International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:3, No:11, 2009

Transform-Domain Rate-Distortion Optimization Accelerator for H.264/AVC Video Encoding

Mohammed Golam Sarwer, Lai Man Po, Kai Guo and Q.M. Jonathan Wu

***Abstract*—**In H.264/AVC video encoding, rate-distortion optimization for mode selection plays a significant role to achieve outstanding performance in compression efficiency and video quality. However, this mode selection process also makes the encoding process extremely complex, especially in the computation of the rate-distortion cost function, which includes the computations of the sum of squared difference (SSD) between the original and reconstructed image blocks and context-based entropy coding of the block. In this paper, a transform-domain rate-distortion optimization accelerator based on fast SSD (FSSD) and VLC-based rate estimation algorithm is proposed. This algorithm could significantly simplify the hardware architecture for the rate-distortion cost computation with only ignorable performance degradation. An efficient hardware structure for implementing the proposed transform-domain rate-distortion optimization accelerator is also proposed. Simulation results demonstrated that the proposed algorithm reduces about 47% of total encoding time with negligible degradation of coding performance. The proposed method can be easily applied to many mobile video application areas such as a digital camera and a DMB (Digital Multimedia Broadcasting) phone.

***Keywords*—**Context-adaptive variable length coding (CAVLC), H.264/AVC, rate-distortion optimization (RDO), sum of squared difference (SSD).

I. Introduction

H.264/AVC is the newest hybrid video coding standard developed by ITU-T Video Coding Expert Group (VCEG) and ISO/IEC MPEG Video Group named Joint Video Group (JVT) [1, 2]. H.264/AVC has proved its superiority in coding efficiency over its precedents, e.g., it shows a more than 40% rate reduction over H.263 [3]. The rate-distortion optimization (RDO) is one of the essential parts of the H.264/AVC encoder to achieve the much better coding performance. However, the computational complexity of the RDO technique is extremely high. Hence, the cost function computation makes H.264/AVC impossible to be realized in real time applications without high computing hardware.

In order to accelerate H.264/AVC video coding, a number of researches have been made to explore fast algorithms in motion estimation [4-5] and mode decision [6-7]. To reduce complexity, a new cost function for intra 4x4 mode decisions was proposed in [9]. In this cost function, sum of absolute integer transform difference (SAITD) is used in distortion part and a rate prediction algorithm is used in rate part. A new transform domain rate estimation and distortion measure was proposed to reduce the rate-distortion cost function computation with ignorable coding performance degradation [10]. But the computation reduction is low. In order to avoid the entropy coding method during mode decision process, several methods [11, 12, 13] are introduced to predict the bit rate of 4x4 block. To reduce the complexity of distortion part of RDO technique, a new fast sum of squared difference (FSSD) computation algorithm with use of an iterative table-lookup quantization process was proposed in [14]. This FSSD algorithm was based on the theoretical equivalent of the SSDs in spatial and transform domain. This approach avoided the inverse quantization/transform and pixel reconstruction processes with nearby no rate-distortion performance degradation.

In this paper, a transform-domain rate-distortion optimization accelerator is developed based on fast sum of squared difference (FSSD) and rate estimation algorithm. A method to estimate the rate of context-adaptive variable length coding (CAVLC) is proposed. An efficient hardware structure for implementing the proposed transform-domain RDO accelerator is also introduced.

The remainder of this paper is organized as follows. Section 2 provides the review of rate distortion optimized mode decision technique. In section 3, transform domain fast sum of squared difference (FSSD) calculation is described. Section 4 introduces the fast rate estimation method. In section 5, algorithm of proposed rate distortion optimization accelerator is presented. Hardware architecture of proposed RDO accelerator is proposed in section 6. The simulation results of the proposed method are presented in section 7. Finally, section 8 concludes the paper.

Mohammed Golam Sarwer was with City University of Hong Kong, Hong Kong. He is now with the University of Windsor, Windsor, ON, Canada (e-mail: sarwer@uwindsor.ca).

Lai Man Po and Kai guo are with the Department of Electronic Engineering, City University of Hong Kong, Kowloon, Hong Kong, (e-mail: eelmpo@cityu.edu.hk).

Q. M. Jonathan Wu is with the Department of Electrical and Computer Engineering, University of Windsor, Windsor, ON, Canada(e-mail: jwu@uwindsor.ca).

II. Rate Distortion Optimized Mode Decision of H.264/AVC

To take the full advantages of all modes, the H.264/AVC encoder can determine the mode that meets the best RD tradeoff using RDO mode decision scheme. The best mode is the one with minimum rate-distortion (RD) cost and this cost is defined as

$$J_{RD} = \mathrm{SSD}(\mathbf{S},\mathbf{C}) + \lambda \cdot R \qquad (1)$$

where λ is the Lagrange multiplier, R is the number of coded bits for each macroblock which can be written as follows:

$$R = R_{header} + R_{motion} + R_{res} \qquad (2)$$

where R_{header}, R_{motion} and R_{res} means the number of bits need to encode the header information, motion vectors and quantized residual block, respectively. The SSD(**S**,**C**) in (1) is the sum of the squared difference (SSD) between the original blocks **S** and the reconstructed block **C**, and it can be expressed as

$$\mathrm{SSD}(\mathbf{S},\mathbf{C}) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(S_{ij} - C_{ij})^2 \qquad (3)$$

where S_{ij} and C_{ij} are the *(i, j)*th elements of the current original block **S** and the reconstructed block **C**, respectively. In order to compute RD cost for each mode, same operation of forward and inverse transform/quantization and entropy coding is repetitively performed. All of these processing explains the high complexity of RD cost computation.

III. Fast Sum of Absolute Difference Calculation

The cause of difference between original block and reconstructed block is due to the quantization error. Hence the spatial domain SSD and transform domain SSD is theoretically equivalent [14]. Mathematically,

$$SSD(S,C) = \sum_{i=0}^{N-1}\sum_{j=0}^{N-1}(q_{ij}[f_{ij}^{*} - \hat{f}_{ij}^{*}])^2 \qquad (4)$$

where, f_{ij}^{*} is the *(i ,j)*th element of integer cosine transformed block, $\hat{f}_{ij}^{*}$ is the *(i ,j)*th element of inverse quantized transform block and q_{ij} is the *(i ,j)*th element of scaled transform matrix . The relationship between f_{ij}^{*} and $\hat{f}_{ij}^{*}$ is defined as [14]

$$\hat{f}_{ij}^{*} = Q^{-1}(Q(f_{ij}^{*})) = \Delta_{ij}\cdot round(f_{ij}^{*}/\Delta_{ij}) = \Delta_{ij} * Z_{ij} \qquad (5)$$

where $\Delta_{ij} = \Delta / q_{ij}$ represents the new quantization step size define in [14] and Δ is the quantization step size of original H.264 encoder. In order to avoid the multiplication and division operation in (5), we proposed an iterative table-lookup quantization process in [14].

