is organized as follows: Section II introduces the proposed algorithm. Experimental results are shown in Section III. Section IV is the conclusion of this paper.

II. PROPOSED FAST INTER MODE DECISION ALGORITHM

H.264 supports seven modes for each macroblock: SKIP, Inter 16×16, Inter 16×8, Inter 8×16, Inter P8×8, INTRA 4×4 and INTRA 16×16; in the case of Inter P8×8, each Inter P8×8 mode can be further divided into four sub-macroblock modes: Inter 8×8, Inter 8×4, Inter 4×8, Inter 4×4. All modes have been shown in Fig. 1. MD operation consumes about 50% compression time when RDO is enabled [7]. RDO enabled MD is done by minimizing the Lagrangian cost as described in (1):

$$J(s,c,MODE/QP,\lambda_{MODE}) = SSD(s,c,MODE/QP) + \lambda_{MODE}\cdot R(s,c,MODE/QP) \quad (1)$$

where *MODE* is one candidate mode, *QP* is the quantization parameter, λ_{MODE} is the Lagrangian multiplier for MD as defined in (2), *SSD* is the sum of the squared differences between *s* (original block) and *c* (reconstructed block), *R(s, c, MODE/QP)* gives the number of bits based on the *MODE* and *QP*.

$$\lambda_{MODE} = 0.85\times 2^{(QP-12)/3} \quad (2)$$

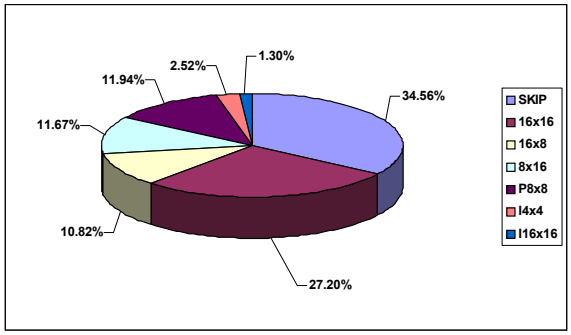


Fig. 2. Macroblock mode distribution of Foreman.

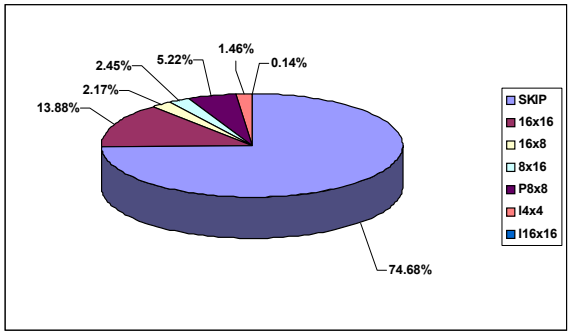


Fig. 3. Macroblock mode distribution of News.

The encoder selects one mode as the best mode which minimizes the value of $J(s, c, MODE/QP, \lambda_{MODE})$, and that means all inter prediction and intra prediction must been done before system could make the choice. The above processes, particularly the inter prediction, are really time-consumed. If some inter modes can be judged as redundant and removed from the candidates list earlier, the computational burden could be reduced, and that is the main motivation of proposed algorithm. Considering the most complexity comes from the inter prediction, the algorithm is focused on the simplification of inter modes selection.

