An Improved Rate-Distortion Optimized Quantization Algorithm and its Hardware Implementation

Genki Moriguchi, Takashi Kambe†, Gen Fujita‡
Graduate School of Science and Engineering, Kindai University
3-4-1 Kowakae, Higashi-Osaka City, Osaka, Japan
†Depart. of Electric and Electronic Engineering, Kindai University
‡Faculty of Information and Communication Engineering,
Osaka Electro-Communication University, Osaka, Japan

Abstract— Rate-distortion optimized quantization (RDOQ) is an important technology in H.264/AVC for improving video coding performance. RDOQ is able to determine the optimal value among multiple quantization candidates based on rate-distortion (RD). We propose improvements to the algorithm to reduce its complexity by changing the bit-rate estimation method and by excluding low scored quantization candidates. We also implement the algorithm in hardware using the Bach C high-level synthesis tool. Finally, the performances of the proposed algorithm and hardware design results are evaluated.

I. Introduction

H.264/AVC [1] has achieved a data compression rate of 1/80 for high definition TV due to the progress in video compression technology. Although the Rate-Distortion Optimized Quantization (RDOQ) in H.264/AVC improves the compression rate, processing times are nevertheless high and implementing it as hardware is complex.

In this paper we improve the RDOQ algorithm by reducing its complexity for hardware implementation while maintaining its performance. In addition, the proposed RDOQ is implemented in hardware and its performance is improved using a high-level synthesis.

II. Rate-Distortion optimized quantization (RDOQ)

The quantization process in the H.264/AVC reference software JM18.0 [2] is implemented using the rate-distortion optimization technique (RDOQ). RDOQ is able to determine the optimal quantization level among multiple quantization candidates by minimizing the sum of rate-distortion (RD) costs in each block.

As shown in Fig.1, RDOQ consists of five steps to achieve higher encoding performance than conventional approaches.

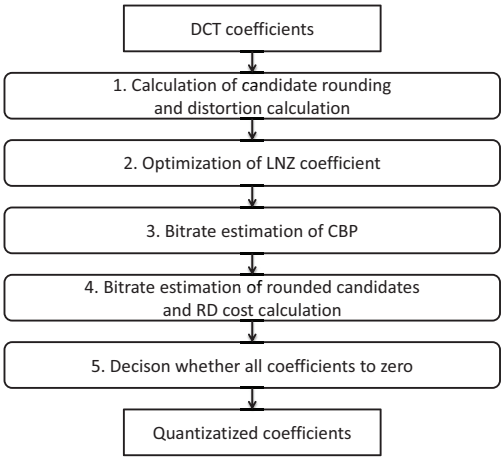

Fig. 1. The RDOQ process in JM18.0

1. Three l_i^{float} rounding candidates, (a) zero level l_i^0, (b) floor rounding level l_i^{floor}, and (c) ceiling rounding level l_i^{ceil} are enumerated and their distortion values D_i^j are calculated. Here, i denotes the coefficient number, l_i^{float} denotes the quotient (before rounding) of DCT coefficient c_i divided by quantization step Q_{step}, and j denotes the rounding candidate level.

2. The last non-zero(LNZ) coefficient is found in zigzag scanning order to minimize RD cost.

3. CBP (Coded Block Pattern) is estimated.

4. The bit-rate estimation of each rounding candidate and its RD cost is calculated using equation (1).

$$J_i^j = err_i^j + \lambda \times bits_i^j \quad (1)$$

where, $bits_i^j$ represents the number of bits obtained by performing entropy coding on the quantized level l_i^j, err_i^j indicates the quantization error if the coefficient c_i is quantized to value l_i^j, and λ is constant for each quantization parameter (QP).

5. Determine whether or not all coefficients are set to zero by comparing steps 1 to 4 with the sum of RD costs.

RDOQ has a higher encoding performance than conventional techniques but it involves complex processes such as optimizing the quantization value for each quantization coefficient and updating the context of each coefficient. Also, because coefficients for bit-rate calculation are dependent on each other, hardware acceleration using parallel processing is difficult.

