Proceedings of the 12th IASTED International Conference
Signal and Image Processing (SIP 2010)
August 23-25, 2010 Maui, Hawaii, USA

A VC-1 TO H.264/AVC INTRA TRANSCODING USING ENCODING INFORMATION TO REDUCE RE-QUANTIZATION NOISE

Takeshi Yoshitome, Yoshiyuki Nakajima, and Kazuto Kamikura
NTT Cyber Space Laboratories,
NTT Corporation,
Yokosuka-shi, 239-0847, JAPAN

Shoji Makino and Nobuhiko Kitawaki
Graduate School of Systems
and Information Engineering
University of Tsukuba
Tsukuba, Japan, 305-8573, JAPAN

ABSTRACT

We propose a VC-1 to H.264/AVC intra transcoding method. This method uses the encoding information from a VC-1 stream and keeps as many DCT coefficients of the original VC-1 bitstream as possible. Experimental results show that the proposed method improves PSNR by about 0.44-1.34 dB compared with a conventional method.

KEY WORDS

Transcoding, VC-1, H.264, Quantization noise, Encoding information

1 Introduction

Recently, it has become important to re-compress video contents to decrease bitstream size by using a higher performance encoding tool such as H.264[2]. In this decade, a number of video transcoding studies have been reported [5, 6, 7] that aim to reduce the H.264 encoding time, which is several times larger than that of old compression tools, such as MPEG-2[1] and VC-1[3]. For example, [8, 9] have reported that the use of an MPEG-2 motion vector and compression mode information helps to reduce the processing time of H.264 re-encoding. A direct mathematical matrix conversion from the 8x8 real DCT of MPEG-2 to the 4x4 integer DCT of H.264 has also been reported[10]. However, the image quality obtained with these approaches becomes lower than that of conventional approaches using a directly connected MPEG-2 or VC-1 decoder and H.264 encoder.

We have focused on the fact that all decisions of the first encoding procedure are repeated during the transcoding procedure to reduce re-quantization noise[4], where "all decisions" means decisions on picture type, macroblock type, motion vectors, DCT type, and so on. This noise suppression mechanism is effective when the compression standards of the first and second encoding are identical, such as for MPEG-2/MPEG-2 or H.264/H.264 conversion. We have tried to expand this approach to apply it to MPEG-2/H.264 transcoding for progressive bitstreams[11, 12], because the image quality obtained with this approach may become higher than that of conventional method mentioned above. This paper reports our attempt to apply this noise suppression mechanism to VC-1 intra bit-streams. First, we discuss the noise suppression mechanism for progressive bit-streams. Next, we show the differences between the VC-1 and H.264/AVC standards, analyze how they affect the noise reduction, and propose a VC-1 to H.264/AVC transcoding method. Finally, we describe simulation results we obtained and compare the transcoding noises generated by our method and a conventional method.

2 Noise reduction mechanism using first encoding information

Data flow examples of transcoding with and without first VC-1 encoding information are shown in Fig. 1. The data flow of Enc1→Dec1→Enc2→Dec2 is with encoding information, and that of Enc1→Dec1→Enc3→Dec3 is without it. For simplicity, the macro-block (MB) size is 2x2. In the first VC-1 encoding, an input MB that includes the pixel values (A, B, C, and D) is processed by intra prediction and orthogonal transformation. Then, the transformed DCT coefficients (43, 31, 24, and 13) are quantized by a quantization step Δ=10. The quantized DCT coefficients (4, 3, 2, and 1) are converted to bitstream#1 by variable length coding (VLC). The quantization from (43, 31, 24, and 13) to (4, 3, 2, and 1) adds quantization noise so that the decoded MB pixels (a, b, c, and d) differ from the input MB pixels (A, B, C, and D). Until this point, transcoding both with and without encoding information does the same work.

In the transcoding without encoding information, Enc3 searches the best motion vector to minimize motion estimation error (p', q', r', and s'). If the motion search performance of Enc3 is higher than that of Enc1, Enc3 can reduce its motion estimation error (p', q', r', and s') compared with Enc1's motion estimation error (A', B', C', and D'). In the same way, if the orthogonal transformation performance of Enc3 is higher than that of Enc1, Enc3 can reduce the size of bitstream#3 compared with that of bitstream#1. However, the re-quantization noise occurs when DCT coefficients (34, 33, 14, and 4) are quantized to (3, 3, 1, and 0). In short, Enc3 adds re-quantization noise even if the bitstream becomes smaller. On the other hand, in the transcoding with encoding information, every type of en-

