J Real-Time Image Proc (2009) 4:3–12
DOI 10.1007/s11554-008-0098-5

SPECIAL ISSUE

Low-complexity quantization for H.264/AVC

Yun Zhang · Gangyi Jiang · Mei Yu

Received: 4 February 2008 / Accepted: 8 September 2008 / Published online: 30 September 2008

Abstract In this paper, an improved quantization technology with low-complexity is presented for H.264/AVC video codec. Multiplication factors of H.264/AVC quantizer are modified. Therefore, it is possible to reduce the bit width of the quantization and substitute large bit-width multiplier by some small bit-width adders without noticeable rate-distortion degradation in integrated circuits (ICs) design. Quantization error introduced by the modified multiplication factors is not only theoretically but also experimentally analyzed. Quantizer is optimized on register transfer level of IC design, and under the same cell CMOS technology, about 75.2% area and 76.3% dynamic power consumption are saved in each quantization unit on average compared with original H.264/AVC quantization. Experimental video coding results show that the Bjontegaard delta peak signal-to-noise ratio (PSNR) and Bjontegaard delta bit rate between the improved and original H.264/AVC quantization are very slight, which means that the improved quantization scheme is approximately the same as the original quantization scheme of H.264/AVC in rate-distortion performance.

Y. Zhang · G. Jiang · M. Yu
Institute of Computing Technology, Chinese Academic of Science, 100080 Beijing, China

G. Jiang (✉) · M. Yu
Faculty of Information Science and Engineering, Ningbo University, 315211 Zhejiang, China
e-mail: jianggangyi@126.com

Y. Zhang
Graduate School of Chinese Academic of Science, 100080 Beijing, China

Keywords H.264/AVC video coding · Integer transform · Quantization · Low-complexity

1 Introduction

H.264/AVC is the latest international video coding standard developed jointly by ITU-T Video Coding Experts Group and the ISO/IEC Moving Picture Experts Group [1]. Many techniques are adopted in H.264/AVC such as the 4×4 integer transform and quantization [2, 3], rate-distortion optimization (RDO), the deblocking filter, enhanced intra prediction [4], variable block sizes and multiple reference frames for motion compensation [5], and context-based adaptive binary arithmetic code (CABAC). However, the superior compression efficiency of H.264/AVC is at the expense of very high computational complexity. Therefore, optimizing H.264/AVC coding arithmetic and reducing the computational complexity are the hotspots in the video coding fields.

Many researches have been focused on fast inter and intra prediction schemes to speed up H.264/AVC encoder [6–15], because the two processes are most important but complex in the encoder. Besides these, other processes are also optimized. Wang proposed a method which can early detect zero quantized discrete cosine transform (DCT) coefficients before implementing DCT and quantization so as to reduce redundant computations for H.264/AVC [16]. Chen et al. [17] presented a fast and low cost context adaptive binary arithmetic encoder for H.264/AVC through both algorithm level and architecture level optimizations. Fan et al. [18] developed fast 1-D and 2-D 4×4 forward integer transform algorithms by utilizing matrix factorization. Since the scaling matrix in forward integer transform is merged to the quantization

process in H.264/AVC standard, quantization has been a core process for integer transform. Even though the quantization process is relatively simple in H.264/AVC standard compared with other processes, it is indispensable, thus, the optimization of quantization is also a valuable work for complexity reduction, which is the focus of this paper.

Compared with conventional DCT, there are some primary advantages by adopting integer transform that can be listed as follows: (1) all operations can be carried out with integer arithmetic, without loss of accuracy and occurrence of mismatch between encoders and decoders, (2) no floating-point arithmetic, which speeds the codec up and is thus more suitable for the implementation on hardware platform. Integer transform leads to a significant complexity reduction, with an impact in peak signal-to-noise ratio (PSNR) of less than 0.02 dB [2]. The core part of the transform is multiply free [2], i.e., it only requires additions and shifts, which fits to the integrated circuit (IC) design. However, quantization procedure of H.264/AVC still has multiplications. Multiplications not only cost a great deal of power and area when implemented on very large scale integration (VLSI), but also compromise the performance of the core part of the integer transform in pipeline architecture. In this paper, a low-complexity quantization scheme is proposed for VLSI design. By modifying the multiplication factors, the multiplication in quantization procedure is implemented with some shifts and small bit-width additions, so that the complexity of the quantizer is significantly reduced. Quantization error introduced by modifying multiplication factor is theoretically and experimentally analyzed in order to control rate-distortion degradation.

The rest of the paper is organized as follows. We review the 4 × 4 residual integer transform and quantization in H.264/AVC in the next section. In section 3, the problems for the H.264/AVC quantization is described, then, low-complexity quantization architecture is presented. Moreover, mismatch error caused by the improved quantization is analyzed in detail. Then, experimental comparisons on area, power consumption and rate-distortion performance between the original and improved quantization are given in Sect. 4. Finally, conclusions are drawn in Sect. 5.

2 Overview of the 4 × 4 integer transform and quantization in H.264/AVC

The 4 × 4 forward integer transform $\mathbf{Y}$ of an input residual signal $\mathbf{X}$ in H.264/AVC is given by [2]

$$\mathbf{Y} = (\mathbf{C}_f\mathbf{X}\mathbf{C}_f^T) \otimes \mathbf{E} = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 2 & 1 & -1 & -2 \\ 1 & -1 & -1 & 1 \\ 1 & -2 & 2 & -1 \end{bmatrix} [\mathbf{X}] \begin{bmatrix} 1 & 2 & 1 & 1 \\ 1 & 1 & -1 & -2 \\ 1 & -1 & -1 & 2 \\ 1 & -2 & 1 & -1 \end{bmatrix} \otimes \begin{bmatrix} a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \\ a^2 & ab/2 & a^2 & ab/2 \\ ab/2 & b^2/4 & ab/2 & b^2/4 \end{bmatrix}, \quad (1)$$

where $a = 1/2$, $b = \sqrt{2/5}$. $(\mathbf{C}_f\mathbf{X}\mathbf{C}_f^T)$ is a 'core' 2D transform, $\mathbf{E}$ is a matrix of scaling factors, and the symbol $\otimes$ indicates that each element of $(\mathbf{C}_f\mathbf{X}\mathbf{C}_f^T)$ is multiplied by the scaling factors at the same position in the matrix $\mathbf{E}$. Equation (1) is an approximation to the 4 × 4 DCT. The inverse transform $\mathbf{X}'$ is given by