From (4), it is shown that to calculate the SSD we need 16 multipliers and 16 square operators. Actually the q_{ij} is the constant value. The value of q_{ij} at different pixel position is given in Table 1. In the first region of Table 1 multiplication by q_{ij} can easily by implemented by shifting right by two bits. To avoid the multiplication operation in second region q_{ij} is approximated as $\frac{1}{10} \approx \frac{1}{2^3}$. Hence multiplication in this region is replaced by shifting right by three bits. Similarly, for region III, q_{ij} is approximated as $\frac{1}{4}\sqrt{\frac{2}{5}} \approx (\frac{1}{2^3} + \frac{1}{2^5})$. From simulations, it is shown that these approximations do not significantly affect the SSD calculation as well as rate distortion performance. It should be noted that no computation is necessary for shifting operator. In order to avoid the square operation, we have used a square table. It should be noted again that f_{ij}^{*} is the coefficients before quantization and $\hat{f}_{ij}^{*}$ is the coefficients after inverse quantization. So the sign of f_{ij}^{*} and $\hat{f}_{ij}^{*}$ is always same. Additionally, the differences between f_{ij}^{*} and $\hat{f}_{ij}^{*}$ must be smaller than quantization step size Δ_{ij}. Mathematically,

$$f_{ij}^{*} - \hat{f}_{ij}^{*} < \Delta_{ij}$$

$$f_{ij}^{*} - \hat{f}_{ij}^{*} < \frac{\Delta}{q_{ij}}$$

$$q_{ij}(f_{ij}^{*} - \hat{f}_{ij}^{*}) < \Delta \qquad (6)$$

TABLE I
q_{ij} AT DIFFERENT POSITION

Region	Position (i, j)	q_{ij}
I	(0,0),(0,2),(2,0),(2,2)	$a^2 = \frac{1}{4}$
II	(1,1),(1,3),(3,1),(3,3)	$\frac{b^2}{4} = \frac{1}{10}$
III	Others	$\frac{ab}{2} = \frac{1}{4}\sqrt{\frac{2}{5}}$

For example, the value of Δ is 10, 16, 26, 40 and 64 for QP is 24, 28, 32, 36, and 40, respectively. If we choose QP=24, the left hand side of (6) is always less that 10. Additionally, the values of left hand sides are always integer. If we store the squares of all integer values which are less than 10 (for QP=24), then 16 square operation can be saved. In this way, we need additional memory requirement. But the size of memory is not large. We have to store only 10 elements in case of QP=24. Thus, the computational intensive multiplication and square operations can be avoided.

IV. Fast Rate Estimation

Context adaptive variable length coding (CAVLC) uses five syntax elements to encode the 4x4 block. To estimate the bits for quantized transform coefficients, we estimate the number of bits for each of five different types of symbols of CAVLC separately.

A. *Coefficient Token*

Both the total number of nonzero coefficients (Ncoeff) and the number of trailing +/- 1s (T_L) are coded as combined event. For all elements, except chroma DC, a choice between three tables and one fixed length codeword is made [15]. N_T is a parameter used for table selection. Assume Nu and N_L is the number of non-zero coefficients of upper and left block, respectively. If both upper and left blocks are available then N_T = (Nu+ N_L)/2. If only upper block available N_T =Nu; if only left block available N_T =N_L; if neither is available N_T =0. Since to encode coefficient token, three different VLC tables are used, it needs more memory to store all the possible symbols. This is not efficient for hardware implementation in terms of memory requirement. It is well known that neighboring blocks are highly spatially correlated. Hence there is high correlation between parameter N_T and number of non-zero coefficients of current block (Ncoeff). Therefore, we can replace N_T by Ncoeff for selection of VLC tables. The algorithm for table selection of proposed method given below:

```
if ( 0<= $N_{coeff}$ <2): Num-VLC0
if ( 2<= $N_{coeff}$ <4): Num-VLC1
if ( 4<= $N_{coeff}$ <=7): Num-VLC2
if ($N_{coeff}$ >7): 6 bit fixed length codeword
```

TABLE II
Probability of wrong table selection

Sequence	QP			
	28	32	36	40
Akiyo	0.13	0.09	0.05	0.02
Claire	0.06	0.05	0.05	0.02
Foreman	0.21	0.19	0.15	0.13
Container	0.26	0.21	0.17	0.18
Stefan	0.24	0.17	0.18	0.19
Miss_America	0.09	0.06	0.03	0.03
Silent	0.24	0.18	0.15	0.16
Salesman	0.20	0.23	0.14	0.16

In order to justify the above algorithm, we have done several experiments and calculated the probability of wrong table selection of different video sequences with different QP values. Table 2 shows the probability of wrong table selection. From Table 2, we have seen that this probability is very low for low motion sequences such as Akiyo, Claire, and Miss America. Although this probability for medium and high motion sequences is large, it does not greatly influence the rate-distortion performance. This is because number of bits for a particular coefficient token is almost similar in all VLC tables. Let us assume a symbol with number of non-zero coefficient is 4 and number of trailing ones is 3. The number of bits using VLC tables 0 is 6, while the number of bits is 4 for both VLC tables 1 and 2 [15]. Similarly when the number of non-zero coefficients is 3 and number of trailing one is 0, number of bits to encode this symbol are 9, 7and 6 for VLC tables 0, 1 and 2, respectively. By combing the mentioned algorithm with VLC tables, we have developed a rate table to encode the coefficient token. Rate table is generated by collecting the length of codeword from the different tables. For example, when Ncoeff is either 0 or 1, data is collected from VLC table0 and while the value of Ncoeff is either 2 or 3, data is collected from VLC table 1. From observation of VLC tables, we have seen that the rate is exactly equal to 6 while number of non-zero coefficient of current block (Ncoeff) is greater than 7. Otherwise, rate of 4x4 block (Rcoeff) is estimated based on a new rate table defined in Table 3. Therefore, algorithm for proposed rate estimation for coefficient token of 4x4 block is given as follows:

```
if (Number of nonzero coefficients> 7) $R_{coeff}$=6;
else search rate from Table 3;
```

TABLE III
Rate Table for Coefficient token of 4x4 block

Trailing ones (T_L)	0	1	2	3
Total Coefficients (N_{Coeff})				
0	1	-	-	-
1	6	2	-	-
2	6	5	3	-
3	7	6	6	4
4	7	5	5	4
5	7	5	5	4
6	7	6	6	4
7	7	6	6	4

B. *Sign of Trailing ones*

One bit is used to signal sign information of trailing +/- 1s. 0 is used for positive and 1 is used for negative. So bit consumption to encode the sign of trailing ones (R_{trail1}) is exactly equal to the number of trailing ones (N_L).