Experiments display that the distribution of candidate modes is not equipotent. This indicates that there are potential redundant modes in the encoding process. It would not obviously affect the system performance if some low-frequency modes were disabled. Fig. 1 and Fig. 2 show the distribution of each selected macroblock mode for Foreman and News two video sequences (300 frames in QCIF for each) using standard H.264 JM model with Full Search (FS) ME scheme. From the figures, the frequency of selected modes is different, and this statistical characteristic is more obviously found for other video sequences with low-speed motion or large static scene background. Based on the above analysis, a fast inter MD algorithm is proposed. The algorithm can be outlined into two parts. Part one is the statistical learning encoding process. For the first few frames, all inter modes are enabled and checked as normal; the selection frequency of each mode is recorded. At the end of this part, the low-frequency inter mode(s) will be disabled according to the statistical score. Taking into account the real life videos, the change of objects and backgrounds in a video sequence can not be ignored. The modes selected in the first part could lead to poor encoding performance if a sudden change of video scenes occurs. In order to minimize this downgrade, an adaptive scheme for modes selection is necessary and is introduced as the second part of algorithm. Normally, the change existing in the video sequence results in the undulation of PSNR and bitrate during the encoding process, we therefore choose them as two criteria to inspect the alteration of video contents. However, the bitrate will keep at the similar level if rate control (RC) is enabled, and the adjustment based on the bitrate is not suitable; only PSNR adjustment scheme is active for this case.

A. The Steps of the proposed Algorithm

The detailed algorithm is summarized as follows:

Stage One: Statistical Learning Encoding Stage

Step1: Encode the first N P-frames by exploiting all modes specified in H.264 standard. For each encoded frame, calculate and update the ratio (r_x) of inter mode x as statistical data.

A: Inter MB	B: Inter sub-MB
r_2 - 16×8	r_4 - 8×8
r_3 - 8×16	r_5 - 8×4
r_8 - P8×8	r_6 - 4×8
	r_7 - 4×4

Step2: If RC is enabled, skip to Step3.
Otherwise, setup a moving window to record the bitrate of the most 30 encoded frames for adaptive adjustment based on the bitrate. For each frame, the number of bits of new encoded frame will be put into the moving window. The old value will be thrown out if the window is fully filled.

Step3: Setup another moving window to record the most 5 encoded frames' PSNR values for adaptive adjustment based on the video quality. For each frame, the PSNR value of new encoded frame will be put into the moving window. The old output will be thrown out if the window is fully filled.

Stage Two: Reduced Modes Encoding Stage

Step4: If (r_8 < threshold λ_1)
{Disable sub-modes: 8×8, 8×4, 4×8, 4×4}
If (r_x < threshold λ_1)
{Disable mode x}
Then a new list of modes will be used for the rest video sequence.

Stage Three: Adaptive Adjustment Encoding Stage

Step5: Encode next frame and monitor the undulation of bitrate (only when RC is disabled) and

PSNR.

Step6: If RC is enabled, skip to Step7.
Otherwise, if the undulation of bitrate exceeds the threshold λ_2 for 5 consecutive frames, turn on all disabled modes, and go back to Step1.

Step7: If the undulation of PSNR exceeds the threshold λ_3 for 5 consecutive frames, turn on all disabled modes, and go back to Step1.

Step8: If current frame is not the last frame of the sequence, go back to Step5. Otherwise, the process is finished.

B. Parameters

The parameters (elimination threshold λ_1, adjustment thresholds λ_2 and λ_3, and learning period N) mentioned in the last paragraph are defined in this section.

1. Inter modes elimination threshold λ_1:

The threshold λ_1 is used to determine which mode(s) will be turned off in order to reduce the system encoding time. If the value of λ_1 is set too low, just a few or even none of inter block size will be disabled and there would be only slight encoding time improvement. If a large value of λ_1 is used, too many inter modes could be filtered out. Although the encoding process will be speeded up significantly, the video quality could be very poor, and the bitrate would be increased. From our intensive experimental results, the better λ_1 value is chosen as 5%. Table I and Table II list the frequency of all modes for 10 different video sequences proved that 5% yields better performance.