III. Improvement of RDOQ algorithm

In this section, several kinds of improvement to the RDOQ algorithm are proposed. [6] proposed reduction of quantization level candidates and simplification of bit-rate estimation. In this paper, the bit-rate estimation approach in [6] is improved on further, and three new improvements, (a) acceleration of distortion calculation, (b) reduction of LNZ coefficients search range, and (c) decision about whether all coefficients are zero, are also proposed.

A. Reduction of quantization level candidates [6]

When quantization level l_i^{float} is greater than 1.5, the possibility of selecting quantization level l_i^0 is very low. In that case, only l_i^{floor} and l_i^{ceil} should be candidates. Because the number of candidates is reduced from 3 to 2, the equation for quantization level selection (2) can be formulated using only the difference between the values.

$$\Delta J_i = J_i^{floor} - J_i^{ceil} = \Delta err_i + \lambda \times \Delta bits_i \qquad (2)$$

where, $\Delta\ err_i$ and $\Delta\ bits_i$ represent the quantization distortion and bit-rate differences. From equation (2), the calculation complexities of distortion and RD cost are also reduced by using only the difference instead of the sum for RD cost calculation.

Table I shows the probability that l_i^0 is selected as quantization level, when l_i^{float} is greater than 1.5 in JM18.0 software using several video sequences. Table II shows that the reduction in the number of distortion calculations achieved when our proposed method is applied to several video sequences is between 13% and 27% compared with JM18.0.

TABLE I
Probability of l_i^0

Sequence	Resolution	Probability[%]
news	CIF	0.0013
container	CIF	0.0022
football	CIF	0.0407
paris	CIF	0.0371
PartyScene	832x480	0.2236
In to tree	720p	0.0114

TABLE II
Reduction rate of distortion calculation

Sequence	JM18.0	Proposed	Reduction [%]
news	12,021,409	10,404,051	13.5
container	12,653,105	10,681,845	15.6
football	14,320,665	11,417,882	20.3
paris	14,178,314	10,734,385	24.3
PartyScene	61,147,492	44,814,814	26.7
In to tree	112,233,107	81,594,873	27.3

B. Acceleration of distortion calculation

The distortion err_i calculation for RD cost in the first RDOQ step requires many multiplications for the three rounding candidates. To speed up this calculation, the difference in quantization distortion between l_i^{floor} and l_i^{ceil} is calculated using the following equation (3).

$$\begin{aligned}\Delta err_i &= err_i^{floor} - err_i^{ceil} \\ &= (l_i^{floor} - l_i^{float})^2 - (l_i^{floor} + 1 - l_i^{float})^2 \times Qstep^2 \\ &= (2 \times \Delta l_i - 1) \times Qstep^2\end{aligned} \qquad (3)$$

where, $\Delta\ l_i$ denotes the difference between l_i^{floor} and l_i^{float} before rounding, and $\Delta\ l_i$ denotes the fraction that is rounded off to l_i^{floor}. A floating point calculation is needed for this operation. We propose calculating $\Delta\ err_i$ using a bit-shift operation instead of division as follows:

$$\begin{aligned}\Delta err_i &= [(2 \times |c_i| \wedge (Qstep - 1) - Qstep] \times Qstep \\ &= [\{(|c_i| \wedge (Qstep - 1)) << 1\} - Qstep] << Qbits\end{aligned} \qquad (4)$$

where, $Qbits$ represents the $Qstep$ bit shift value. By using only a shift operation, the hardware implementation of the $\Delta\ err_i$ calculation can be accelerated.

C. Acceleration of LNZ coefficient search

The 2nd step of the RDOQ process searches for the LNZ coefficient. Based on quantization level $l_{i,float}$, three levels (numlevs) are defined in Table III. The definitions of Condition 1 and 2 are explained later.