C. *Encode the Level*

CAVLC makes the choice of the VLC look-up table for the level adaptive in a way where the choice depends on the recently coded levels [16]. One out of 7 level VLC tables is used to encode the level information [15]. Level VLC0 has its own structure while the other tables, level VLCN, N=1 to 6, share common structure. From the observation of VLC tables, we have seen that bit consumption to encode a coefficient is increasing with increasing the absolute value of that coefficient. Additionally, the bit consumption also depends on the absolute value of previously coded level coefficient. From the observation, rate for level is estimated as follows

$$R_{level(i)} = |L_i| \quad \text{for highest frequency component} \qquad (7.a)$$

$$R_{level(i)} = \frac{\alpha_1 |L_i| + \beta_1 |L_{i-1}|}{\alpha_1 + \beta_1} \quad \text{for other component} \qquad (7.b)$$

where L_{i-1} is the recently encoded level of i th level. In order to set the weighting factors, we have done several experiments for different video sequences (*Akiyo, Foreman, Stefan*, and *Mobile*) with QCIF format at different QP values. We have observed RD performance of these video sequences at different combinations of weighting factor. Better RD performance was found at $\alpha_1 = 3$ and $\beta_1 = 1$.

D. *Encode Total Zeros*

The codeword Total_zeros is the number of zeros between the last non-zero coefficient of the zig-zag scan and its start. In case of mode decision process, only number of bits to encode the symbol is necessary. If only the length of codeword (instead of codeword) is retrieved from the total_zero table, then we can avoid the extra calculation for codeword generation of CAVLC during mode decision process.

E. *Encode Runs*

Since after transformation and quantization, blocks typically contain mostly zeros, CAVLC algorithm uses run-level coding to represent strings of zeros compactly. Run means the number of preceding zeros before each non-zero coefficient and Zeros_left called the number of zeros left before each non-zero coefficient. Based on Run and Zeros_left, a run VLC tables is used to encode this element [15]. From observation of run VLC table, it is shown that no of bits to encode this coefficient is increasing with the value of run. The estimated rate for run is proposed as

$$R_{run(i)} = \alpha_2 Run_i + \beta_2 \qquad (8)$$

Here α_2 and β_2 are the weighting factors. From observation of run table, we have seen that value of α_2 and β_2 are depends on the Run. Table 4 shows the value of α_2 and β_2 with different value of run.

TABLE IV
VALUE OF α_2 AND β_2

R_{run}	α_2	β_2
0,1,and 2	1	1
3,4,5,and 6	0	3
>6	1	-3

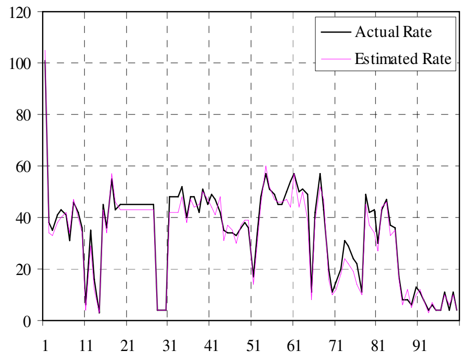


Fig. 1 Estimated and the actual rates of Foreman (X-axis: Block number, Y-axis: number of bits)

Therefore, the number of bits to encode the 4x4 block is calculated by summing the estimated bits of all symbols. Mathematically,

$$R_{res(est)} = R_{coeff} + R_{trail1} + \sum_{i=1}^{P} R_{level(i)} + R_{zero} + \sum_{i=1}^{Q} R_{run(i)} \qquad (9)$$

where P and Q are the number of level and run, respectively. Fig. 1 shows the comparison of our proposed method with actual rate for the Foreman sequence. Data is collected from the first 100 4x4 block. QP factor is set as 28. It is shown that the proposed method is very closely matched with actual rate.

V. RATE-DISTORTION OPTIMIZATION ACCELERATOR

The overall rate-distortion cost computation process using the proposed method combined with FSSD [14] can be summarized as:

Step 1: Compute the predicted block using inter or intra frame prediction: **P**
Step 2: Compute the residual (difference) block: **D** = **S** – **P**
Step 3: ICT transform the residual block: $\mathbf{F}^*$ = ICT (**D**)
Step 4: for *i*=0 to *N-1* and *j*=0 to *N-1* determine the quantized coefficients Z_{ij} by the following iterative table-lookup quantization process:

Step I: Set *k*=0 and if $f_{ij}^* < 0$ then set *sign*=-1, otherwise *sign*=1.

Step II: If $|f_{ij}^*| \geq k\Delta_{ij}$, then *k*=*k*+1 and goto Step II;

Step III: Set $Z_{ij} = sign \times k$ and $\hat{f}_{ij}^* = sign \times k.\Delta_{ij}$;

Step 5: Calculate SSD;
Step 6: Estimate the number of bits $R_e = R_{header} + R_{motion} + R_{res(est)}$
Step 7: Calculate the R-D cost : $J_{RD} = SSD + \lambda \cdot R_e$

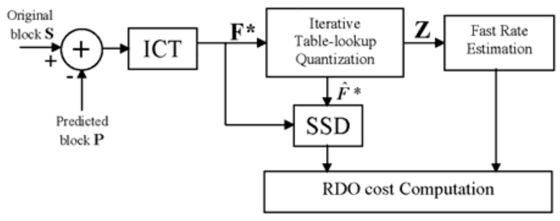


Fig. 2 Block diagram of proposed RDO

The block diagram of proposed RDO accelerator is shown in Fig. 2. The major difference is that the conventional quantization process is replaced by the iterative table-lookup quantization and the SSD calculation is preformed in the ICT transform domain. Additionally, entropy coding technique is replaced by fast rate estimation algorithm. We can obtain the rate-distortion cost without computations of two scale transforms, quantization, inverse quantization, inverse ICT transform and entropy coding.