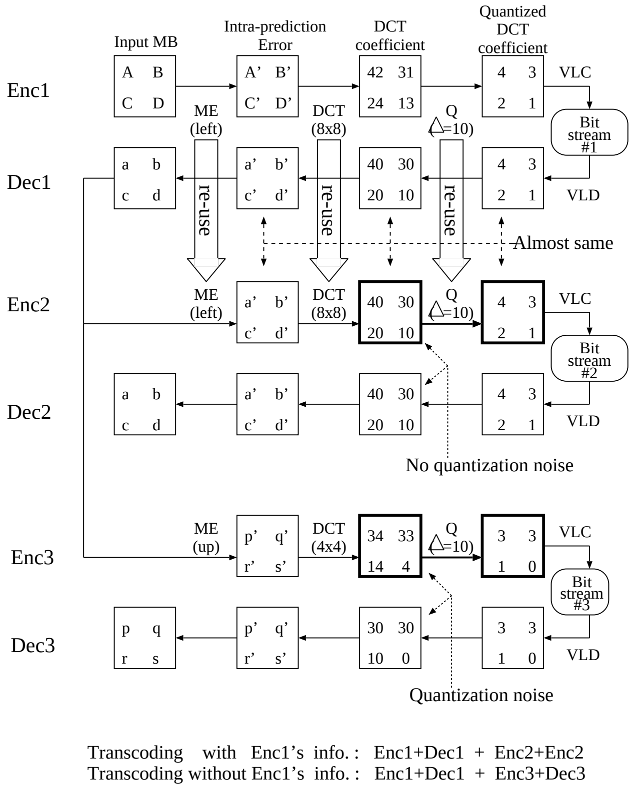


Figure 1. Transcoding with and without first VC-1 encoding information

coding information, such as MB type, motion vector, DCT mode, and quantization step, is re-used in Enc2. This re-use of encoding information does not produce any significant difference between motion compensation errors of the first encoding and those of the second encoding, nor does it produce any notable difference between the DCT coefficients of the two encodings. This means that the re-quantization noise added in the second encoding phase is almost zero. As a result, the same motion estimation error and the same DCT quantization coefficients are obtained. Unfortunately, there is no bit reduction because the sizes of bitstream#1 and #2 are almost the same when these encodings make use of the same compression tool. But transcoding using different compression tools, such as VC-1 to H.264 transcoding, may reduce the bitstream size because the performance of H.264's entropy coding (CABAC) is higher than that of VC-1's entropy coding (VLC).

3 Differences between VC-1 and H.264

As depicted in Fig. 1, there is a close resemblance between the decoded images of the first and second decoders when the compression standards of the first and second encodings are exactly the same. To re-enact the effect of this noise reduction mechanism, we must control H.264 behavior to imitate VC-1 behavior as much as possible when the

Table 1. Major differences between VC-1 and H.264 for intra encoding

	VC-1	H.264	H.264 re-enact
Intra prediction block size	8x8	4x4 8x8	OK
Direction of intra prediction	left up no-pred.	8 directions without no-pred.	?
Orthogonal transformation	unique	integer DCT	?
Size of orthogonal transformation	8x8	8x8(*1) 4x4	OK
Specification of quantization matrix	no	yes(*1)	OK
Interval of quantization step	linear	logarithmic	?
Deadzone in quantization	yes/no (*2)	no	?

(*1): only high-profile
(*2): depends on encoder setting

first encoder is VC-1 and the second encoder is H.264. The two encoders are similar but are not upper compatible. The differences between them are listed in Table 1. The "ok" means that H.264 can recreate the same VC-1 function in the table. The "?" means that H.264 can process the function only in a similar but not identical way that VC-1 does. The following four functions are not exactly the same between VC-1 and H.264, and are not investigated: (1) Intra prediction, (2) Available quantization step, (3) Orthogonal transformation, (4) Quantization with and without deadzone. In the next section, we describe these differences in detail and show a way to overcome them.

3.1 Intra prediction

There are three intra prediction modes, including the no prediction mode, in the VC-1 specification. Adjoining pixels in the upper or left blocks are referred to in this prediction. H.264 has eight prediction angles to increase compression efficiency. The two prediction modes (mode=up and mode=left) in VC-1 can be converted to the mode=0 and mode=1 in H.264, as shown in Fig.2. But it is impossible to exactly equalize the VC-1 and H.264 intra blocks when VC-1 selects the no prediction mode because H.264 does not have this mode. The best way to imitate this mode is to use the DC prediction mode in H.264. If this is used, the orthogonal transformation will change only the DC coefficient. None of the coefficients except the DC one are affected by the difference between the VC-1 and H.264 predictions. Intra prediction mode mapping from VC-1 to H.264 is shown in Table2.

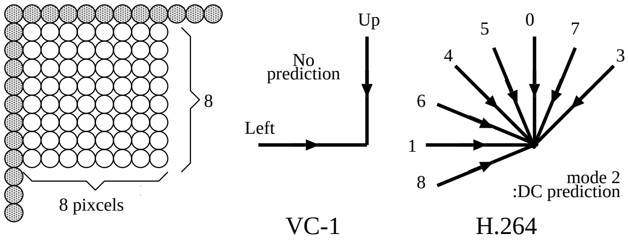


Figure 2. Intra_8x8 prediction of VC-1 and H.264

Table 2. Intra prediction mode mapping from VC-1 to H.264

VC-1 prediction mode	H.264 prediction mode
Up	0
Left	1
No prediction	2 (DC)

3.2 Available quantization step size

The derivation manner from the quantization code to the quantization step in VC-1 is similar to that of MPEG-2 when the parameter q_scale_type is equal to zero in MPEG-2. The quantization step equals the quantization code in MPEG-2. The quantization step equals twice the quantization code in VC-1. Additionally, VC-1 can also use the quantization steps, (3, 5, 7, .. , 15 , 17) using its 1bit parameter, $HALFQP$, which increases the quantization step by one. Thus, there are 39 available quantization steps in VC-1. In contrast, there are 51 available quantization steps in H.264. The minimum value is 0.625 and the maximum value is 224. The ratio of the two proceeding quantization steps is an almost constant value in H.264.