For example, the search is done from the last $numlevs = 3$ to the next $numlevs = 2$ in the zigzag scanning order shown in Fig.2.

l_i^{floor}, $err(c_i, 0)$ and $err(c_i, floor)$ are needed for this search. However, coefficients that have small values after

TABLE III
THE LEVEL DEFINITION OF l_i^{float}

numlevs	Condition 1		Condition 2
	Integer	Fractional	
3	1 or more	0.5 or more	1.5 or more
2	1 or more	less than 0.5	0.5 or more
	less than 1	0.5 or more	and less than 1.5
1	0	less than 0.5	less than 0.5

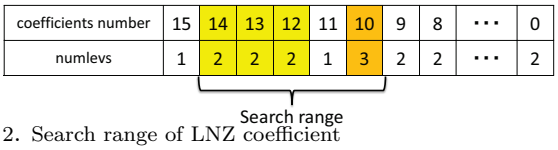

Fig. 2. Search range of LNZ coefficient

the last $numlevs = 3$ is found, are out of the search range. Neither $err(c_i, 0)$ nor $err(c_i, floor)$ are needed to calculated, so (4) can be used for the distortion calculation.

C..1 Reverse zigzag scanning order

Although distortion is calculated for each rounding candidate initially using reverse zigzag scanning order, the other distortions can be calculated using equation (4) after quantization level $numlevs = 3$ appears.

TABLE IV
EQUATION (4) APPLICATION RATE

Sequence	JM18.0[%]	Reverse order[%]	Search range[%]
news	1.9	11.9	13.6
container	2.4	14.9	17.0
football	3.7	21.9	24.9
paris	3.3	25.4	28.7
PartyScene	4.3	31.9	35.8
In to tree	1.6	6.7	7.8

C..2 Search range reduction

Another acceleration method is search range reduction. In JM18.0, when the quantization level is $numlevs = 2$, condition 1 in Table III is applied. However, when quantization level l_i^{float} is greater than 1.5, the probability that it is optimized to 0 is low. Therefore the level definition of numlevs is changed to condition 2 in Table III. Table IV shows the equation (4) application rate when reverse zigzag scanning order and search range reduction are applied. The results are several times better than JM18.0.

D. *Simplification of bit-rate estimation*

A fast bit-rate estimation method to avoid the CABAC process was proposed in [6]. A more accurate method than [6] is proposed in this section.

D..1 Bit-rate estimation

The bit-rate is mainly related to four syntax elements, significant_coeff_flag, last_significant_coeff_flag, coeff_greater_one_flag, and coeff_abs_level_minus2. Coeff_greater_one_flag indicates whether or not the absolute value of quantized level l_i is zero and the coeff_abs_level_minus2 has a value which is 2 less than the absolute value of quantized level l_i.

The bit-rate of the CABAC encoder is estimated using entropy of probability distribution based on context model. Therefore, the bit-rate B can be estimated using the equation for the distribution probability P (5).

$$B = -log_2 P \tag{5}$$

Assume that the symbols $P_{sc}, P_{ls}, P_{cg}, P_{ca}$ signify the distribution probabilities for significant_coeff_flag, last_significant_coeff_flag, coeff_greater_one_flag, and coeff_abs_level_minus2 respectively, and the symbols $B_{sc}, B_{ls}, B_{cg}, B_{ca}$ indicate the corresponding bit-rates. B_{ca} consists of B_{ca}^1 and B_{ca}^0. B_{ca}^{1or0} means the number of one (or zero) bits generated when coeff_abs_level_minus2 is encoded. The total number of bits for a quantized level l_i, measured in $Bits$, is formulated as

$$\begin{aligned} Bits_i = {} & B_{sc} + B_{ls} + B_{cg} \\ & + (l_i - 2) \times B_{ca}^1 + B_{ca}^0 \end{aligned} \tag{6}$$

[6] used B_{ca}^1 only in the equation (6) and had the problem that the estimated bit-rate of coeff_abs_level_minus2 becomes zero when the quantized level l_i is two. Our method avoids this problem by using B_{ca}^0. The calculation of the distribution probabilities of $P_{sc}, P_{ls}, P_{cg}, P_{ca}$ is formulated in equations (7) - (10) below and the list of symbols for them is shown in Table V.