$$\mathbf{X}' = \mathbf{C}_i^T(\mathbf{Y} \otimes \mathbf{E}_i)\mathbf{C}_i = \begin{bmatrix} 1 & 1 & 1 & 1/2 \\ 1 & 1/2 & -1 & -1 \\ 1 & -1/2 & -1 & 1 \\ 1 & -1 & 1 & -1/2 \end{bmatrix} \left([\mathbf{Y}] \otimes \begin{bmatrix} a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \\ a^2 & ab & a^2 & ab \\ ab & b^2 & ab & b^2 \end{bmatrix} \right) \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1/2 & -1/2 & -1 \\ 1 & -1 & -1 & 1 \\ 1/2 & -1 & 1 & -1/2 \end{bmatrix} \quad (2)$$

Let *Qstep* be a quantization step size, with a total of 52 values that are supported by the H.264/AVC video coding standard and indexed by a quantization parameter, QP. Let Y_{ij} be a coefficient of the 2D transform, W_{ij} be an unscaled coefficient after the core transform $(\mathbf{C}_f\mathbf{X}\mathbf{C}_f^T)$, and Z_{ij} be a quantized coefficient. Then, the forward quantization operation in H.264/AVC is represented as

$$\begin{aligned} Z_{ij} &= \text{round}\left(\frac{Y_{ij}}{Qstep}\right) = \text{round}\left(W_{ij}\frac{PF_{ij}}{Qstep}\right) \\ &= \text{round}\left(W_{ij}\frac{MF_{ij}}{2^{qbits}}\right) \end{aligned} \quad (3)$$

where PF_{ij} is a^2, $ab/2$ or $b^2/4$ depending on the position (i, j) in 4 × 4 block, and $qbits = 15 + \text{floor}(QP/6)$. MF_{ij} is a multiplication factor at position (i, j), $MF_{ij} = PF_{ij} \cdot 2^{qbits}/Qstep$, defined as in Table 1. With addition and shift operations rather than division operation, the above quantization can be implemented by

$$Z_{ij} = \text{sign}(W_{ij})(|W_{ij}|MF_{ij} + f) \gg qbits, \quad (4)$$

where '≪' indicates a binary shift-right operation, $f = 2^{qbits}/3$ for intra blocks or $f = 2^{qbits}/6$ for inter blocks. The inverse quantizer is defined as

Table 1 Multiplication factor MF_{ij} in H.264/AVC

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	13,107	5,243	8,066
1	11,916	4,660	7,490
2	10,082	4,194	6,554
3	9,362	3,647	5,825
4	8,192	3,355	5,243
5	7,282	2,893	4,559

Table 2 Rescaling factor V_{ij} in H.264/AVC

QP	Positions (0,0),(2,0),(2,2),(0,2)	Positions (1,1),(1,3),(3,1),(3,3)	Other positions
0	10	16	13
1	11	18	14
2	13	20	16
3	14	23	18
4	16	25	20
5	18	29	23

$$Y'_{ij} = Z_{ij}Qstep. \tag{5}$$

Furthermore, the pre-scaling factor (**E**) is incorporated for the inverse transform, and then W'_{ij}, as an inverse quantization result of W_{ij}, is computed by

$$W'_{ij} = Z_{ij}QstepPF_{ij}64 = Z_{ij}V_{ij}2^{\mathrm{floor}(QP/6)}. \tag{6}$$

The H.264/AVC standard does not specify $Qstep$ or PF_{ij} directly. Instead, the parameter $V_{ij}2^{\mathrm{floor}(\mathrm{QP}/6)} = QstepPF_{ij}64$ is defined, shown as in Table 2. After the inverse core transform, defined as $\mathbf{X}' = \mathbf{C_i}^{\mathbf{T}}\mathbf{W}'\mathbf{C_i}$, we can get 4×4 residual samples, $\mathbf{X}''$, output by a post-scaling operation, defined as $\mathbf{X}'' = \mathrm{round}(\mathbf{X}'/64)$.

The core part of integer transform is a simplex multiply free orthogonal transform, and it can be implemented only with shifts and additions. Thus, the core transform can be efficiently implemented by butterfly architecture with parallel pipelining. However, there are multiplications in quantization module which not only increase computational complexity but also cause long delay. Multiplier is also a bottleneck of the pipeline of integer transform and quantization architecture. Therefore, an efficient approach to quantizer optimization is required.

3 The proposed low-complexity quantization for H.264/AVC

In this section, the problem of quantization design is presented first. Then, technical solution for low-complexity quantization design is proposed by modifying multiplication factor. However, this will cause mismatch error between quantizer and de-quantizer. Therefore, the mismatch error is analyzed and constraints for multiplication factor modification are given. Finally, we present a group of typical modified multiplication factors.

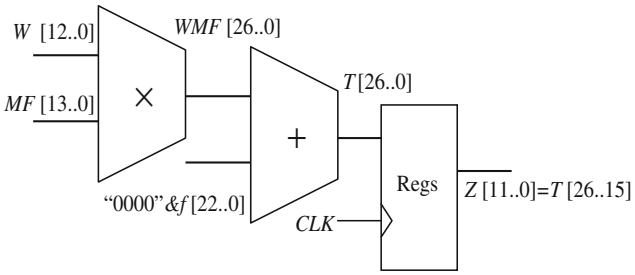

Fig. 1 Diagram of H.264/AVC quantization unit architecture with common implementation ($MF_{ij}/2^{qbits} = 13107/2^{15+\mathrm{floor}(\mathrm{QP}/6)}$)

3.1 Problem description and solution

As introduced in Sect. 2, the H.264/AVC quantization can be realized according to (4). Here, we take the quantization unit $MF_{00} = 13{,}107$ as an example to illustrate how to implement the H.264/AVC quantization in hardware. Figure 1 shows the diagram of H.264/AVC quantization unit architecture with respect to $MF_{00} = 13107$. In the figure, 'W' is the unscaled coefficient after the core transform and 'MF' is the multiplication factor. The former and the later numbers in square bracket denote the most significant bit (MSB) and the least significant bit (LSB), respectively. Therefore, a 13-bit × 14-bit-multiplier is used to realize the multiplication in (4). The product WMF[26..0] is then added with f[22..0], in front of which '0000' is extended to make the two addends have the same bit width, that is, 27 bit width. The sum of WMF[26..0] and '0000'&f[22..0] represented by T[26..0] is stored in register group 'Regs' at the rising edge of one bit clock signal 'CLK'. Finally, the 12 MSBs of T[26..0] are output as the quantized coefficient Z[11..0], while the 15 LSBs of T[26..0] is discharged to realize the right shift operation in (4).