Available quantization steps in VC-1 and in H.264 are shown in Table3. Δ value of "O" means there is availability. There are 14 cases in which VC-1 can specify Δ and H.264 can not. In these cases, the nearest quantization step of H.264 is used in place of that of VC-1. The ratio difference between the VC-1 and H.264 quantization steps is from 3.2% to 8.3%. The negative effects caused by this difference will be described in detail in Section 5.2. The amount of this difference is within the acceptable level.

Table 3. Available Δ in VC-1 and in H.264

Δ	H.264	VC-1	Δ	H.264	VC-1
0.625	O	-	20	O	O
0.6875	O	-	22	O	O
0.8125	O	-	24	-	O
0.875	O	-	26	O	O
1	O	-	28	O	O
1.125	O	-	30	-	O
1.25	O	-	32	O	O
1.375	O	-	34	-	O
1.625	O	-	36	O	O
1.75	O	-	38	-	O
2	O	O	40	O	O
2.25	O	-	42	-	O
2.5	O	-	44	O	O
2.75	O	-	46	-	O
3	-	O	48	-	O
3.25	O	-	50	-	O
3.5	O	-	52	O	O
4	O	O	54	-	O
4.5	O	-	56	O	O
5	O	O	58	-	O
5.5	O	-	60	-	O
6	-	O	62	-	O
6.5	O	-	64	O	-
7	O	O	72	O	-
8	O	O	80	O	-
9	O	O	88	O	-
10	O	O	104	O	-
11	O	O	112	O	-
12	-	O	128	O	-
13	O	O	144	O	-
14	O	O	160	O	-
15	-	O	176	O	-
16	O	O	208	O	-
17	-	O	224	O	-
18	O	O	-	-	-

3.3 Orthogonal transformation

In VC-1 intra compression, the orthogonal transformation which has 8x8 size is used. The transformation, and the quantization and de-quantization, are given in the following Eqs. (1) and (2) :

$$Y = (V \cdot X \cdot V^t) \circ N \tag{1}$$

$$Y_d = Dequant_{vc1}\ (\ Quant_{vc1}\ (Y)) \tag{2}$$

The inverse orthogonal transformation in VC-1 is described in Eqs. (3) and (4):

$$E = (Y_d \cdot V + 4) >> 3 \tag{3}$$

$$\hat{X} = V^t \cdot E + C_8 \cdot 1_8 \tag{4}$$

where X is the input image, Y is the transformed coefficient, N is the normalization matrix, the operator "V^t " represents a transposition of matrix V, the operator "$>>$" represents an arithmetic right shift performed by entry-wiseon a matrix, the operator "$\cdot$" represents matrix multiplication, the operator "$\circ$" is a component-wise multiplication, Y_d is the de-quantized transformed coefficient that was quantized, $\hat{X}$ is the decoded image, V is the transfer matrix, $Dequant_{vc1}()$ and $Quant_{vc1}()$ are the quantization and de-quantization functions of VC-1, V^t is the transposed matrix of V, E is an intermediate matrix, $C_8 = (0\ 0\ 0\ 0\ 1\ 1\ 1\ 1)^t$, and 1_8 is an eight length row vector of ones. The (i,j) component of V is described in Eq.5 as follows:

$$V = \begin{pmatrix} 12 & 12 & 12 & 12 & 12 & 12 & 12 & 12 \\ 16 & 15 & 9 & 4 & -4 & -9 & -15 & -16 \\ 16 & 6 & -6 & -16 & -16 & -6 & 7 & 16 \\ 15 & -4 & -16 & -9 & 9 & 16 & 4 & -15 \\ 12 & -12 & -12 & 12 & 12 & -12 & -12 & 12 \\ 9 & -16 & 4 & 15 & -15 & -4 & 16 & -9 \\ 6 & -16 & 16 & -6 & -6 & 16 & -16 & 6 \\ 4 & -9 & 15 & -16 & 16 & -15 & 9 & -4 \end{pmatrix} \tag{5}$$

The (i,j) component of V is described as follows:

$$N_{ij} = c_j c_i^t$$

$$where\quad c = \left(\tfrac{8}{288}\quad \tfrac{8}{289}\quad \tfrac{8}{292}\quad \tfrac{8}{289}\quad \tfrac{8}{288}\quad \tfrac{8}{289}\quad \tfrac{8}{292} \right)$$