2. Learning period N:

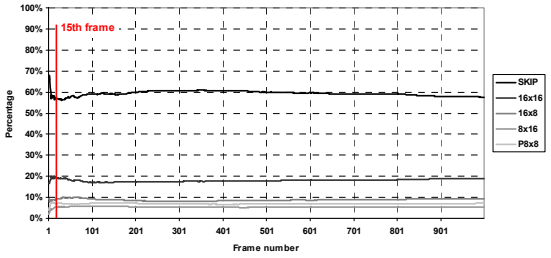
Period N is the indication of termination of algorithm's statistical learning stage. After N P-frames have been encoded, the encoder can turn off the low-frequency inter modes in order to reduce the processing time according to the λ_1 defined in the last paragraph. The statistical information collected from the first N frames should be sufficient to make the decision of mode elimination. It is desirable to use smaller number of frames for system's statistical learning stage to

TABLE I
PROPORTION OF MACROBLOCK MODE

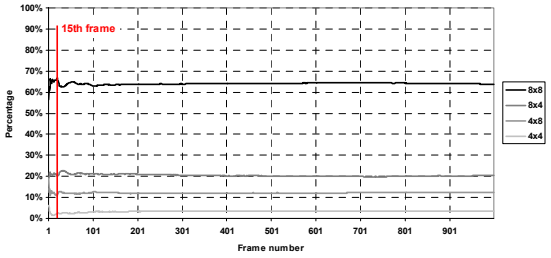
Sequence	SKIP	16×16	16×8	8×16	P8×8	I4×4	I16×16
foreman	34.56%	27.20%	10.82%	11.67%	11.94%	2.52%	1.30%
salesman	78.25%	13.10%	0.87%	1.32%	5.92%	0.32%	0.22%
carphone	37.46%	26.58%	7.29%	10.37%	15.63%	1.28%	1.39%
news	74.68%	13.88%	2.17%	2.45%	5.22%	1.46%	0.14%
akiyo	88.49%	6.70%	1.08%	1.31%	2.42%	0.00%	0.00%
silent	72.35%	12.83%	3.11%	3.98%	5.52%	1.74%	0.47%
bridge-far	97.94%	1.85%	0.04%	0.02%	0.12%	0.02%	0.01%
coastguard	11.93%	51.23%	9.91%	9.80%	13.86%	2.79%	0.48%
highway	57.83%	17.48%	9.29%	5.89%	7.77%	0.78%	0.96%
mobile	1.88%	54.62%	6.35%	4.59%	31.92%	0.54%	0.10%

TABLE II
PROPORTION OF SUB-MACROBLOCK MODE

Sequence	8×8	8×4	4×8	4×4
foreman	59.71%	17.82%	18.86%	3.61%
salesman	68.30%	12.55%	14.21%	4.94%
carphone	61.54%	16.16%	18.38%	3.92%
news	58.27%	12.74%	24.80%	4.19%
akiyo	68.52%	12.39%	16.70%	2.39%
silent	69.02%	11.09%	17.69%	2.20%
bridge-far	87.29%	6.39%	6.06%	0.26%
coastguard	60.78%	15.41%	20.32%	3.49%
highway	62.83%	17.95%	15.72%	3.50%
mobile	61.87%	16.22%	16.23%	5.68%

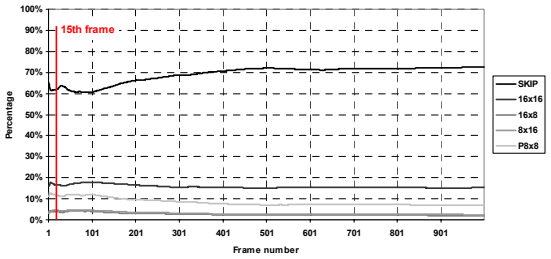


(a) Macroblock mode

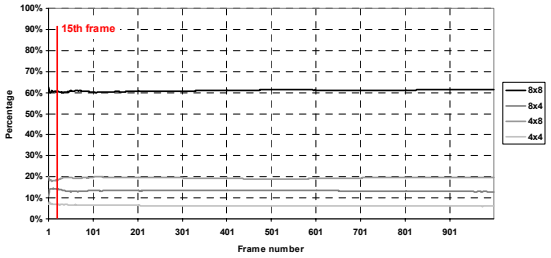


(b) Sub-macroblock mode

Fig. 4. Proportion of selected modes vs. frame number for Highway (1000 QCIF).