In general, multiplication can be replaced with several additions and shift operations. However, it is useless to replace multiplication in (4) with additions and shifts directly, since most of the multiplication factors of H.264/AVC quantization are large numbers. For instance, the MF_{ij} at position (0, 0) is 13107 and 11001100110011B in binary. Thus, the multiplication should be implemented with seven additions and shifts. It means that one multiplier requires seven adders for replacement in VLSI design. In this case, few optimizations will be achieved because the

area of a multiplier is usually 8–10 times as that of an adder with the same bit width.

On the other hand, in the quantization procedure, there is 27-bit multiplication followed by more than 15-bit shift-right operation for high computational precision in Fig. 1. However, large-scale quantization error exists persistently when quantization procedure is introduced in video encoder for high compression ratio. Thus, it is unnecessary to maintain extremely high computational precision at the cost of long delay, large area and power consumptions in IC. From the analysis above, bit width of the quantization can be reduced so that few adders with small bit width can be used to replace the multiplier in VLSI design for the sake of area and power saving. Therefore, a feasible quantization scheme of modifying MF_{ij} and $qbits$ is proposed for complexity reduction.

In order to reduce the bit width, we implement the quantization unit with modified multiplication factors instead of $MF_{ij}/2^{qbits}$ in Fig. 1. For example, we use $MF'_{ij}/2^{qbits'} = 13/2^{5+\mathrm{floor(QP/6)}}$ to replace $MF_{ij}/2^{qbits} = 13107/2^{15+\mathrm{floor(QP/6)}}$, so that the integer multiplication can be realized by shift and addition operations effectively. In general, shifting N places left is the same as multiplying by 2 to the power N (written as 2^N). For example, data W multiplied by 13 can also be realized with $W \times 13 = W \times 2^3 + W \times 2^2 + W \times 2^0 = W \ll 3 + W \ll 2 + W$, where the operator '$\ll$' represents the left shift operation. It should be noted that sometimes subtraction is more efficient than addition, for instance, $W \times 7 = W \times 2^3 - W \times 2^0 = W \ll 3 - W$, rather than $W \times 7 = W \times 2^2 + W \times 2^1 + W \times 2^0 = W \ll 2 + W \ll 1 + W$.

Figure 2 gives a diagram of the improved quantization unit architecture with respect to modified $MF'_{ij}/2^{qbits'} = 13/2^{5+\mathrm{floor(QP/6)}}$. In IC design, 13-bit data $W[12..0]$ is appended with three binary '0' in LSB, denoted by $W[12..0]\&'000'$, to implement $W[12..0] \ll 3$. Additionally, we also add a bit '0' in MSB to avoid overflow errors. Other '0' bits between the MSB and W are added to keep the addends having the same bit width. In Fig. 2, only three 17-bit adders are used to implement the quantization unit. On the other hand, bit width of adders is also significantly reduced to save area and power consumption in IC. 12 MSBs of $T[16..0]$ is output as the final quantized result of the improved quantization.

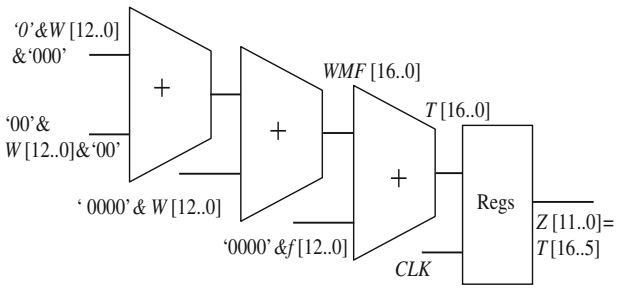

Fig. 2 Diagram of improved quantization unit architecture ($MF'_{ij}/2^{qbits'} = 13/2^{5+\mathrm{floor(QP/6)}}$)

Let $\tilde{Z}_{ij}$ be a quantized coefficient of the improved quantization, then the improved quantization procedure is mathematically written as

$$\begin{aligned}\tilde{Z}_{ij} &= \mathrm{sign}(W_{ij})\left\lfloor \frac{|W_{ij}|MF'_{ij} + f'}{2^{qbits'}} \right\rfloor \\ &= \mathrm{sign}(W_{ij})\left\lfloor \frac{|W_{ij}|(MF_{ij} + \alpha) + f + \beta}{2^{qbits}} \right\rfloor,\end{aligned} \tag{7}$$

where $\lfloor\ \rfloor$ denotes floor operation, and factors α and β are defined as

$$\alpha = \left(\frac{MF'_{ij}}{2^{qbits'}} - \frac{MF_{ij}}{2^{qbits}}\right)2^{qbits}, \tag{8}$$

$$\beta = \frac{2^{qbits}}{2^{qbits'}}f' - f. \tag{9}$$

Equation (7) is a simplification of quantization at the encoder. Since $MF_{ij}/2^{qbits}$ is replaced with $MF'_{ij}/2^{qbits'}$ in the improved quantization while the de-quantizer is unchanged, mismatch error may be caused by the modification. Compared (7) with (4), it is noticed that the mismatch is caused by α and β and this will be discussed in detail in the following subsection.