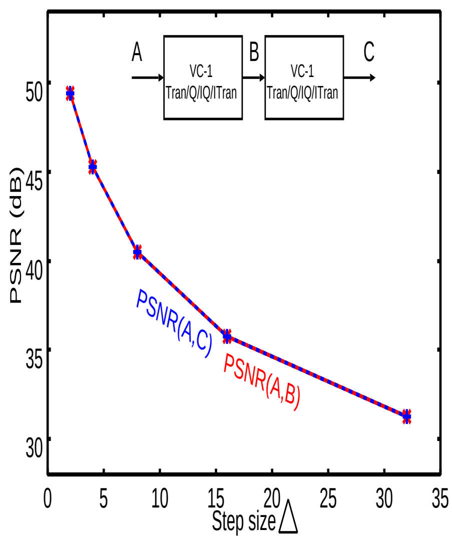


Figure 3. PSNR of VG 1/VG 1 re-encoding

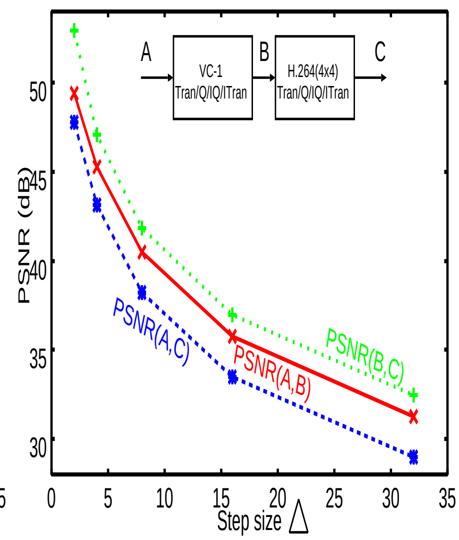


Figure 4. PSNR of VC-1/H.264 (H=4x4) re-encoding

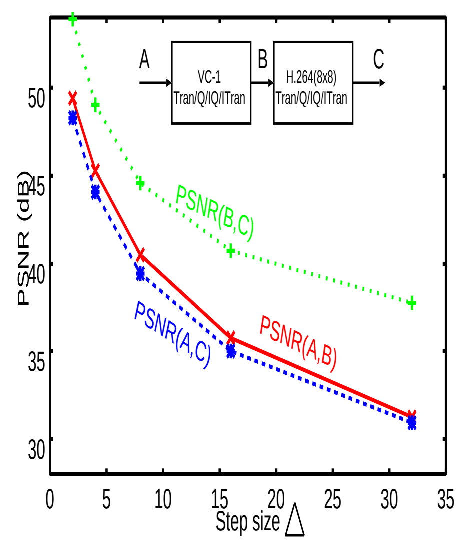


Figure 5. PSNR of VC-1/H.264 (H=8x8) re-encoding

In H.264, if we assume that Z is the integer DCT output and H is the integer DCT matrix, $Dequant_{H264}()$ and $Quant_{H264}()$ are the quantization and de-quantization functions of H.264, and Z_d is the de-quantized integer DCT coefficient that was quantized. Then the integer DCT, the quantization and de-quantization, and the inverse integer DCT are described in Eqs. (6), (7), (8) below:

$$Z = HXH^t \tag{6}$$

$$Z_d = Dequant_{H264}\left(\, Quant_{H264}\,(Z)\right) \tag{7}$$

$$\hat{X} = H^t Z_d H \tag{8}$$

H with N=4 is shown in Eq. (9).

$$H = \begin{pmatrix} \frac{1}{2} & \frac{1}{2} & \frac{1}{2} & \frac{1}{2} \\ \sqrt{\frac{2}{5}} & \sqrt{\frac{1}{10}} & -\sqrt{\frac{1}{10}} & -\sqrt{\frac{2}{5}} \\ \frac{1}{2} & -\frac{1}{2} & -\frac{1}{2} & \frac{1}{2} \\ \sqrt{\frac{1}{10}} & -\sqrt{\frac{2}{5}} & \sqrt{\frac{2}{5}} & -\sqrt{\frac{1}{10}} \end{pmatrix} \tag{9}$$

H with N=8 is shown in Eq.(10).

$$H = \begin{pmatrix} a & a & a & a & a & a & a & a \\ \frac{3c}{2} & \frac{5c}{4} & \frac{3c}{4} & \frac{3c}{8} & -\frac{3c}{8} & \frac{3c}{4} & -\frac{5c}{4} & -\frac{3c}{2} \\ b & \frac{b}{2} & -\frac{b}{2} & -b & -b & -\frac{b}{2} & \frac{b}{2} & b \\ \frac{5c}{4} & -\frac{3c}{8} & -\frac{3c}{2} & -\frac{3c}{4} & \frac{3c}{4} & -\frac{3c}{2} & \frac{3c}{8} & -\frac{5c}{4} \\ a & -a & -a & a & a & -a & -a & a \\ \frac{3c}{4} & -\frac{3c}{2} & \frac{3c}{8} & \frac{5c}{4} & -\frac{5c}{4} & -\frac{3c}{8} & \frac{3c}{2} & -\frac{3c}{4} \\ \frac{b}{2} & -b & b & -\frac{b}{2} & -\frac{b}{2} & b & -b & \frac{b}{2} \\ \frac{3c}{8} & -\frac{3c}{4} & \frac{5c}{4} & \frac{3c}{2} & -\frac{3c}{2} & -\frac{5c}{4} & \frac{3c}{4} & -\frac{3c}{8} \end{pmatrix}$$

$$where \quad a = \sqrt{\frac{1}{8}}, \;\; b = \sqrt{\frac{1}{5}}, \;\; c = \sqrt{\frac{32}{289}} \tag{10}$$

To compare the 64 components of Y and Z, an example image is processed by two consecutive stages using mathematical operations. In the first stage, the image is processed by the VC-1's transformation , the quantization and de-quantization, and the VC-1's inverse transformation in Eq.(1) - Eq(4). In the second stage, the output image of the first stage is processed by following the different functions.