TABLE V
SYMBOL LIST

N_l	Cumulative number of zero level in front of the last non-zero coefficient
N_o	Cumulative number of absolute level 1
N_g	Cumulative number of absolute level 2 or higher
N_b	Cumulative number of blocks that non-zero coefficients exists
S_g	The sum of the absolute level greater than 1

In JM18.0, the coeff_abs_level_minus2 takes its estimated bit-rate from a look-up table when the quantized

$$P_{sc} = \begin{cases} \dfrac{N_l}{N_l + N_o + N_g} & significant\_coeff\_flag = 0 \\ 1 - \dfrac{N_l}{N_l + N_o + N_g} & significant\_coeff\_flag = 1 \end{cases} \quad (7)$$

$$P_{ls} = \begin{cases} 1 - \dfrac{N_b}{N_o + N_g} & last\_significant\_coeff\_flag = 0 \\ \dfrac{N_b}{N_o + N_g} & last\_significant\_coeff\_flag = 1 \end{cases} \quad (8)$$

$$P_{cg} = \begin{cases} \dfrac{N_g}{N_o + N_g} & coeff\_greater\_one\_flag = 0 \\ 1 - \dfrac{N_g}{N_o + N_g} & coeff\_greater\_one\_flag = 1 \end{cases} \quad (9)$$

$$P_{ca} = \begin{cases} \dfrac{N_g + 1}{S_g + 1} & coeff\_abs\_level\_minus2 = 0 \\ 1 - \dfrac{N_g + 1}{S_g + 1} & coeff\_abs\_level\_minus2 = 1 \end{cases} \quad (10)$$

level l_i is less than 27, but uses *exponential Golomb coding* when it is greater than 27. The greater the quantized level l_i, the longer *exponential Golomb coding* takes to calculate the estimated bit rate. We propose applying only one calculation for each estimated bit-rate for any size of quantized level. The computational complexity of this bit-rate estimation is much lower than that in JM18.0. In the JM18.0 RDOQ, the bit-rate estimation depends on the position information for each of the coefficients in significant _coeff _flag and last _significant _coeff _flag. The coeff_greater_one_flag and coeff_abs_level_minus2 have dependency among coefficients for all estimations.

The proposed method uses only the estimated bit-rate from equations (5), (7) - (10) for all coefficients. Thus, register access for the estimated bit-rate is done only once for each block and the number of gates required for storing bit-rate information can be reduced. Further, since there is no dependency between coefficients, this enables parallel processing of all the coefficients in the block when implemented in hardware. Table VI shows that the reduction in data access of the estimated bit-rate is over 90%.

CBP (Coded Block Pattern) is also used to estimate the bit-rate. CBP is another syntax element in H.264, which specifies which sub-macroblocks may contain non-zero transform coefficient levels. Because information about neighbor blocks is required to calculate this, multiple data accesses are needed. However, CBP does not have a large impact on the accuracy of the estimation. So our proposed method eliminates CBP estimation to reduce the computational complexity.

D..2 Lambda Scaling

RDOQ calculates the RD cost using equation (1). In general, the quantization error is larger than the quantization distortion. Therefore λ scaling is applied to match the scale between the two parameters. The proposed method is faster and simpler than JM18.0 and λ is adjusted to maintain estimation accuracy based on several kinds of video sequences.

TABLE VI
REDUCTION OF DATA ACCESS OF ESTIMATED BIT-RATE

significant_coeff_flag			
Sequence	JM18.0	Proposed	Reduction [%]
news	4,208,705	276,031	93.4
container	5,825,528	322,797	94.5
football	8,348,793	398,406	95.2
paris	7,226,950	364,320	95.0
PartyScene	38,608,598	1,580,599	95.9
In to tree	62,593,826	3,147,906	95.0
last_significant_coeff_flag			
Sequence	JM18.0	Proposed	Reduction[%]
news	3,502,458	276,031	92.1
container	4,628,884	322,797	93.0
football	6,915,035	398,406	94.2
paris	6,189,006	364,320	94.1
PartyScene	32,242,538	1,580,599	95.1
In to tree	41,993,888	3,147,906	92.5