3.2 Theoretical analyses on quantization error

When quantization and de-quantization are combined together in one equation, we can obtain de-quantized coefficients W'_{ij}

$$W'_{ij} = \mathrm{sign}(W_{ij})\left\lfloor \frac{|W_{ij}|MF_{ij} + f}{2^{qbits}} \right\rfloor V_{ij}2^{\lfloor QP/6 \rfloor}. \tag{10}$$

Similarly, output coefficients $\tilde{W}'_{ij}$ after improved quantization and original de-quantization is calculated as

$$\begin{aligned}\tilde{W}'_{ij} &= \mathrm{sign}(W_{ij})\left\lfloor \frac{|W_{ij}|(MF_{ij} + \alpha) + \beta + f}{2^{qbits}} \right\rfloor V_{ij}2^{\lfloor QP/6 \rfloor} \\ &= \mathrm{sign}(W_{ij})\left\lfloor \frac{\left(|W_{ij}| + |W_{ij}|\frac{\alpha}{MF_{ij}} + \frac{\beta}{MF_{ij}}\right)MF_{ij} + f}{2^{qbits}} \right\rfloor \\ &\quad V_{ij}2^{\lfloor QP/6 \rfloor}.\end{aligned} \tag{11}$$

Let $\bar{e}_m(\alpha, \beta)$ be the average mismatch between the improved quantization and original de-quantization. $\bar{e}_m(\alpha, \beta)$is caused by α and β in the improved

quantization. Let $\bar{e}_q(QP)$ be the average quantization error of H.264/AVC, and $g(\alpha,\beta)$ be the ratio of $\bar{e}_m(\alpha, \beta)$ to $\bar{e}_q(QP)$ and written as

$$\begin{cases} g(\alpha, \beta) = \dfrac{\bar{e}_m(\alpha, \beta)}{\bar{e}_q(QP)} \\ \bar{e}_m(\alpha, \beta) = \dfrac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|\tilde{W}'_{ij} - W'_{ij}\right| \\ \bar{e}_q(QP) = \dfrac{1}{mn}\sum_{i=0}^{m-1}\sum_{j=0}^{n-1}\left|W'_{ij} - W_{ij}PF_{ij}^2 64\right| \end{cases}, \tag{12}$$

where m and n are width and height of image, respectively. As α and β increase, rate-distortion performance decreases due to the increased $\bar{e}_m(\alpha, \beta)$. However, area and power consumption in IC will also decrease as α and β increase. It is a contradiction between rate-distortion performance of video coding and area/power saving ratio of IC design.

Let T_{PCT} be up-boundary of $g(\alpha,\beta)$. If $g(\alpha,\beta)$ is smaller than T_{PCT}, $\bar{e}_m(\alpha, \beta)$ is neglectable compared with $\bar{e}_q(QP)$ and it will not affect rate-distortion performance. Let (α,β) denote the area and power saving ratio achieved by the improved quantization design. We shall find the values of α and β to maximize $APSR(\alpha,\beta)$ subjected to $g(\alpha,\beta) \le T_{PCT}$. It is mathematically expressed as

$$\begin{cases} \max(APSR(\alpha, \beta)) \\ s.t. g(\alpha, \beta) \le T_{PCT} \end{cases}. \tag{13}$$

Let p_r and p_w be the probabilities of $\tilde{W}'_{ij} = W'_{ij}$ and $\tilde{W}'_{ij} \neq W'_{ij}$, respectively, which satisfy $p_r + p_w = 1$. p_w is calculated by

$$p_w = \frac{\left||W_{ij}|\alpha + \beta\right|}{2^{qbits}}. \tag{14}$$

Because $|W_{ij}|\frac{\alpha}{MF_{ij}} + \frac{\beta}{MF_{ij}}$ in (11) is a tiny adjustment relative to $|W_{ij}|$, $W'_{ij} - W'_{ij} = V_{ij}2^{\lfloor QP/6\rfloor}$ when $\tilde{W}'_{ij} \neq W'_{ij}$. Therefore, we can get

$$\bar{e}_m(\alpha, \beta) = p_r 0 + p_w V_{ij} 2^{\lfloor QP/6\rfloor}. \tag{15}$$

According to the definition of MF_{ij} and V_{ij}, $MF_{ij}V_{ij}$ $2^{\lfloor QP/6\rfloor} = PF_{ij}^2\, 2^{qbits}64$. Therefore, by substituting (14) in (15), $\bar{e}_m(\alpha, \beta)$ can be rewritten as

$$\bar{e}_m(\alpha, \beta) = \left||W_{ij}|\frac{\alpha}{MF_{ij}} + \frac{\beta}{MF_{ij}}\right| PF_{ij}^2 64. \tag{16}$$

Similarly, the average quantization error $\bar{e}_q(QP)$ can be represented by

$$\bar{e}_q(QP) = \frac{1}{2}V_{ij}2^{\lfloor QP/6\rfloor}. \tag{17}$$

Substituting (16) and (17) in (12), we can get

$$g(\alpha, \beta) = \frac{\left||W_{ij}|\frac{\alpha}{MF_{ij}} + \frac{\beta}{MF_{ij}}\right| PF_{ij}^2 64}{\frac{1}{2}V_{ij}2^{\lfloor QP/6\rfloor}}. \tag{18}$$

In the following subsections, mismatch error caused by α and β are analyzed, respectively.

3.3 Distortion resulted from factor β

Factor β leads to an additive error to W_{ij}, which can be expressed as

$$e(\beta) = \frac{\beta}{MF_{ij}} PF_{ij}^2 64. \tag{19}$$

From (9), β is rewritten as

$$\beta = \left\lfloor \frac{2^{\phi_{ij}+\lfloor QP/6\rfloor}}{k_{IP}} \right\rfloor 2^{15-\phi_{ij}} - \left\lfloor \frac{2^{15+\lfloor QP/6\rfloor}}{k_{IP}} \right\rfloor. \tag{20}$$

where $\phi_{ij} = qbits' - \lfloor QP/6\rfloor$ and it determines the bit width of the quantization unit, k_{IP} equals to 3 for intra blocks or 6 for inter blocks. As ϕ_{ij} decreases, the bit width of the quantization decreases, which leads to larger power and area saving ratio. However, error $e(\beta)$ increases as ϕ_{ij} decreases. Figure 3 depicts the relationship of $e(\beta)$ and ϕ_{ij}. It is clear that the value of $e(\beta)$ is within the shadow in Fig. 3. As ϕ_{ij} increases, $e(\beta)$ decreases in the trend of decaying exponential, and $e(\beta) < 1$ when $\phi_{ij} \ge 3$. In general video application systems, QP is usually larger than 18 for satisfying rate-distortion performance, the average quantization error $\bar{e}_q$(QP) is greater than 40. So from Fig. 3, we can conclude that $e(\beta)$ is neglectable compared with $\bar{e}_q$(QP) when $\phi_{ij} \ge 3$. Related experimental results will be given in section 4.1.