1. VC1-Trans/quantize/de-quantize/VC1-InverseTrans (Eq.1-Eq.4, PSNR is shown in Fig. 3)

2. 4x4int-DCT/quantize/de-quantize/4x4int-IDCT (Eq.6-8 with N=4 . PSNR is shown in Fig. 4)

3. 8x8int-DCT/quantize/de-quantize/8x8int-IDCT (Eq.6-8 with N=8 . PSNR is shown in Fig. 5)

In the second stage, the quantization and de-quantization use the same quantization step used in the first stage. The quantization and de-quantization without dead-zone are used because only the difference between the orthogonal transformations of VC-1 and H.264 reflects the PSNR results. From Fig. 3 to Fig. 5, A means the input image, B means the decoding image of the first encoding, C means the decoding image of the second encoding, and PSNR (A, B) means the PSNR value between A and B. Figure 3 shows that PSNR (A, B) equals PSNR (A, C). This indicates that the repetition of orthogonal transformations of VC-1 does not decrease the PSNR, as mentioned in Section 2. We omitted PSNR (B, C) from Fig. 3 because the re-quantization noise showed the lowest value and PSNR (B, C) rose to over 100 dB. When the second encoding is the 4x4 integer DCT of H.264 (Eq.(6) - Eq.(8), N=4), the PSNR (A, B) is slightly higher than the PSNR (B, C) shown in Fig.4, because the amount of quantization noise of the second encoding is slightly lower than that of the first encoding. This indicates there is a slight similarity between the VC-1's 8x8 transform matrix and the H.264's 4x4 transform matrix. The final PSNR (A, C) is lower than PSNR (A, B) by nearly 3 dB because the similarity is not large. On the other hand, when the second encoding is the 8x8 integer DCT of H.264 (Eq. (6) - Eq. (8),N=8), PSNR (A, B) < PSNR (B, C) and the difference becomes larger as Δ becomes larger, as shown in Fig. 3. This indicates there is a large similarity between the VC-1's 8x8 transform matrix and the H.264's 8x8 transform matrix. Figure. 5 also shows that a large Δ decreases the PSNR reduction caused by the difference between the VC-1's 8x8 transformation and the H.264's 8x8 transformation. The differences between PSNR (A, B) and PSNR (A, C) are (1.2, 1.2, 1.0, 0.8, and 0.3) when Δ = (2, 4, 8, 16, and 32).

3.4 Quantization

For intra MB quantization, VC-1 can use the two different quantization modes shown in Fig. 6. Figure 6 (a) shows a non-uniform quantization and Fig. 6 (b) shows a uniform quantization. X_i means quanization output and b_i means quanization bin. If the input is in b_i, the quantized output equals X_i. In a uniform quantization, the size of quanization bin is constant. In a non-uniform quantization, the size of b_0 is larger than that of the others. The non-uniform quantization decreases the bit-stream size because it increases the number of X_0. The uniform quantization enables accurate quantization, so it is suitable for high bitrate encoding. H.264 uses only uniform quantization that has no deadzone. The noise reduction effect, mentioned in Section.2, decreases when VC-1 selects the non-uniform quantization. Although the VC-1 specification allows the use of these two quantization modes in the same intra picture, most VC-1 bit-streams are quantized using the uniform quantization because the intra picture needs to be more accurate in terms of encoding quality than the inter picture. The negative effects caused by this uniformity difference between the first VC-1 encoding and the second H.264 re-encoding will be described in detail in Section 5. The amount of this difference is within the acceptable level.

4 Proposed Method

The proposed VC-1 to H.264 conversion rules are listed in Table 4. The H.264 high profile enables the use of an 8x8 integer DCT that has the same orthogonal transformation size as that of VC-1. All coefficients of the H.264 quantization matrix are set to sixteen to equalize the flat VC-1's quantization characteristics. The proposed method chooses the H.264's quantization step whose value is nearest to that of the VC-1's quantization step. The intra prediction mode is set to the DC mode to imitate the VC-1 standard. The H.264 standard can select one of the intra MB modes from among I4x4, I8x8, and I16x16 and each DCT matrix size is either 4x4, 8x8, or 4x4. The proposed method uses only the I8x8 mode because its DCT matrix size is the same as that of VC-1.