D..3 All coefficients to zero decision

The decision whether or not to make all coefficients zero is made using equation (11). In our approach, $bits_{CBP}$ is set to zero for the reason described in the previous section.

$$\sum_{i=0}^{M} err_{i,0} + \lambda \times bits_{CBP} < \sum_{i=0}^{M} J_{i,best} \quad (11)$$

Both the difference distortion and zero distortion are calculated from this equation. However, when the quantized level l_i^{float} is greater than 1.5, encoding accuracy can be maintained even after skipping this decision. Thus we are able to use the difference distortion calculation. Table VII show that BD-BR [7] is not degraded by skipping this process.

TABLE VII
BD-BR BY SKIPPING ZERO DECISION

Sequence	BD-BR[%]
news	-0.063
container	0.073
football	-0.109
paris	0.072
PartyScene	-0.013
In to tree	-0.090
Average	-0.022

IV. Hardware implementation using high-level synthesis technology

This section describes several kinds of hardware implementation of the proposed RDOQ algorithm on a 4×4 block for inter-frame prediction using high-level synthesis technology.

A. Parallel processing

Because there are no dependencies among coefficients, the proposed RDOQ algorithm can calculate the RD costs of all coefficients in parallel. However, when enumerating rounding candidates and deciding the distortion calculation LNZ coefficient search range there are data dependencies among coefficients. To deal with this problem, a search range flag is set at rounding candidates enumeration and the LNZ coefficient search range is determined using this flag after distortion calculation. This modification eliminates the data dependency between the two processes and they can process all the coefficients in parallel. One-cycle loop pipelining is also applied. All flags are bit-concatenated in reverse coefficient number order and the search range is decided using a one-cycle priority encoder.

B. Function based pipelining

The RDOQ process is implemented using four-stage functional pipelining. The pipeline consists of (a) rounding candidates enumeration and distortion calculation, (b) optimization of LNZ coefficient, (c) bit-rate estimation and RD cost calculation, and (d) decision of all coefficients to zero. The stages communicate using synchronization signals.

In addition, higher level functional level pipelining is applied to RDOQ processing for 4×4 blocks. RDOQ has data dependencies for intra-frame prediction, and we will propose a solution in our next paper.

V. Design results of the improved algorithm and its evaluation

In this section, the performance of the proposed algorithm is evaluated.

A. Performance of the proposed RDOQ algorithm

CABAC has different probability information for stream information even if it is the same syntax element. The proposed method uses one probability. As a result, there is no data dependency among coefficients and by λ scaling the RD cost calculation, the estimation performance can be kept high.

Table VIII shows the environment for measuring coding performance. The coding performance of the proposed method and of the JM18.0 RDOQ are compared. Table IX shows that, on average, the differences in BD-BR and BD-PSNR of the proposed method from JM18.0 are 0.11% and -0.00239% respectively.

TABLE VIII
Measurement of coding performance

Compile	Visual Studio 2010 (Debug mode)
CPU	Core i7-3770 (3.40GHz)
Number of frames	50 frames
QP	22,27,32,37
QP Optimizarion	5 patterns

TABLE IX
Coding performance of the proposed method

Sequence	Resolution	BD-BR[%]	BD-PSNR[dB]
news	CIF	-0.19	0.00099
container	CIF	-0.03	0.00072
football	CIF	0.42	-0.02644
paris	CIF	0.16	-0.00793
Desert	720x400	0.70	-0.02857
PartyScene	832x480	0.18	-0.01194
BBCPan13	720x576	-1.12	0.03516
old town cross	720p	0.28	-0.01194
In to tree	720p	0.36	0.00692
rush hour	1080p	0.33	-0.01051
crowd run	1080p	0.15	-0.00561
Average		0.11	-0.00239

By removing the context information data dependency, the bit-rate estimation processing time is reduced by about 77%. The time for the distortion calculation is reduced about 2%. The reduction is small probably because the modifications to the equation are oriented to hardware acceleration. LNZ coefficient and RD cost calculations are speeded up about 43% due to the search range and estimated bit-rate memory access reductions. Eliminating the CBP bit-rate estimation provides about 30% speed-up.