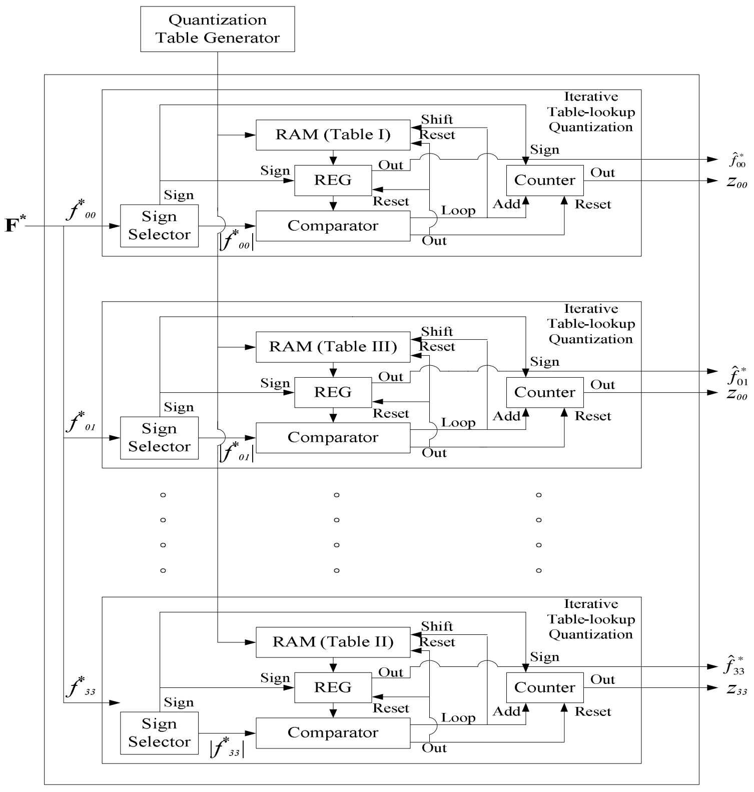


Fig. 3. Hardware architecture for iterative table-lookup quantization process

VI. Hardware Architecture of Proposed RDO Accelerator

Fig. 3 describes a possible hardware structure for the iterative table-lookup quantization implementation [14]. As each coefficient does not depend on the others, the quantization process can be performed in parallel. In the other words, we can quantize all the coefficients of f_{ij}^* simultaneously. At first, the quantization points and boundary points are generated by quantization table generator at the beginning of encoding process and then stored in the RAM, in which there is a pointer that points to the current boundary point. Then the current boundary point and quantization point are loaded in a register, which will compare with $|f_{ij}^*|$ after the sign of f_{ij}^* is abstracted by a sign selector. If the comparator judges that $|f_{ij}^*|$ is larger, then 'Loop' signal takes effect which tells counter to add one and the pointer to shift forward and point to the next boundary point. When $|f_{ij}^*|$ is smaller than the current boundary point, 'Out' signal takes effect which resets the pointer to the initial position, asks the counter to output the accumulated value z_{ij} and asks the register to output the current quantization value, $\hat{f}_{ij}^*$. From the structure in Fig. 3, we can find that the proposed table-lookup quantization process is very suitable for hardware implementation, since it does not require complicated operation modules and can execute in the parallel mode.

Fig. 4 shows the architecture of SSD calculator. It is shown that 16 multiplication operations are replaced by 24 shifting operators. It should be noted that no computation is necessary for shifting operator. In order to avoid the square operator, we have used a square table. In the square table, square of all integers within quantization step size Δ are stored. In this way, we need additional memory requirement. But the size of memory (ROM) is not large because of the limited value of Δ. For example encoder has to store 64 values if the quantization parameter QP is 40. Architecture of shifter for region III is given in Fig. 5. Here multiplier is replaced by two shift right operators.

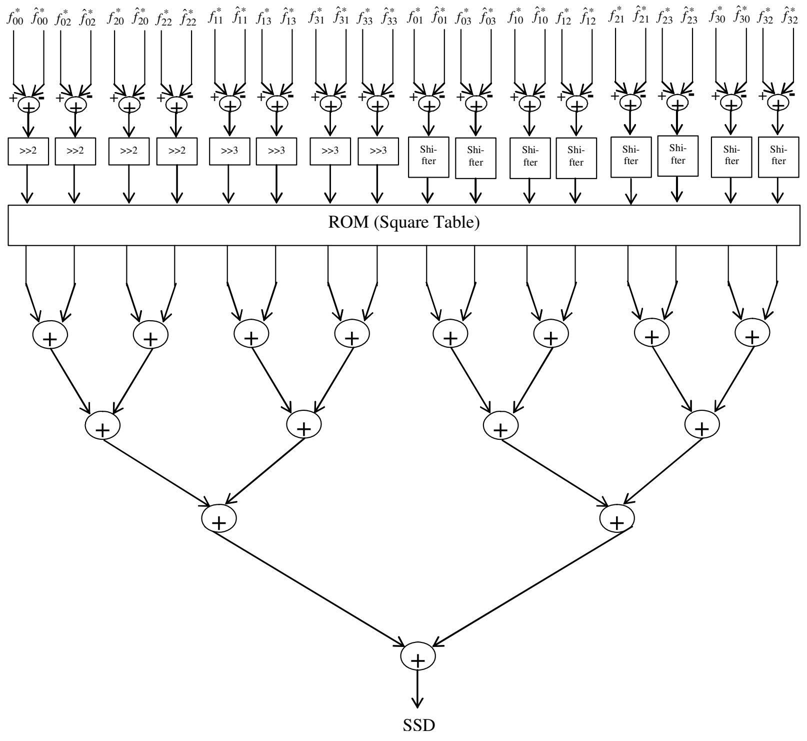


Fig. 4. Architecture of SSD calculator

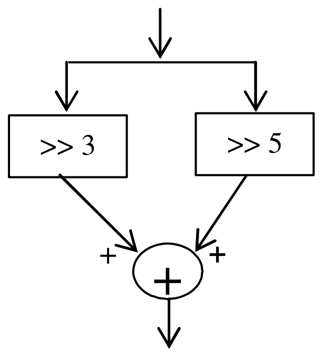


Fig. 5 Architecture of shifter for region III

The hardware architecture for proposed fast rate estimation algorithm is shown in Fig. 6. The proposed architecture contains a number of counters and register to store the information for the block. Non-zero coefficients counter is used to store the number of non-zero coefficients (N_{coeff}). Trailing one counter is used to store the number of trailing +/-1s. Total zero counter is used to store the total number of zeros before the last non-zero coefficient. Run before register is used to store the number of zeros preceding each non-zero coefficient. The level detector checks if the coefficient is either level or not. The proposed architecture begins by reading the coefficients from the input buffer in reverse zig-zag order. In each cycle, it reads one coefficient from the input buffer analyzes the coefficient and updates the information stored in the related counter and register. Reverse zig-zag scanning enables us to determine the necessary information for encoding 4x4 block. It takes 16 cycles to read the coefficients and stored the corresponding information in the counter and register file for each 4x4 block. Based on the number of non-zero coefficients, comparator 1 select either the rate table for coefficient token (ROM1) should be used or not. Comparator 1 compares the output of non-zero coefficient counter with content of register 4. Register 4 always store a fixed value 7. Comparator1 sent a signal “1” when the number of non-zero coefficients is greater than the fixed value 7. Then the register 1 is enabled. If comparator 1 sent the signal “0”, rate table for coefficient token is activated. In this case, rate is found by searching the rate table based on the number of non-zero coefficients and number of trailing ones. Similarly number of bits to encode the total zero is calculated by rate table for total zeros (ROM2). The level detector checks if the coefficient is zero or not. If the level is non-zero, it will be stored into the level-first-in-first-out buffer (FIFO), and the corresponding run information will be also stored to the run-FIFO. Based on the content of FIFO, level and run rate estimators predict the rate of level and run, respectively.