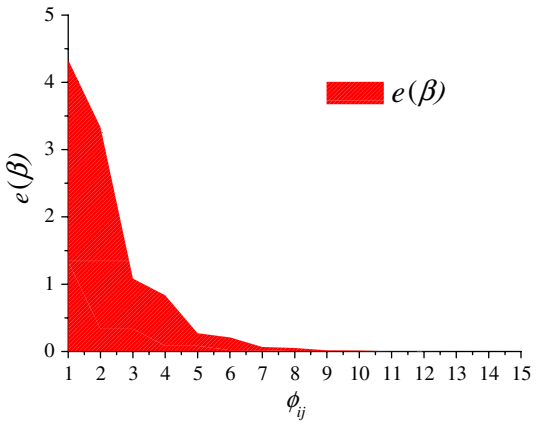

Fig. 3 Relationship of $e(\beta)$ and ϕ_{ij}

3.4 Constraints for multiplication factor modification

$MF_{ij}/2^{qbits}$ is replaced with $MF'_{ij}/2^{qbits'}$ in the proposed improved quantization design. We define

$$\frac{MF'_{ij}}{2^{qbits'}} = \frac{MF_{ij}}{2^{qbits}} \frac{1}{1+\delta_{ij}}. \tag{21}$$

Hence,

$$\delta_{ij} = \frac{MF_{ij}}{2^{qbits}} \Big/ \frac{MF'_{ij}}{2^{qbits'}} - 1, \tag{22}$$

where δ_{ij} denotes the modification percent. From (8) and (21), we can rewritten (8) as

$$\alpha = \frac{-\delta_{ij}}{1+\delta_{ij}} MF_{ij}. \tag{23}$$

From the analysis of Sect. 3.3, $e(\beta)$ is omitted in (18). So (13) can be rewritten as

$$g(\alpha,\beta) = \frac{|W_{ij}| \frac{|\alpha|}{MF_{ij}} PF_{ij}^2 64}{\frac{1}{2} V_{ij} 2^{\lfloor QP/6 \rfloor}} \le T_{PCT}. \tag{24}$$

α is a function of δ_{ij} as shown by (23), we rewrite (24) by substituting (23) in (24) in order to get the constraints for δ_{ij}, that is

$$|\delta_{ij}| \le \frac{T_{PCT} \frac{1}{2} V_{ij} 2^{\lfloor QP/6 \rfloor}}{|W_{ij}| PF_{ij}^2 64 - T_{PCT} \frac{1}{2} V_{ij} 2^{\lfloor QP/6 \rfloor}}. \tag{25}$$

In general video coding applications, QP > 18, and here we take QP = 24 as an example. T_{PCT} is set as 10% empirically. It means that $\bar{e}_q(QP)$ causes the major distortion for the compressed video when $\bar{e}_m(\alpha,\beta)$is smaller than $10\%\bar{e}_q(QP)$. Statistical experiments have been conducted on JM8.5 baseline profile platform with I-P-P mode. Video sequences 'foreman' in CIF and QCIF format have been coded for the statistical analysis. Figure 4 shows distribution map of W_{ij} for 'foreman'. It indicates the percentage of W_{ij} located in range $[-T_w, T_w]$, where x-axis is T_w and y-axis represents the percentage of W_{ij}. Curves 'qcifall' and 'cifall' denote distribution map of W_{ij} with respect to QCIF and CIF formats, respectively. We can see that 95% W_{ij}s are located within [−95, 95] for QCIF and within [−60, 60] for CIF. The value of $|W_{ij}|$ is set to be 77.5 for calculation in (25). Therefore, we can get the constraints for δ_{ij}s. The values of δ_{ij}s is better to be limited within [−2.65, 2.65%] at positions (0,0), (2,0), (0,2) and (2,2), within [−34.77, 34.77%] at positions (1,1), (1,3), (3,1) and (3,3), and within [−9.16, 9.16%] at other positions.

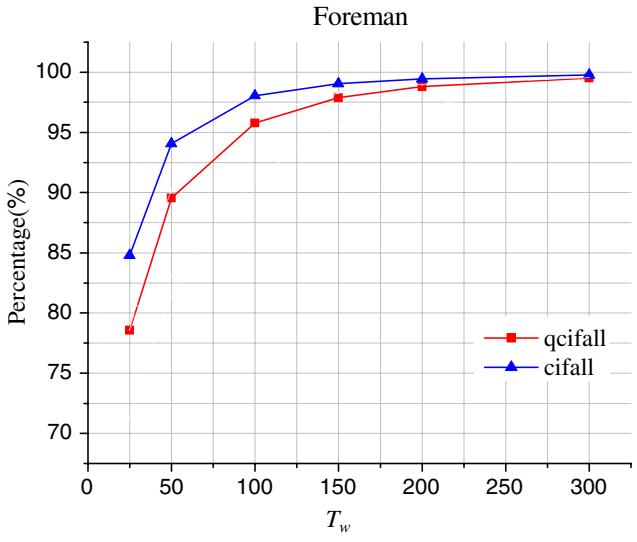

Fig. 4 W_{ij} distribution map ('foreman')

Table 3 Modified $MF'_{ij}/2^{qbits'}$

QP	(i,j)		Other positions
	(0,0), (2,0), (2,2), (0,2)	(1,1), (1,3), (3,1), (3,3)	
0	$13/2^5$	$5/2^5$	$1/2^2$
1	$3/2^3$	$9/2^6$	$7/2^5$
2	$5/2^4$	$1/2^3$	$13/2^6$
3	$9/2^5$	$7/2^6$	$3/2^4$
4	$1/2^2$	$7/2^6$	$5/2^5$
5	$7/2^5$	$3/2^5$	$9/2^6$

3.5 Modified $MF'_{ij}/2^{qbits'}$ and Corresponding δ_{ij}

The modified multiplication factors for $0 \le QP \le 5$ are specified in Table 3. There are a little deviation between the multiplication factors in Table 3 and the corresponding factors of H.264/AVC. The deviation is represented by the modification percent, δ_{ij}, as defined in (22). As shown in Table 4, δ_{ij}s are limited within [−3.02, 1.59%], [−6.39, 2.40%], and [−1.60, 5.19%] at different positions to