(a) Macroblock mode



(b) Sub-macroblock mode

Fig. 5. Proportion of selected modes vs. frame number for Paris (1000 CIF).

enable the mode elimination. Experiments show that there is a particular pattern of modes selection for encoding a video sequence, and this pattern can be confirmed close to its steady state after encoding certain frames. On one side, shorter statistical learning period could result in analysis's excursion from the actual data, and the rest encoding operations could be unreliable. On the other side, longer period's accumulation needs more frames and will result in unnecessary MD operations but less encoding time improvement. Our large amount simulations show that the system reaches relatively steady state around the 15th frame. Fig. 4 and Fig. 5 are two examples illustrating the distribution of each selected mode for Highway (QCIF) and Paris (CIF) sequences with total number of frames equals to 1000.

3. Bitrate adaptive adjustment threshold λ_2:

The adjacent frames are highly correlated in temporal domain, and they possess very similar details such as objects and backgrounds. However, some special cases, high-speed motion of objects and sudden moving of the camera, can not be ignored. Those abnormal factors could affect the encoding performance such as bitrate. Therefore, an adjustment scheme based on the undulation of output bitrate is proposed to adjust MD adaptively.

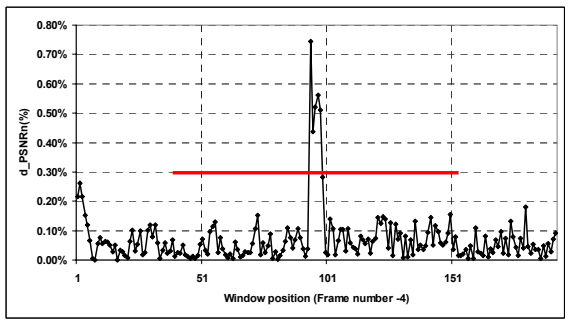
After I-frame has been encoded, the number of bits of each encoded P-frame is recorded as b_i for the i^{th} P-frame. The bitrate of every 30 frames (the frame rate of video sequence) denotes as X_Bit_n shown in (3). For example, X_Bit_1 is the averaged output bits of 30 P-frames between [1st, 30th], and X_Bit_2 represents the averaged output bits of 30 P-frames between [2nd, 31st]. This forms a moving window to monitor the undulation of bitrate. The absolute difference between X_Bit_n and X_Bit_{n-1} is named as d_Bit_n and is defined in (4). If the consecutive five (five frames can avoid inaccurate adjustment according to the casual changes of the video scenes) d_Bit_n values exceed threshold λ_2, there could exist obvious change in the video stream, and current MD scheme needs to be re-evaluated.

$$X\_Bit_n = \frac{b_n + b_{n-1} + \cdots + b_{n-29}}{30} \tag{3}$$

$$d\_Bit_n = ABS\left(\frac{X\_Bit_n - X\_Bit_{n-1}}{X\_Bit_{n-1}}\right) \times 100\% \tag{4}$$

According to the experimental results, the threshold λ_2 is set to 3%, and this value can be used to effectively turn on the disabled modes.

(a) Original PSNR values.



(b) d_PSNR_n

Fig. 6. An example for threshold λ_3.

4. PSNR adaptive adjustment threshold λ_3:

Compared to the bitrate, the magnitude of PSNR is very small. The wave of long window's averaged PSNR values could be close to flat, and the peaks of wave are not obvious enough for inspecting. Based on the simulation results, the size of PSNR monitoring window is defined as 5. The PSNR based evaluation process is very similar to the bitrate based adaptive adjustment; (5) and (6) give the calculation definitions of X_PSNR_n and d_PSNR_n. According to the experimental results, the threshold λ_3 is set to 0.3%. Fig.6 gives an example that the change of video contents occurred at the 100th frame.

$$X\_PSNR_n = \frac{SNR\_Y_n + SNR\_Y_{n-1} + \cdots + SNR\_Y_{n-4}}{5} \tag{5}$$

$$d\_PSNR_n = ABS\left(\frac{X\_PSNR_n - X\_PSNR_{n-1}}{X\_PSNR_{n-1}}\right) \times 100\% \tag{6}$$