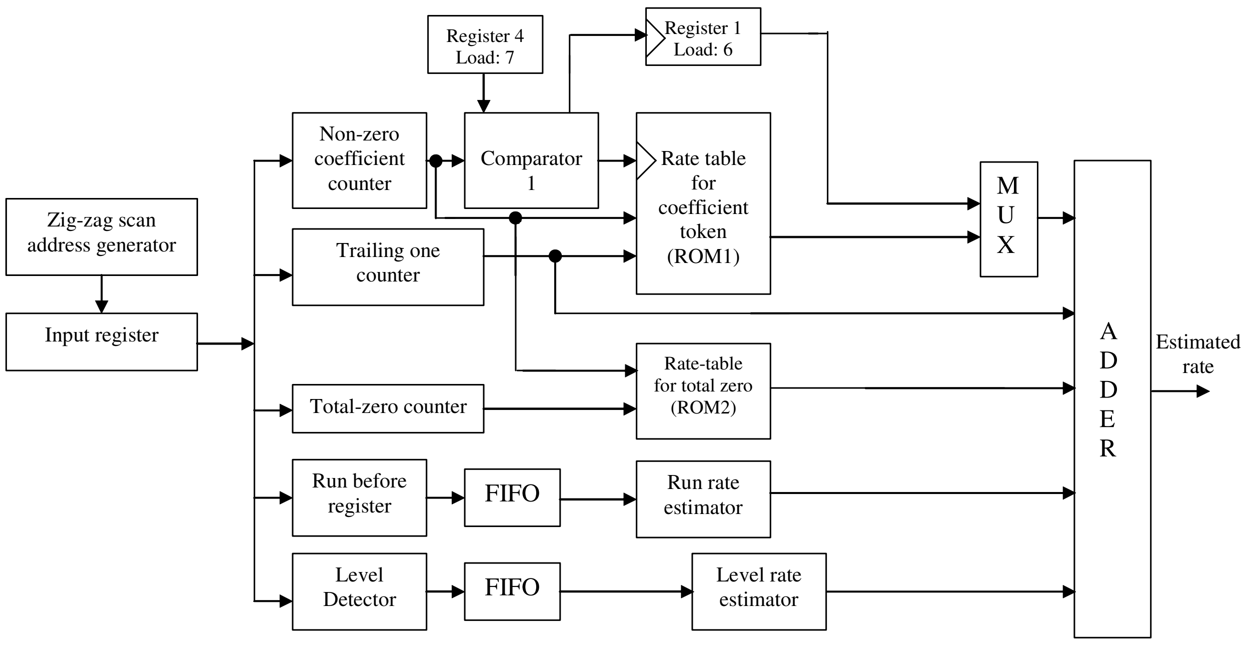


Fig. 6. Hardware structure of proposed rate estimator

VII. Simulation Results

To evaluate the performance of the proposed method, JM 9.6 [8] reference software is used in simulation. Different types of video sequences are used as test materials. A group of experiments were carried out on the test sequences with different quantization parameters. All the simulations are conducted under Windows Vista operating system, with Pentium 4 2.2 G CPU and 1 G RAM. The conditions of the simulation are listed in Table 5. The comparison results are produced and tabulated based on the average difference of total encoding time (ΔT %), the average PSNR differences ($\Delta PSNR$), and the average bit rate differences ($\Delta Rate$ %). PSNR and bit rate differences are calculated according to the numerical averages between RD curves derived from the original and proposed algorithm, respectively. The detail procedure to calculate these differences can be found in [17]. In order to evaluate the complexity reduction, ΔT (%) is defined as follows

$$\Delta T\% = \frac{T_{original} - T_{proposed}}{T_{original}} \times 100\% \qquad (10)$$

where, $T_{original}$ denotes the total encoding time of the JM 9.6 encoder and $T_{proposed}$ is total encoding time of the encoder with proposed method.

TABLE V
Simulation Conditions

GOP Structure	IPPP
Quantization Parameters	24/28/32/36
Number of Reference Frame	1
Hadamard Transform	OFF
Search range	16
Symbol mode	CAVLC/CABAC
RDO optimization	ON
Frame rate	30 fps
Number of frames	50
Fast Motion estimation	Enabled

A. *Results with CAVLC entropy coding method*

In this experiment, after selecting the best mode by proposed technique, CAVLC is used for final entropy coding of best mode. The rate-distortion performance comparison of the conventional RDO and proposed RDO in terms of PSNR, bit-rate, and encoding time are shown in Tables 6. When only rate estimation is used, the degradation of RD performance is very negligible. We have seen that while only rate estimation is used, the average PSNR degradation is 0.017 dB and the average bit rate increment is 0.615%. When both two algorithms are used, the rate-distortion performance differences are always less than 2.2 % and 0.07 dB in terms of bit rate and PSNR, respectively. Fig. 7 shows the comparison of rate-distortion curves of proposed accelerator with original RD optimization method. The performance of proposed

TABLE VI
EXPERIMENTAL OF PROPOSED METHOD WHILE CAVLC IS USED AS ENTROPY CODING

Sequence	Rate estimation only			FSSD + Rate Estimation		
	Δ PSNR	Δ Rate%	ΔT %	Δ PSNR	Δ Rate%	ΔT %
Foreman (CIF)	0.01	0.74	34.11	0.03	1.59	50.32
Flower (CIF)	0.01	0.37	35.66	0.04	1.07	49.32
Bus (CIF)	0.03	1.05	28.96	0.07	2.20	43.69
Akiyo (QCIF)	0.02	0.64	32.5	0.03	0.95	45.61
Mobile (QCIF)	0.01	0.35	36.86	0.05	1.36	50.05
Salesman (QCIF)	0.01	0.41	35.12	0.06	1.81	46.84
Stefan (QCIF)	0.03	0.77	33.13	0.07	1.83	43.7
Average	0.017	0.615	33.76	0.05	1.54	47.07

TABLE VII
EXPERIMENTAL OF PROPOSED METHOD WHILE CABAC IS USED AS ENTROPY CODING

Sequence	Rate estimation only			FSSD + Rate Estimation		
	Δ PSNR	Δ Rate%	ΔT %	Δ PSNR	Δ Rate%	ΔT %
Foreman (CIF)	0.06	2.64	36.84	0.07	3.04	49.43
Flower (CIF)	0.07	1.81	43.56	0.10	2.67	56.67
Bus (CIF)	0.06	1.89	37.14	0.08	2.71	50.25
Akiyo (QCIF)	0.04	1.23	33.64	0.06	1.73	44.65
Mobile (QCIF)	0.07	1.90	46.72	0.10	2.80	59.82
Salesman (QCIF)	0.04	1.49	35.01	0.08	2.58	47.14
Stefan (QCIF)	0.05	1.52	43.97	0.08	2.37	55.18
Average	0.05	1.78	39.55	0.08	2.55	51.88

TABLE VIII
COMPARISON WITH OTHER RATE ESTIMATION METHODS

Sequence	QP	Method in [13]			Method in [11]			Method in [9]		
		PSNR variation in dB	Bit rate variation (%)	Comple-xity Reducti on (%)	PSNR variation in dB	Bit rate variation (%)	Complexity Reduction (%)	PSNR variation in dB	Bit rate variation (%)	Complexity Reduction (%)
Flower (CIF)	24	0.02	-1.12	4.91	0.12	-0.91	-1.54	0.13	-2.36	-5.91
	28	0.02	-0.50	5.06	0.13	-1.13	-0.30	0.11	-2.13	-4.10
	32	0	-0.78	4.29	0.14	-1.08	-2.40	0.15	-2.44	-4.81
Bus (CIF)	24	0.01	-0.64	4.29	0.09	-0.69	-1.31	0.13	-2.88	-4.89
	28	0.02	-1.01	5.25	0.08	-0.37	-2.10	0.08	-2.70	-3.80
	32	0.01	-0.42	6.91	0.09	-0.37	-1.71	0.11	-2.79	-5.64
Foreman (CIF)	24	0.03	-0.21	4.64	0.08	-1.24	-0.20	0.10	-4.83	-6.16
	28	0.01	-0.13	5.12	0.08	-1.74	-1.17	0.08	-4.53	-4.14
	32	0.01	-1.11	5.75	0.04	-1.92	-1.99	0.06	-3.27	-3.77
Average		0.01	-0.66	5.14	0.09	-1.05	-1.41	0.10	-3.10	-4.80

method is similar with actual RD curves. As shown in Table 6, the proposed accelerator can reduce the encoding time by around 47% (on average), in which about 14% reduction is achieved by the FSSD and about 33 % reduction is achieved by rate estimation.