Table 4 δ_{ij} between $MF_{ij}/2^{qbits}$ and $MF'_{ij}/2^{qbits'}$

QP	(i,j)		Other positions (%)
	(0,0), (2,0), (2,2), (0,2) (%)	(1,1), (1,3), (3,1), (3,3) (%)	
0	−1.60	2.40	−1.54
1	−3.02	−1.13	4.49
2	−1.54	2.39	−1.53
3	1.58	1.76	5.19
4	0.00	−6.39	2.40
5	1.59	−5.83	−1.06

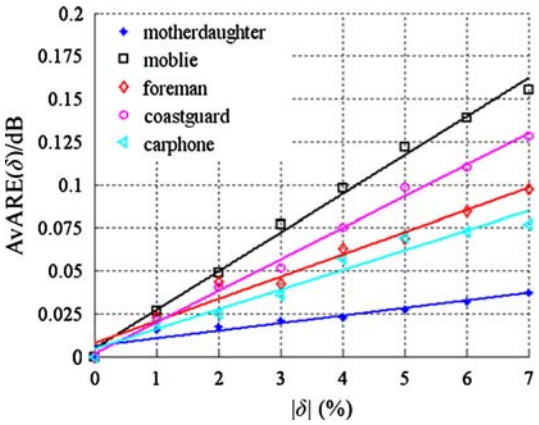

Fig. 5 Relationship between AvARE(δ) and δ

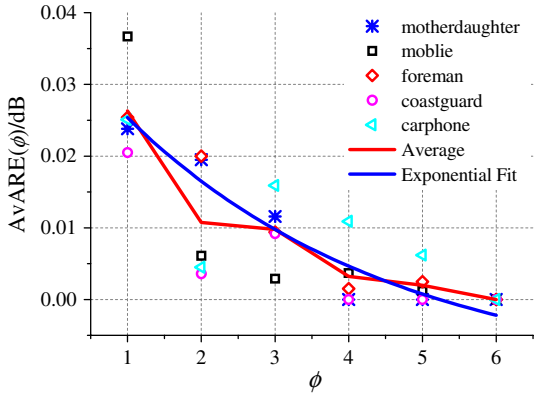

Fig. 6 Relationship between AvARE(ϕ) and ϕ

maintain high rate-distortion performance. Additionally, ϕ_{ij}s at most positions are larger than 4, so only a neglectable error will be caused by ϕ_{ij} in the improved quantization.

4 Experimental results and analyses

Rate-distortion errors caused by δ_{ij} and ϕ_{ij} (i.e., α and β) are experimentally analyzed first. Then, original and improved quantization architectures are implemented, and area and power consumption in IC are compared. Finally, rate-distortion performance between the original and improved quantization is compared.

4.1 Rate-distortion resulted from δ_{ij} and ϕ_{ij}

Relationships between δ_{ij}, ϕ_{ij} and rate-distortion performance are experimentally analyzed to prove the deductions in Sects. 3.3 and 3.4. CIF sequences 'mobile', 'coastguard', 'foreman' and 'mother and daughter', and QCIF sequences 'carphone' is coded with H.264/AVC baseline profile under different δ_{ij} and ϕ_{ij}. Test video sequences are with different image sizes and different characteristics, e.g., acute or gentle motion, smooth or complex texture. Note that δ_{ij} at different (i, j) positions are set to be the same value δ. Similarly, ϕ_{ij} at different (i, j) positions are set to be the same value ϕ in the experiments to testify the mismatch error.

Average value of the absolute rate-distortion error (AvARE) represents the rate-distortion difference between the H.264/AVC codecs with the improved and original quantization. Figure 5 shows the relationship between AvARE(δ) and $|\delta|$, where AvARE(δ) means AvARE caused by modification of δ. The points in the figure are AvARE between the improved and original codecs for different δs. The lines are linear fitted from the points. In Fig. 5, AvARE(δ) approximately linearly increases with $|\delta|$. Moreover, AvARE(δ) is also affected by the texture characteristics of video sequences. For the sequences with complex texture and acute motion, the slopes of the AvARE lines are steeper than those of the sequences with smooth texture and gentle motion. When $|\delta| < 7\%$, the AvARE(δ) is smaller than 0.175 dB.

Figure 6 illustrates the relationship between AvARE(ϕ) and ϕ. Each point is AvARE with respect to different ϕ for different video sequences. The red curve gives average value of the points with respect to the same ϕ, and the blue curve is exponentially fitted from the points. From the curves, it is noticed that AvARE(ϕ) approximates to 0 when ϕ is larger than 4. On the other hand, AvARE(ϕ) is almost independent to video contents and is smaller than 0.025 dB. In Table 4, most ϕ_{ij}s are larger than 3, therefore, only a neglectable degradation is caused by ϕ_{ij} in the improved quantization.

4.2 Synthesis results and comparisons

Figure 2 gives the improved quantization unit architecture. Similarly, other quantization units can be implemented according to the $MF'_{ij}/2^{qbits'}$ in Table 3. The quantization units are implemented with Verilog HDL at register transfer level. Synthesis and verification of the quantization circuits has been conducted on Design Compiler of Synopsys Corp. to analyze the power and area consumption. Additionally, functional verification has been performed on Modelsim platform of Mentor Graphics Corp. From the simulation result, it is seen that under the same cell CMOS technology, up to 75.2% area and 76.3% dynamic power on average are saved in a quantization unit compared with original H.264/AVC scheme, as given in Tables 5 and 6. On the other hand, pipeline speed can also be enhanced by further pipelining improvement owning to the same arithmetic operation as the core part of the transform.