B. Hardware design results

Table XI shows the hardware design environment.

Four circuits in Table XII were designed using the proposed design methods. The design result of circuit v4 will be added in the final paper.

Circuit v0 is a sequential Bach C description of the proposed algorithm. Circuit v1 executes 16 processes in parallel to process the 4×4 block coefficients in steps 1 and 4 in Fig.1, applies bit width optimization to each variable and uses on-chip memory. Although the gate size is bigger, the processing speed is about 77% faster

TABLE X
Comparison of processing time

Sequence	Resolution	Reduction rate[%]
news	CIF	31.01
container	CIF	31.57
football	CIF	32.43
paris	CIF	33.18
Desert	720x400	28.93
PartyScene	832x480	33.48
BBCPan13	720x576	31.46
old town cross	720p	29.84
In to tree	720p	29.45
rush hour	1080p	27.13
crowd run	1080p	31.65
Average		30.92

TABLE XI
The Design environment

Behavioral synthesis	Bach C System ver3.6 [3]
RTL simulation	ModelSim
Logic synthesis tool	Design Compiler 2008
Operating frequency	100 MHz
Cell library	Hitachi 0.18μ CMOS library
Number of frame	1 frame(6248 4x4blocks)

than circuit v0. In circuit v2 one cycle loop pipelining is applied at step 2 by moving multi-cycle multiplications out of loop. It is about 18% faster than circuit v1 and the gate size increases by about 1%. Circuit v3 implements a priority encoder for search range reduction. It is about 7% faster than circuit v2 and 82% faster than v0.

VI. Conclusion

In this paper, we propose several algorithmic improvements to reduce the complexity of the RDOQ process by changing the bit-rate estimation method and by excluding low scored candidates for quantization. The algorithm was also implemented in hardware using the Bach C high-level synthesis tool. The proposed RDOQ algorithm is about 30% faster than JM18.0 while still maintaining encoding performance. Although the designs in this paper used a 4×4 block size, we plan to apply our methods to other block sizes. We also plan to use these techniques and apply them to the next generation video encoding standard HEVC [8].

TABLE XII
Design results

Version	Circuit size[gate]	Processing time[μs]
v0(Sequential)	158,734	37,111
v1(16 parallelization)	778,654	8,699
v2(Loop pipelining)	794,006	7,126
v3(Priority encoding)	785,035	6,626
v4(Pipelining)	-	-

Acknowledgements

The authors would like to thank the Bach system development group in SHARP Corporation, Electronic Components and Devices Development Group, for their help in hardware design using the Bach system. This work is supported by the VLSI Design and Education Center (VDEC), the University of Tokyo in collaboration with Synopsys Corporation.

References

[1] I. E. Richardson: "The H.264 Advanced Video Compression Standard," Publisher: Wiley; 2 edition (August 9, 2010).

[2] JVT reference software version 18.0
http://iphome.hhi.de/suehring/tml/download

[3] K.Okada, et al.: "Hardware Algorithm Optimization Using Bach C," IEICE Trans. Fundamentals vol.E85-A, No.4, pp835-841, 2002.

[4] Marta KarczewiczYan YePeisong Chen: Rate Distortion Optimized QuantizationJVT-AA026

[5] Limin Liu, and Alexis Tourapis: "Rate distortion optimized quantization in the JM reference software, JVT-AA027, April 24-29, 2008.

[6] Jing He, Fuzheng Yang: High-speed implementation of rate-distortion optimized quantization for H.264/AVC ©Springer-Verlag London 2013

[7] G. Bjontegaard: Calculation of average PSNR difference between RD-curves, VCEG-M33, Austin, Texas, USA, April 2001.

[8] Sullivan, G. J., et al.: "Overview of the High Efficiency Video Coding (HEVC) Standard." Circuits and Systems for Video Technology, IEEE Transactions on 22(12), pp.1649-1668, 2012.