B. *Results with CABAC entropy coding method*

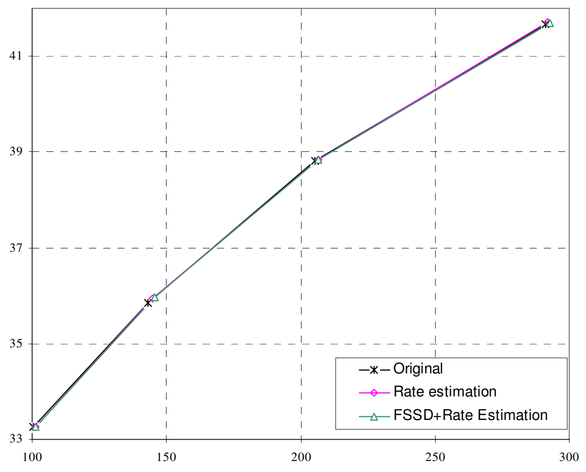
Although the proposed rate-distortion accelerator is developed based on CAVLC, it is also suitable during mode decision when Content Adaptive Binary Arithmetic Coding (CABAC) is used as entropy coding method. This is because the proposed algorithm is used only for RD optimized mode selection process. After selection of best mode, true entropy coding is used. As shown in Table 7, PSNR reduction and bit rate increment are about 0.08 db and 2.5% on average, respectively. Fig. 8 shows the comparison of RD optimized curves. As shown in Fig. 8, the rate distortion performance of proposed method is much closed with actual RD optimized curve.

C. *Comparison with other methods*

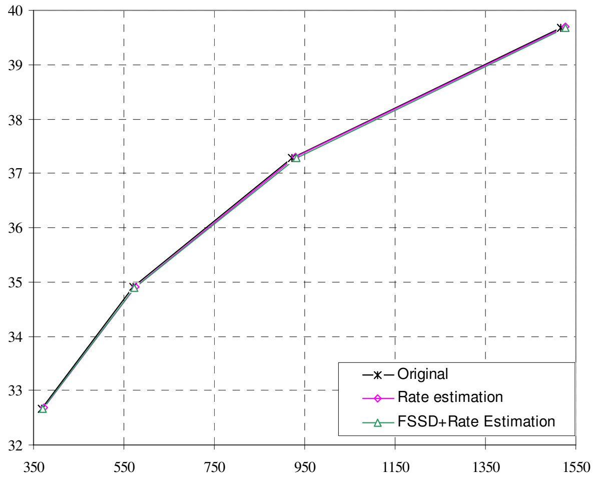
In this experiment, the proposed rate estimation method is compared with other rate estimation methods. Only rate estimation method is enabled and CAVLC is used as final entropy coding method. Complexity reduction is calculated by (11).

$$\text{Complexity reduction } (\%) = \frac{T_{ref} - T_{proposed}}{T_{ref}} \times 100\% \qquad (11)$$

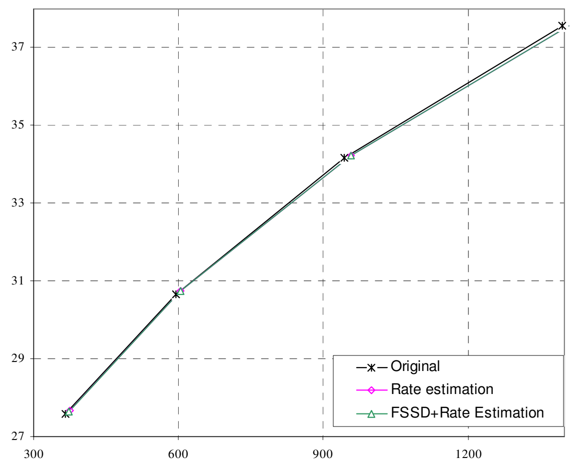
where, T_{ref} is the total encoding time of the other conventional method and $T_{proposed}$ is the total encoding time with proposed rate estimation method only.



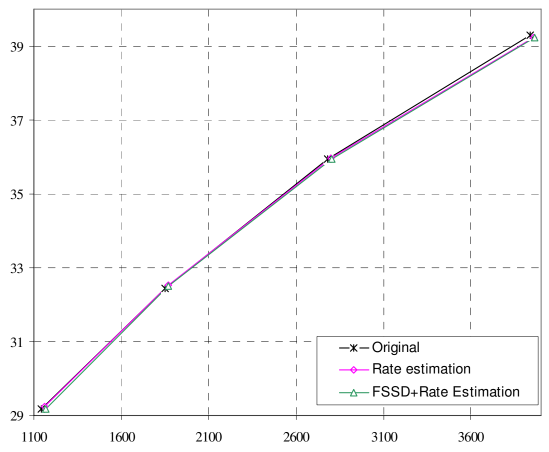
(a) Akiyo (QCIF)



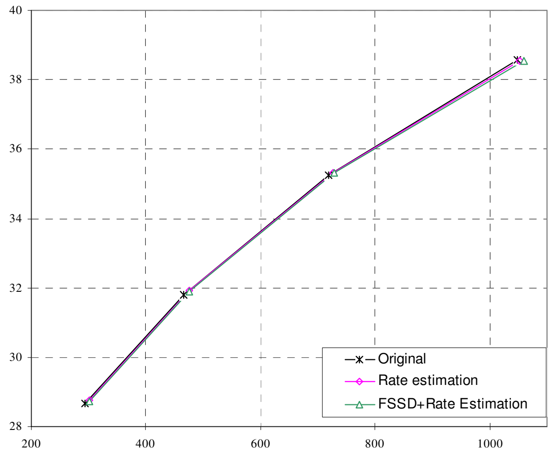
(b) Foreman (CIF)



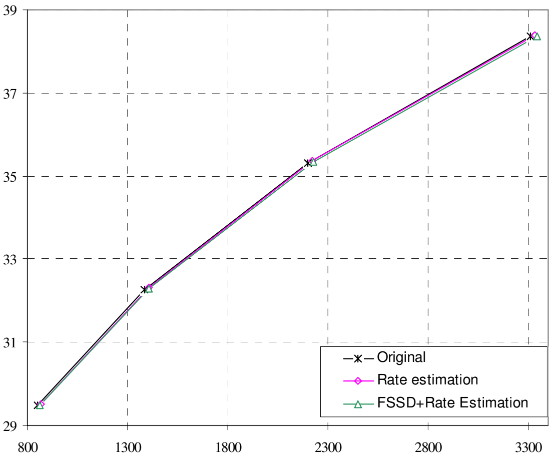
(c) Mobile (QCIF)