Table 5 Comparison on consumed area (implemented with adders)

QP%6	Original quantization			Improved quantization			Saving ratio on average
	(0,0),(2,0), (2,2),(0,2)	(1,1),(1,3), (3,1),(3,3)	Other positions	(0,0),(2,0), (2,2),(0,2)	(1,1),(1,3), (3,1),(3,3)	Other positions	
0	12,668	13,912	9,749	3,792	2,944	1,360	−75.2%
1	11,427	10,888	9,750	2,545	3,020	2,924*	
2	10,548	9,813	11,666	2,621	1,434	3,985	
3	10,103	13,596	9,710	2,698	3,117*	2,698	
4	1,431	12,804	13,064	1,267	3,117*	2,774	
5	10,438	12,285	12,944	2,774*	2,850	2,814	

* Indicate subtraction operation

Table 6 Comparison on dynamic power consumption (implemented with adders) unit:mW

QP%6	Original quantization			Improved quantization			Saving ratio on average
	(0,0),(2,0), (2,2),(0,2)	(1,1),(1,3), (3,1),(3,3)	Other positions	(0,0),(2,0), (2,2),(0,2)	(1,1),(1,3), (3,1),(3,3)	Other positions	
0	21.24	22.82	14.94	6.22	4.71	1.95	−76.3%
1	18.87	18.81	16.46	3.98	4.83	4.49[a]	
2	17.73	15.8	19.74	4.19	2.14	6.54	
3	16.98	20.37	16.47	4.31	4.89[a]	4.23	
4	2.15	21.47	21.39	1.85	4.89[a]	4.45	
5	17.30	21.04	20.71	4.35[a]	4.50	4.48	

Total area in the table is the sum of total cell area and net interconnect area

13-bit width for input at (0,0),(2,0),(2,2),(0,2) positions

15-bit width for input at (1,1),(1,3),(3,1),(3,3) positions

14-bit width for input at other positions

[a] Subtraction operation

Table 7 BDBR compared with original H.264/AVC unit: %

QP	CIF sequences					QCIF sequences		
	Foreman	Mother and daughter	Coastguard	Sign Irene	Mobile	Carphone	Missa	Salesman
18–21	0.3199	0.2871	0.9650	0.2358	1.4630	0.6706	0.5277	0.8405
21–24	0.5260	0.0937	0.7036	0.5118	1.5290	0.6499	0.4232	0.8403
24–27	−0.0409	−0.0130	0.0021	0.0963	0.3188	0.5536	0.1604	0.3090
27–30	0.4119	0.2556	0.1209	0.3356	0.9291	0.2709	0.1652	0.5033
30–33	−0.2669	−0.0547	−0.1461	0.1436	0.1154	−0.1966	−0.0168	−0.0114
33–36	0.3255	−0.1246	0.0061	−0.0369	0.5451	0.2193	1.1268	0.3160
36–39	−0.2538	−0.5129	−0.2541	−0.0255	−0.0834	−0.2662	0.2091	0.7609
39–42	0.5275	0.1869	−0.3265	0.2141	0.1175	−0.1966	0.2011	0.2672
42–45	0.0876	−0.6754	0.1298	−0.1313	−0.0568	−0.4063	−0.5859	0.2329
45–48	0.4089	0.3260	−0.1230	−0.0283	0.4305	0.7552	0.0108	0.0015
48–51	−0.3823	−0.1158	0.6781	−0.6216	0.0709	0.5157	−0.1220	−1.3552
Average	0.1483	−0.0260	0.2018	0.0842	0.5429	0.2472	0.1942	0.3531

Table 8 BDPSNR compared with original H.264/AVC unit: dB

QP	CIF					QCIF		
	Foreman	Mother and daughter	Coastguard	Sign Irene	Mobile	Carphone	Missa	Salesman
18–21	−0.012203	−0.009253	−0.061743	−0.007701	−0.082473	−0.032359	−0.014649	−0.051788
21–24	−0.023353	−0.004545	−0.045328	−0.022803	−0.088941	−0.035222	−0.015278	−0.051516
24–27	0.001756	0.000577	0.000945	−0.004841	−0.017895	−0.029864	−0.007779	−0.020472
27–30	−0.018547	−0.012242	−0.006639	−0.015734	−0.048747	−0.014243	−0.008266	−0.029559
30–33	0.012742	0.002889	0.007464	−0.005442	−0.004726	0.012381	0.002673	0.001284
33–36	−0.015964	0.004431	−0.000581	0.001241	−0.022779	−0.012559	−0.060903	−0.011965
36–39	0.013160	0.023121	0.007583	0.001983	0.004776	0.016923	−0.011849	−0.036883
39–42	−0.031148	−0.009234	0.008928	−0.010854	−0.006875	0.011927	−0.026969	−0.011698
42–45	−0.007427	0.029803	−0.003357	0.005674	0.006733	0.024491	0.064865	−0.011922
45–48	−0.025807	−0.010467	0.003242	0.002420	−0.028244	−0.046320	−0.003132	−0.009027
48–51	0.023070	0.005436	−0.020604	0.026845	−0.004148	−0.028872	−0.000461	0.051062
Average	−0.007268	0.001547	−0.006870	−0.003173	−0.025286	−0.013688	−0.009032	−0.017201

4.3 Rate-distortion performance comparisons

In order to evaluate the rate-distortion performance of the proposed quantization scheme, video sequences with various textures, including 'foreman', 'mother and daughter', 'coastguard', 'Sign Irene', 'mobile', 'carphone', 'missa', and 'salesman' have been coded on JM8.5 baseline profile platform with I–P-P mode (insert an I-frame every 10 P-frames) at 30 fps. Tables 7 and 8 give Bjontegaard delta PSNR (BDPSNR) and Bjontegaard delta bit rate (BDBR) [19] between the two schemes with respect to the above eight typical sequences, where QP is from 18 to 51 with step 1. It is clear that BDPSNR and BDBR are very slight, which means that the improved quantization scheme is approximately the same as the original quantization scheme of H.264/AVC in rate-distortion performance.

5 Conclusion

Core part of integer transform adopted by H.264/AVC standard is a simplex multiply free orthogonal transform. It only requires integer additions and shifts. However, multiplication is integrated into the quantizer. Multiplications not only increase complexity but also cause long delay. The multiplier is also a bottleneck of the pipeline of integer transform and quantization architecture. In this paper, multiplication factors of H.264/AVC quantizer are modified. The modifications of multiplication factors are limited within the range of [−3.02, 1.59%], [−6.39, 2.40%] and [−1.60, 5.19%] at different positions. An improved quantization is proposed by substituting large bit-width multiplier by some small bit-width adders in IC design. About 75.2% area and 76.3% dynamic power consumption in IC are saved on average in each unit compared with original H.264/AVC quantization. On the other hand, experimental results show that rate-distortion performance of the improved quantization is almost the same as that of the original H.264/AVC quantization.