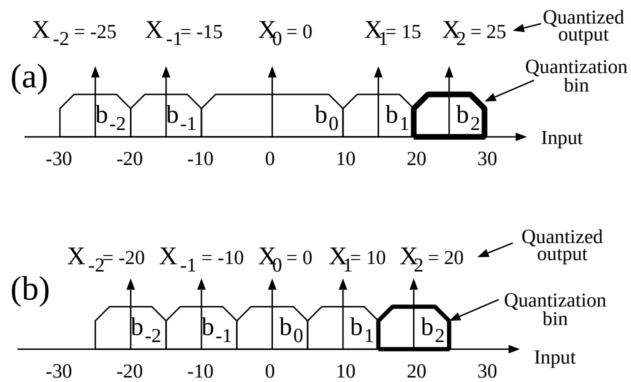


Figure 6. Examples of uniform and non-uniform quantization

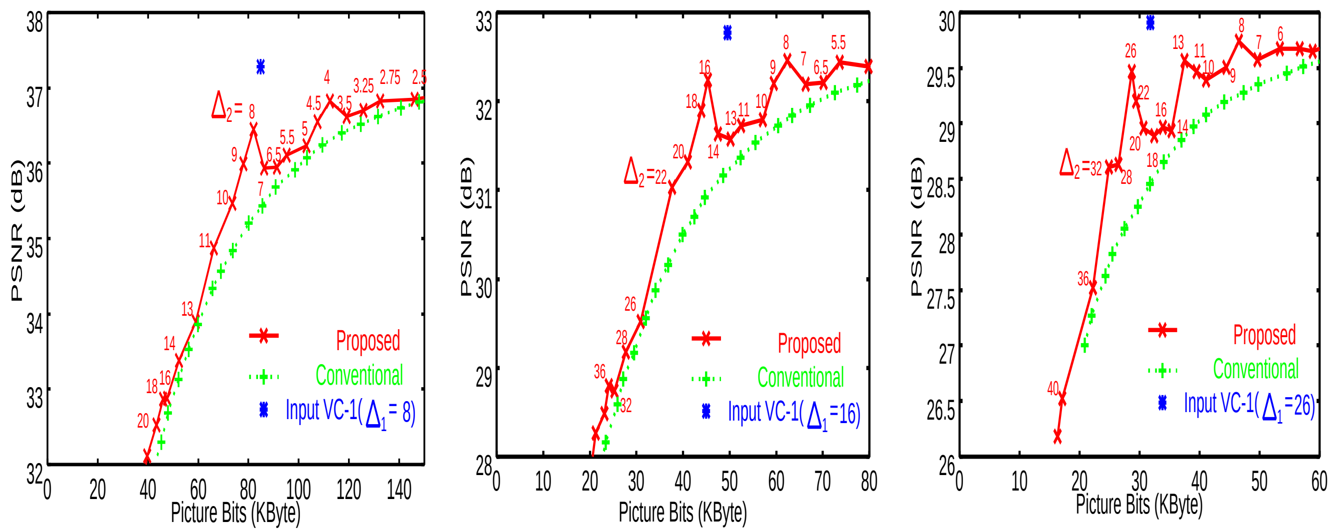
Table 4. Proposed VC-1/H.264 intra transcoding rule

	VC-1	H.264
Profile	simple/main/ advance	high
Intra prediction	up	0
	left	1
	no pred.	2(DC)
MB type	intra	I8x8
Encoder flag setting	-	$transform\_size\_flag$ = 1
Quantization matrix	-	flat matrix
Quantization step	-	nearest to VC-1's step

5 Simulation

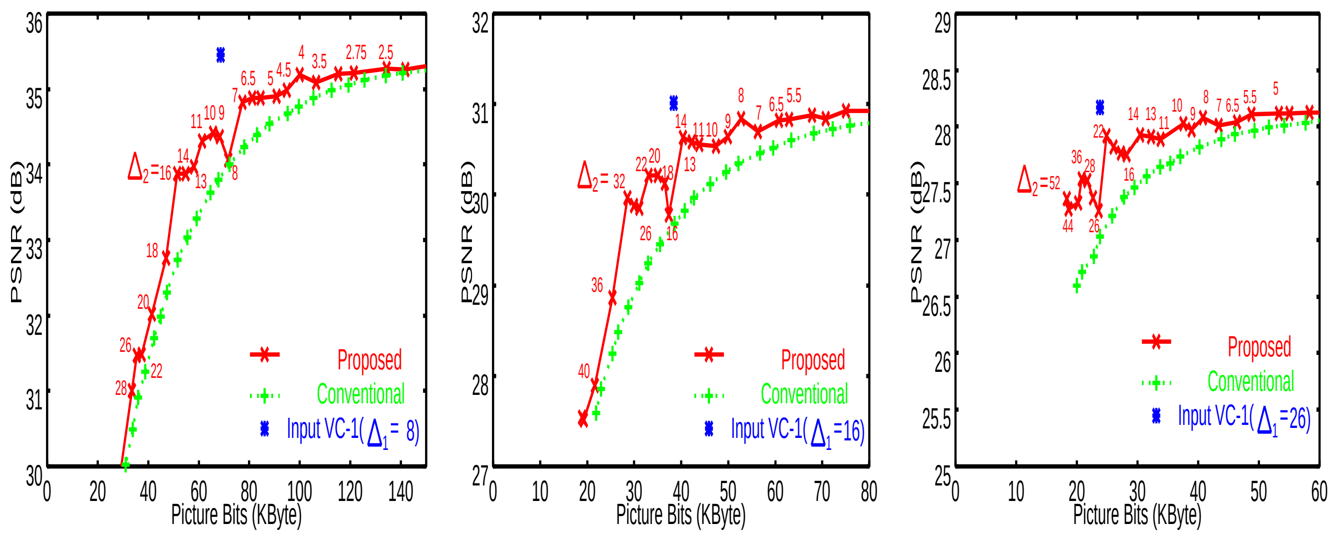
5.1 Conditions

We used experimental simulation to compare our proposed method with a conventional method. The conventional method we used is the ordinary VC-1/H.264 transcoding method. It does not re-use the first encoding information, and but does intra MB mode selection. The VC-1 sample program[14] was used for first encoding under the following conditions: The quantization step Δ was fixed in the picture. VC-1 encoding was done twice for all simulations using uniform and non-uniform quantization. We modified the VC-1 decoder in ffmpeg[15] to extract the first VC-1 encoding information from the VC-1 bitstream. The jm12.1 was used for the conventional second H.264 encoding under the following conditions: The profile was high-profile, the RD optimization was off, the UseHadamrd was on, and the quantization matrix was the H.264 default setting matrix. The conventional second encoding had no restrictions with regard to selecting MB mode and intra prediction mode, so its output bit-streams included the all-MB