(d) Flower (CIF)

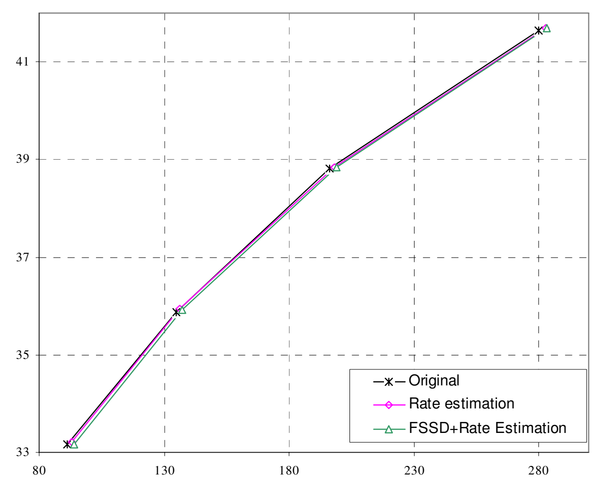


(e) Stefan (QCIF)

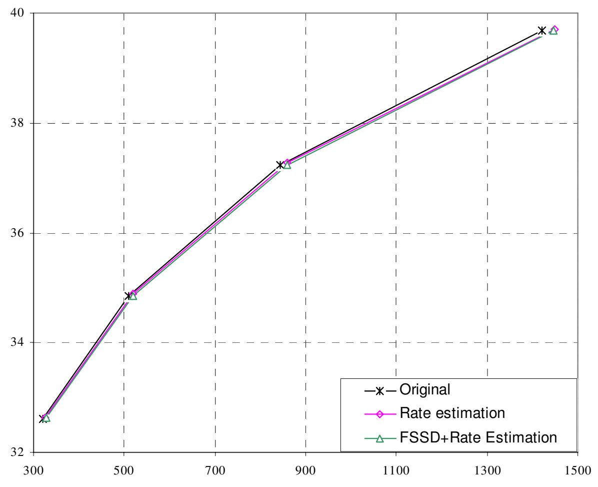


(f) Bus (CIF)

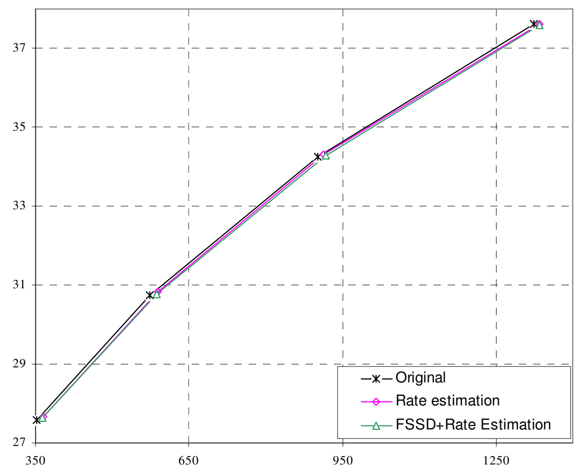
Fig.7. RD performance of proposed accelerator with CAVLC (X-axis: Bit rate in kbps, Y-axis: PSNR in dB)



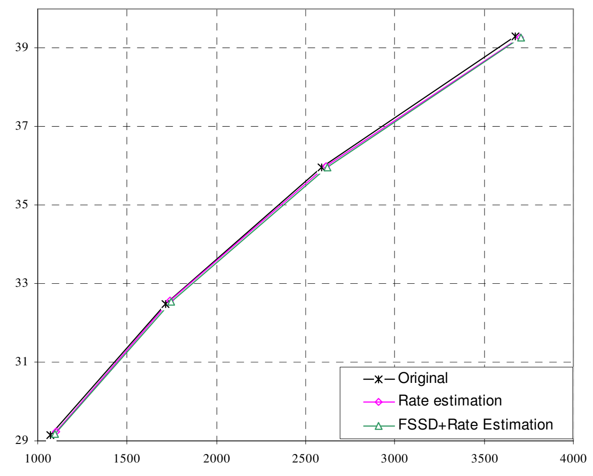
(a) Akiyo (QCIF)



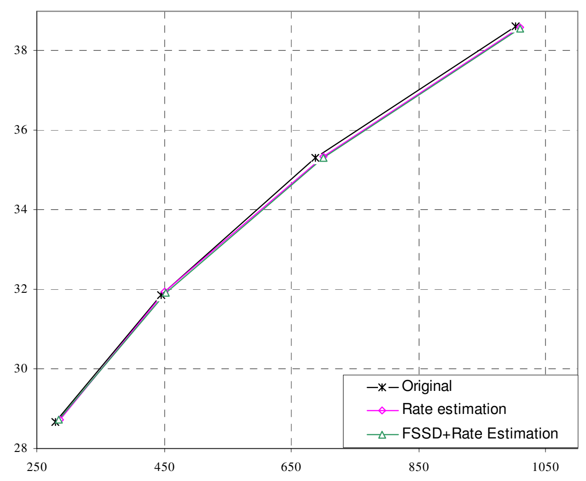
(b) Foreman (CIF)



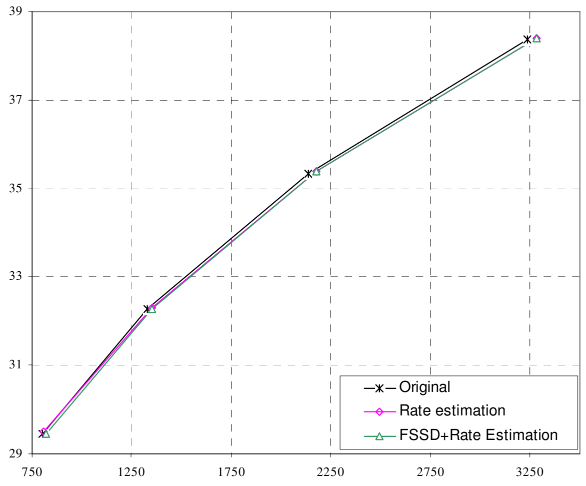
(c) Mobile (QCIF)



(d) Flower (CIF)



(e) Stefan (QCIF)



(f) Bus (CIF)

Fig. 8. RD performance of proposed accelerator with CABAC (X-axis: Bit rate in kbps, Y-axis: PSNR in dB)

Comparison results are presented in Table 8. Positive values for the PSNR and bit rate differences indicate increments, and negative values indicate decrements. Negative values of complexity reductions indicate the increment of complexity of proposed method. We have seen that the RD performance of proposed method is always better than the other methods mentioned in Table 8. As compared to [11], the proposed method increases the video quality by 0.09 dB and reduces the bit rate of 1.05% on average with more similar complexity reduction. With compared to [13], the proposed algorithm achieves a better complexity saving which is about 5% and improves the RD performance slightly. The proposed method achieves more PSNR increment (about 0.10dB) and bit rate reduction (about 3.10%) of [9] with the expanse of a little computational load.