Acknowledgments This work was supported by the Natural Science Foundation of China (grant 60672073), the Program for New Century Excellent Talents in University (NCET–06-0537), Natural Science Foundation of Ningbo (grant 2007A610037).

References

1. Wiegand, T., Sullivan, G.J., Bjontegaard, G., Luthra, A.: Overview of the H264/AVC video coding standard. IEEE Trans. Circuits Syst. Video Technol **13**(7), 560–576 (2003). doi:10.1109/TCSVT.2003.815165
2. Malvar, H.S., Hallapuro, A., Karczewicz, M., Kerofsky, L.: Low-complexity transform and quantization in H.264/AVC. IEEE Trans. Circuits Syst. Video Technol **13**(7), 598–602 (2003). doi:10.1109/TCSVT.2003.814964
3. Sullivan, G.J., Wiegand, T.: Video compression—from concepts to the H.264/AVC standard. Proc. IEEE **93**(1), 18–31 (2005). doi:10.1109/JPROC.2004.839617
4. Huang, Y.-W., Hsieh, B.-Y., Chen, T.-C., Chen, L.-G.: Analysis, fast algorithm, and VLSI architecture design for H.264/AVC intra frame coder. IEEE Trans. Circuits Syst. Video Technol **15**(3), 378–401 (2005). doi:10.1109/TCSVT.2004.842620
5. Jing, X., Chau, L.-P.: An efficient inter mode decision approach for H.264 video coding. IEEE Int. Conf. Multimed. Expo **2**, 1111–1114 (2004)
6. Chen, T.-C., Chen, Y.-H., Tsai, S.-F., Chien, S.-Y., Chen, L.-G.: Fast algorithm and architecture design of low-power integer motion estimation for H.264/AVC. IEEE Trans. Circuits Syst. Video Technol **17**(5), 568–577 (2007). doi:10.1109/TCSVT.2007.894044

7. Su, Y.P., Sun, M.-T.: Fast multiple reference frame motion estimation for H.264/AVC. IEEE Trans. Circuits Syst. Video Technol **16**(3), 447–452 (2006). doi:10.1109/TCSVT.2006.869970
8. Huang, Y.-W., Hsieh, B.-Y., Chien, S.-Y., Ma, S.-Y., Chen, L.-G.: Analysis and reduction of reference frames for motion estimation in MPEG-4/AVC/JVT/H.264. IEEE Trans. Circuits Syst. Video Technol **16**(4), 507–522 (2006). doi:10.1109/TCSVT.2006.872783
9. Cheung, G., Tan, W., Chan, C.: Reference frame optimization for multiple-path video streaming with complexity scaling. IEEE Trans. Circuits Syst. Video Technol **17**(6), 649–662 (2007). doi:10.1109/TCSVT.2007.896620
10. Kim, C., Jay Kuo, C.-C.: Feature-based intra-/intercoding mode selection for H.264/AVC. IEEE Trans. Circuits Syst. Video Technol **17**(4), 441–453 (2007). doi:10.1109/TCSVT.2006.888829
11. Lin, Y.-C., Fink, T., Bellers, E.: Fast mode decision for H.264 based on rate-distortion cost estimation. IEEE Int. Conf. Acoust Speech Signal Process **I**, 1137–1140 (2007)
12. Wang, H., Kwong, S., Kok, C.-W.: An efficient mode decision algorithm for H.264/AVC encoding optimization. IEEE Trans. Multimed **9**(4), 882–888 (2007). doi:10.1109/TMM.2007.893345
13. Wei, Z., Li, H., Ngan, K.N.: An efficient intra mode selection algorithm for H.264 based on fast edge classification. In: IEEE International Symposium on Circuits and Systems (ISCAS 2007), 27–30 May 2007, pp 3630–3633 (2007). doi:10.1109/ISCAS.2007.378539
14. Li, X., Oertel, N., Kaup, A.: Adaptive Lagrange multiplier selection for intra-frame video coding. IEEE Int. Symp. Circuits Syst. 3643–3646 (2007)
15. Xi, Y.-L., Hao, C.-Y., Fan, Y.-Y., Hu, H.-Q.: A fast block-matching algorithm based on adaptive search area and its VLSI architecture for H.264/AVC. Signal Process. Image Commun **21**(8), 626–646 (2006). doi:10.1016/j.image.2006.05.001
16. Wang, H., Kwong, S.: Hybrid model to detect zero quantized DCT coefficients in H.264. IEEE Trans. Multimed **9**(4), 728–735 (2007). doi:10.1109/TMM.2007.893336
17. Chen, J.-L., Lin, Y.-K., Chang, T.-S.: A low cost context adaptive arithmetic coder for H.264/MPEG-4 AVC video coding. IEEE Int. Conf. Acoust. Speech Signal Process **II**, 105–108 (2007)
18. Fan, C.-P.: Fast 2-dimensional 4 × 4 forward integer transform implementation for H.264/AVC. IEEE Trans. Circuits Syst. II Express Briefs **53**(3), 174–177 (2006). doi:10.1109/TCSII.2005.858748
19. Bjontegaard, G.: Calculation of average PSNR differences between RD-curves. ITU-T VCEG, Doc. VCEG-M33, Apr 2001

Author Biographies

Yun Zhang received his B.S. and M.S. degrees in information and electronic engineering from Faculty of Information Science and Engineering, NingBo University, China, in 2004 and 2007. He is now a Ph.D. candidate at Institute of Computing Technology, Chinese Academy of Sciences of China. His research interests mainly include digital video compression and communications, SoC design and embedded system for consumer electronics.

Gangyi Jiang received his M.S. degree from Hangzhou University, in 1992, and received his Ph.D. degree from Ajou University, Korea, in 2000. He is now a professor at Faculty of Information Science and Engineering, Ningbo University, China. His research interest mainly includes digital video compression and communications, multi-view video coding and image processing.

Mei Yu received her M.S. degree from Hangzhou Institute of Electronics Engineering, China, in 1993, and Ph.D. degree form Ajou University, Korea, in 2000. She is now a professor at Faculty of Information Science and Engineering, Ningbo University, China. Her research interests include image/video coding and video perception.