$\Delta_1 = 8$ $\Delta_1 = 16$ $\Delta_1 = 26$

Figure 7. PSNR of proposed method and conventional method when VC-1 uses uniform quantization



$\Delta_1 = 8$ $\Delta_1 = 16$ $\Delta_1 = 26$

Figure 8. PSNR of proposed method and conventional method when VC-1 uses non-uniform quantization

mode and DC prediction, and also included all available DCT sizes. We used the modified jm12.1, which can input the first encoding information, for our proposed transcoding method. The quantization step Δ was fixed for the simulations of both the conventional and proposed methods.

5.2 Results for VC-1 bitstreams encoded by uniform quantization

The PSNR of transcoding from input VC-1 bitstreams encoded using uniform quantization to H.264 bitstreams are shown in Fig.7(a)-7(c). The solid line indicates the proposed method and the dotted line indicates the conventional method. The quantization step of the first encoding, Δ_1, was set to 8, 16, or 26. That of the second encoding, Δ_2, varied from 0.625 to 52. The x-axis means PSNR and the y-axis means picture bits. When input streams were encoded using uniform quantization, Fig. 7 shows that the proposed method showed two peaks when Δ_2=Δ_1 and Δ_2=0.5Δ_1. At these points, the PSNR of the proposed method was higher than that of the conventional method from about 0.98-1.28 dB. The PSNR of the proposed method drops when $0.5\Delta_1 < \Delta_2 < \Delta_1$ and $\Delta_1 < \Delta_2 < 2.0\Delta_1$. This PSNR drop is similar to that reported in MPEG-2/MPEG-2 transcoding. Figure 7(a)-(c) shows that the PSNR of the proposed method is always higher than that of the conven-

Table 5. Comp. ratio and PSNR of proposed method when VC-1 used uniform quantization

Input VC-1				Proposed transcoder			Conv.	PSNR
Image	Δ	PSNR (dB)	MB size (kB)	MB size (kB)	comp. ratio	PSNR (dB)	PSNR (dB)	diff. (dB)
Cheer Leader	8	37.28	84.9	82.00	0.966	36.44	35.29	1.16
	16	32.77	49.6	45.30	0.914	32.24	30.96	1.28
	26	29.91	31.8	29.44	0.925	29.20	28.22	0.98
Bus	8	36.99	79.7	72.66	0.911	36.20	35.14	1.05
	16	32.26	47.3	39.90	0.844	31.80	30.84	0.96
	26	29.30	28.1	24.15	0.861	28.76	28.19	0.56
Mobile and Calendar	8	35.96	124.6	120.64	0.968	34.71	33.86	0.85
	16	30.75	77.7	72.47	0.933	30.10	28.76	1.34
	26	27.44	51.1	48.01	0.939	26.75	25.72	1.03

Table 6. Comp. ratio and PSNR of proposed method when VC-1 used non-uniform quantization

Input VC-1				Proposed transcoder			Conv.	PSNR
image	Δ	PSNR (dB)	MB size (kB)	MB size (kB)	comp. ratio	PSNR (dB)	PSNR (dB)	diff. (dB)
Cheer Leader	8	35.45	68.6	61.30	0.893	34.31	33.42	0.89
	16	31.01	38.4	33.02	0.859	30.21	29.25	0.97
	26	28.17	23.8	20.77	0.872	27.54	26.70	0.84
Bus	8	35.01	65.0	53.98	0.830	34.01	33.26	0.75
	16	30.29	35.5	27.26	0.768	29.67	29.15	0.52
	26	27.53	19.7	15.43	0.782	27.06	26.62	0.44
Mobile and Calendar	8	34.05	105.2	94.05	0.894	32.54	31.75	0.79
	16	28.71	61.2	53.63	0.877	27.85	26.82	1.03
	26	25.41	36.9	33.29	0.902	24.86	23.96	0.91

tional method. As Δ_2 becomes smaller, the PSNR improvement of the proposed method decreases because the noise reduction effect mentioned in Section 2 maximizes when $\Delta_1 = \Delta_2$. As we mentioned in Sec.3.2, if the first encoding VC-1 uses the quantization step Δ_1 that is not allowed in H.264, an approximate quantization step must be used in H.264 transcoding. However, the difference between the true value and the approximation value is as high as 8.3%. The range in which the proposed method is superior to the conventional method exceeds this 8.3% error in Fig.7(a) - (c). This indicates that the proposed method is superior to the conventional method even if the approximate quantization step is used in H.264 transcoding. The compression ratio and PSNR of the proposed method when the quantization step Δ_2 is equal to Δ_1 are listed in Table 5. The conventional PSNR column is the interpolation value when the conventional method generates the same bitstream in terms of size. These interpolation values were calculated using the simulation results of the conventional method. The PSNR of the proposed method is about 0.56-1.34 dB better than that of the conventional method when the quantization step Δ_2 is equal to Δ_1. The re-compression ratio of the H.264 bitstream size to the VC-1 bitstream size ranges from 0.844 to 0.968.