VIII. Conclusion

In this paper, fast rate-distortion optimization method for mode decision of H.264/AVC is proposed. FSSD algorithm avoids conventional quantization, inverse quantization and inverse cosine transform. Additionally, entropy coding is replaced by fast rate estimation method. The proposed rate distortion accelerator can be used for both CAVLC and CABAC in H.264 encoding. Of course, the accuracy for CAVLC should be better but our simulation results show that the performance for CABAC with use of proposed accelerator in mode selection is similar to full RD cost encoding using CABAC. The proposed technique reduces total encoding time by about 47 % on average with negligible degradation of coding performance. The proposed method is suitable for many video application areas such as a digital camera and a mobile phone.

Acknowledgment

This work was substantially supported by a CERG grant from Hong Kong Government, Hong Kong SAR, China. [Project No.9041251].

References

[1] ISO/IEC 14496-10, Information Technology-Coding of audio-visual objects-Part:10: Advanced Video Coding. ISO/IEC JTC1/SC29/WG11 (2004)

[2] Weigand T., Sullivan, G. Bjontegaard, and G.,Luthra, A., " Overview of H.264/AVC Video Coding Standard," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 13, pp 560-576, July 2003.

[3] T.Wiegand, H.Schwarz, A.Joch, F. Kossentini, and G.J. Sullivan," Rate-constrained coder control and comparison of video coding standards," *IEEE Trans. Circuits and Syst. Video Technol.*, vol 13, no. 7, pp.688-703, July 2003

[4] Zhibo Chen, Peng Zhou, Yun He, "Fast Integer Pel and Fractional Pel Motion Estimation for JVT," Joint Video Team (JVT) Docs, *JVT-F017*, Dec. 2002.

[5] Yang L., Yu K., Li J., Li S. "An effective variable block size early termination algorithm for H.264 video coding," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 15, no. 6, pp.784-788, June 2005.

[6] Feng Pan, Xiao Lin, Susanto Rahardja, Keng Pang Lim, Z.G. Li, Dajun Wu, and Si WU, " Fast Mode Decision Algorithm for Intra-prediction in H.264/AVC Video Coding," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 15, no. 7, pp-813-822, July 2005.

[7] K.P.Lim, S. Wu, D. J. Wu, S. Rahardja, X.Lin, F.Pan, and Z.G.Li," Fast inter mode selection," *Joint Video Team(JVT)*, Doc. JVT-I020, Sep-2003.

[8] Joint Video Team (JVT) reference software version 9.6, http://iphome.hhi.de/suehring/tml/download/old_jm/

[9] C.H. Tseng, H.M. Wang, and J. F. Yang, "Enhanced Intra 4x4 Mode Decision for H.264/AVC Coders," *IEEE Trans. Circuits and Syst. Video Technol.*, vol. 16, no. 8, pp.1027-1032, August 2006.

[10] Yu-Kuang Tu, Jar-Ferr Yang and Ming-Ting Sun, "Efficient Rate-Distortion Estimation for H.264/AVC Coders", *IEEE Trans. Circuits and Syst. Video Technol.*, vol, 16 , pp. 600 – 611, May 2006.

[11] Mohammed Golam Sarwer, and Lai Man Po, " Fast bit rate estimation for mode decision of H.264/AVC," *IEEE Trans. Circuits and Syst. Video Technol.*, vol, 17 , no. 10, October 2007, pp. 1402 – 1407.

[12] Q. Chen and Y. He, " A fast bit estimation method for rate distortion optimization in H.264/AVC," in *Proceeding of Picture Coding Symp.*, 2004.

[13] Zhenyu Wei and King Ngi Ngan, "A fast rate-distortion optimization algorithm for H.264/AVC," *IEEE ICASSP 2007*, pp. I-1157-I-1160.

[14] Lai-Man Po and Kai Guo, "Transform-Domain fast sum of the squared difference computation for H.264/AVC Rate-Distortion Optimization", *IEEE Trans. Circuits and Syst. Video Technol.*, vol, 17 , no 66, pp. 765 – 773, June 2007.

[15] ITU-T Rec. H.264/ISO/IEC 11496-10, "Advanced Video Coding", *Final Committee Draft, Documents JVT-E022*, September 2002.

[16] Iain E. G. Richardson, H.264 and MPEG-4 Video Compression – video coding for next generation multimedia, John Wily & Sons, pp. 198-207, 2003.

[17] G. Bjontegaard, "Calculation of average PSNR differences between RD-curves," presented at the *13th VCEG-M33 Meeting*, Austin, TX, April 2001.

Mohammed Golam Sarwer was born in Comilla, Bangladesh in 1978. He received B.Sc. and M.Phil degree both in electronic engineering from Khulna University of Engineering and Technology and City University of Hong Kong in 2001 and 2007, respectively.
Mr. Sarwer was working as a full time faculty member in Khulna University of Engineering and Technology from 2001 to 2005. Currently, he is PhD candidate in the department of electrical and computer engineering, University of Windsor, Canada.
He is the member of International Association of Engineers (IAENG). In 2003, he won the prime minister gold medal due to outstanding academic performance of undergraduate level

Lai-Man Po (M'92) received the B.Sc. and Ph.D. degrees, both in electronic Engineering, from the City University of Hong Kong in 1988 and 1991, respectively. Since November 1991, he has been with the Department of Electronic Engineering, City University of Hong Kong, and is currently Associate Professor. His research interests are in vector quantization, motion estimation and image/video compression. Dr. Po was awarded First Prize (1988) in the Paper Contest for Students and Non-corporate Members organized by the Institute of Electronics and Radio Engineers of Hong Kong, and a Postgraduate Fellowship of the Sir Edward Youde Memorial Council (1988 to 1991).

Q.M. Jonathan Wu (M'92) received his Ph.D. degree in electrical engineering from the University of Wales, Swansea, U.K., in 1990.
From 1995, he worked at the National Research Council of Canada (NRC) for 10 years where he became a senior research officer and group leader. He is currently a Professor in the Department of Electrical and Computer Engineering at the University of Windsor, Canada. Dr. Wu holds the Tier 1 Canada Research Chair (CRC) in Automotive Sensors and Sensing Systems. He has published more than 100 peer-reviewed papers in areas of computer vision, image processing, intelligent systems, robotics, micro-sensors and actuators, and integrated micro-systems. His current research interests include 3D image analysis, active video object tracking and extraction, vision-guided robotics, sensor analysis and fusion, wireless sensor network, and integrated micro-systems.
Dr. Wu is an Associate Editor for IEEE Transaction on Systems, Man, and Cybernetics (part A). Dr. Wu has served on the Technical Program Committees and International Advisory Committees for many prestigious conferences.