III. Experiments and Discussion

Twelve video sequences were selected for experiments: {foreman in QCIF, bridge-far in QCIF, coastguard in QCIF, news in QCIF, akiyo in QCIF, silent in QCIF, m_d in QCIF, container in QCIF, mobile in QCIF, akiyo in CIF, hall in CIF and container in CIF}, and each sequence contains 300 frames. Our fast MD scheme was integrated into reference software JM10.1 [8]. The baseline profile was selected. The search range was set to ±16. Each sequence was encoded following the IPPP pattern. The FS scheme was deployed in ME, and one frame was used as reference. The experiments were carried out on PC with an AMD-processor 1.81 GHz and 256MB memory. The evaluation of performance was focused on the change rate of encoding time, PSNR and bitrate, i.e., $\triangle Time(\%)$, $\triangle PSNR(dB)$ and $\triangle Bitrate(\%)$. They are defined to describe the difference between exhaustive encoding processing and proposed encoding processing, and their definitions are shown in (7), (8) and (9).

TABLE III
THRESHOLD Λ_1 = 5% @ 15TH FRAME

Sequence	ΔTime(%)	ΔPSNR	ΔBitrate(%)
QP = 28			
foreman	-8.92%	-0.02dB	0.11%
bridge-far	-43.33%	-0.02dB	0.54%
coastguard	-7.38%	0.01dB	0.03%
news	-22.67%	-0.02dB	1.58%
akiyo	-29.45%	-0.09dB	1.26%
silent	-31.99%	-0.02dB	2.29%
m_d	-19.57%	-0.06dB	0.55%
contianer	-29.49%	-0.03dB	1.84%
mobile	-7.27%	0.01dB	0.22%
akiyo (CIF)	-14.19%	0.01dB	0.69%
hall (CIF)	-21.26%	-0.03dB	1.63%
container (CIF)	-13.59%	0.00dB	0.10%
QP = 32			
foreman	-5.64%	0.01dB	-0.28%
bridge-far	-38.20%	0dB	0.00%
coastguard	-8.97%	-0.01dB	0.51%
news	-22.42%	-0.03dB	2.05%
akiyo	-24.05%	-0.05dB	1.73%
silent	-24.87%	-0.01dB	1.29%
m_d	-30.92%	-0.16dB	2.08%
container	-22.19%	-0.02dB	0.93%
mobile	-15.47%	-0.01dB	0.21%
akiyo (CIF)	-14.31%	-0.04dB	0.62%
hall (CIF)	-12.98%	0.01dB	0.82%
container (CIF)	-11.85%	0.01dB	0.57%

$$\Delta Time = \frac{Time_{proposed} - Time_{JM10.1}}{Time_{JM10.1}} \times 100\% \quad (7)$$

$$\Delta PSNR = PSNR_{proposed} - PSNR_{JM10.1} \quad (8)$$

$$\Delta Bitrate = \frac{Bitrate_{proposed} - Bitrate_{JM10.1}}{Bitrate_{JM10.1}} \times 100\% \quad (9)$$

Table III lists the experimental results. It is clearly shown that the proposed fast MD algorithm can reduce encoding time up to 43.33%, and the decrease of PSNR and the increase of bitrate are very limited. Fig. 7 and Fig. 8 present the variation of PSNR values with the increasing of bitrate for JM10.1 model and proposed algorithm using Foreman (QCIF) and Paris (CIF) sequences respectively. The results are very comparable. Fig. 9 and Fig. 10 give an example that the proposed algorithm adaptively adjusts the mode selection when sudden change occurs in the video sequence.