5.3 Results for VC-1 bitstreams encoded using non-uniform quantization

The PSNR values of transcoding from input VC-1 bitstreams encoded using non-uniform non-quantization to H.264 bitstreams are shown in Fig.8(a)-8(c). When the first VC-1 encoding is done by non-uniform quantization, the PSNR improvement is less than that obtained with uniform quantization. Figure 8 shows the PSNR values obtained with the proposed method when Δ_2=$\theta\Delta_1$ (θ=1.1-1.3) . This shift in the optimum Δ_2 was generated by the difference between the quantization uniformities of VC-1 and H.264. This phenomenon was also reported in MPEG-2/H.264 inter transcoding research [13]. Despite the difference in quantization uniformity, Fig.8(a)-(c) shows that the PSNR of the proposed method is always higher than that of the conventional method. The compression ratio and PSNR of the proposed method when the quantization step Δ_2 is equal to $1.2\Delta_1$ are listed in Table.6. At these shifted points, the PSNR of the proposed method is higher than that of the conventional method from about 0.44-1.03 dB.

6 Conclusion

We proposed a re-quantization noise reduction method for VC-1 to H.264 intra transcoding. This method uses encoding information from the VC-1 stream and keeps as many

DCT coefficients of the original VC-1 bitstream as possible. We analyzed the difference between VC-1 and H.264 regarding the intra prediction and quantization steps and orthogonal transformation to reduce the re-quantization noise in H.264. Experimental results showed that the proposed method improves PSNR by about 0.44-1.34 dB compared with the conventional method. The advantages of the proposed method are that it not only results in high PSNR but also that it does not require complex calculation to select an MB mode and an intra prediction mode.

References

[1] ISO/IEC IS 13818-2, ITU-T Recommendation H.262, “Generic coding of moving pictures and associated audio information“, Nov. 1994.

[2] “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec.H.264/ISO/IEC 14486-10 AVC“, in Joint Video Team (JVT) of ISO/IEC MPEG and ITU-T VCEG, JVT-G050,2003

[3] SMPTE, Standard for Television: VC-1 Compressed Video Bitstream Format and Decoding Process, SMPTE 421M-2006.

[4] P. Guilotel, et al, ”Adaptive Encoders: The New Generation of MPEG-2 Encoders” SMPTE journal (April 2000)

[5] J.-B. Lee, and H. Kalva, ”An Efficient Algorithm for VC-1 to H.264 Video Transcoding in Progressive Compression, Proc. IEEE International Conf. on Multimedia and Expo, pp.53-56, July 2006.

[6] M. Pantoja, N. Ling, and W. Shang, ”Coefficient Conversion for Transform Domain VC-1 to H.264 Transcoding,” Proc. 2007 IEEE Workshop on Signal Processing Systems, Shanghai, China, pp. 363-367, October 17-19, 2007.

[7] M. Pantoja, and N. Ling, ”Adaptive transform size and frame-field selection for efficient VC-1 to H.264 high profile transcoding,” Proc. 2009 IEEE Workshop Circuits and Systems, pp. 2357-2360, ISCAS 2009.

[8] Xiaoan Lu, et.al ”Fast mode decision and motion estimation for H.264 with a focus on MPEG-2/H.264 transcoding” in Proc. of ISCAS 2005, 23-26 May 2005, pp1246-1249 Vol.2

[9] Gao Chen, et.al ”Efficient block size selection for MPEG-2 to H.264 transcoding” Proc. of the 12th annual ACM international conference on Multimedia

[10] Joo-Kyong Lee, et.al ”Quantization/DCT Conversion Scheme for DCT-domain MPEG-2 to H.264/AVC Transcoding” IEICE Trans. Commun., Vol.E87-B, No.7 July 2004

[11] T. Yoshitome, et al, ”A study of MPEG-2 to H.264 Intra Transcoding for Progressive Contents” ITE Journal, **62**,11,pp1819-1824 (Nov. 2008) (in Japanese)

[12] T. Yoshitome, K. Kamikura, and N. Kitawaki, ”A Requantization Noise Reduction Method in MPEG-2 to H.264 Intra Transcoding” IEEE Symposium on Industrial Electronics and Application (ISIEA 2009), Kuala Lumpur, Malaysia, pp.94-99, Oct 4-6, 2009

[13] T. Yoshitome, et al, ”An MPEG-2 to H.264 Transcoding Method Preserving DCT Information for Progressive Contents” ITE Journal, **63**,6,pp.837-846 (June 2009) (in Japanese)

[14] Intel Integrated Performance Primitives 6.1 http://software.intel.com/en-us/intel-ipp/

[15] http://ffmpeg.org/

[16] Joint Video Team(JCT), “Reference Software JM12.1”, http://iphome.hhi.de/suehring/tml/

[17] High vision digital standard image data http://www.ite.or.jp/shippan/test_chart_index.html