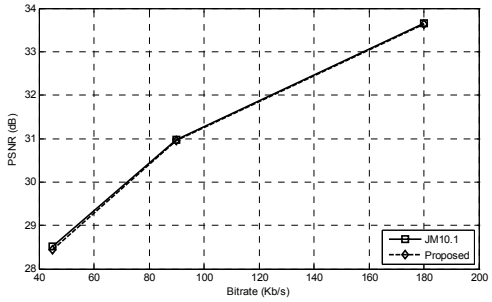


Fig. 7. Bitrate vs. PSNR for Foreman (QCIF).

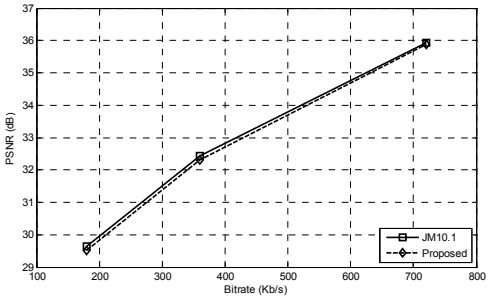


Fig. 8. Bitrate vs. PSNR for Paris (CIF).

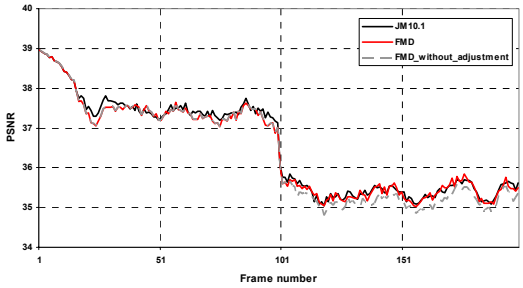


Fig. 9. PSNR vs. frame number.

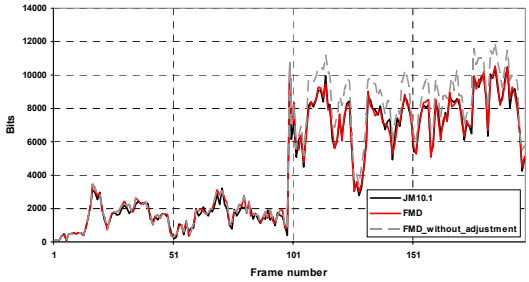


Fig. 10. Bits vs. frame number.

IV. CONCLUSIONS

A novel fast inter MD algorithm has been introduced in this paper. Statistical data based scheme can significantly improve the encoding efficiency. Adaptive strategies based on the video quality and the bitrate can minimize the degradation of encoding performance. Furthermore, the proposed algorithm does not involve any complex computation to support it. It is very attractive to real-time and small mobile communication applications. The parameters adaptation of algorithm is under further investigation.

REFERENCES

[1] "Draft ITU-T Rec. and FDIS of Joint Video Spec. (H.264 | ISO/IEC 14496-10 AVC)," JVT of MPEG and VCEG, Doc. JVT-G050rl, May 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits and Systems for Video Technology*, vol. 13, pp.560-576, July 2003.

[3] B. Jeon and J. Lee, "Fast mode decision for H.264," ITU-T Q.6/16, Doc. #JVT-J033, 2003.

[4] A. C. Yu, "Efficient block-size selection algorithm for interframe coding in H.264/MPEG-4 AVC," *Proc. IEEE ICASSP*, vol.3, pp. 169-172, May 2004.

[5] H. Zhu, C. K. Wu, W. Y. Li and Y. Fang, "Fast mode decision for H.264/AVC based on macroblock correlation," *Proc. AINA'05. 19th International Conference*, vol.1, pp.775-780, March 2005.

[6] K. H. Han and Y. L. Lee, "Fast macroblock mode determination to reduce H.264 complexity," *IEICE Trans. Fundamentals*, vol.E88-A, pp. 800-804, March 2005.

[7] Y. F. Ling, Z. H. He, and I. Ahmad, "Analysis and design of power constrained video encoder," *Proc. IEEE 6th CAS Symp. on Emerging Technologies: Frontiers of Mobile and Wireless Communication*, pp. 57-60, May 2004.

[8] JVT reference software unofficial version JM10.1, http://iphome.hhi.de/suehring/tml/download/jm10.1.zip.