Efficient Level and Zero Coding Methods for H.264/AVC Lossless Intra Coding

Jin Heo, *Student Member, IEEE*, and Yo-Sung Ho, *Senior Member, IEEE*

***Abstract*—Since H.264/AVC was designed mainly for lossy video coding, the entropy coding methods in H.264/AVC are not appropriate for lossless video coding. Based on statistical differences of residual data in lossy and lossless coding, we develop efficient level and zero coding methods. Therefore, we design an improved context-based adaptive variable length coding (CAVLC) scheme for lossless intra coding by modifying the relative entropy coding parts in H.264/AVC. Experimental results show that the proposed method provides approximately 6.8% bit saving, compared with the H.264/AVC FREXt high profile.**

***Index Terms*—CAVLC, H264/AVC, intra coding, lossless video coding.**

I. Introduction

H.264/AVC is a new international video coding standard [1]–[3]. Since H.264/AVC has been applied mostly for lossy video coding, it does not provide good coding performance for lossless video coding. In order to provide improved functionality for lossless video coding, H.264/AVC included a *pulse-code modulation* (PCM) macroblock coding mode and a transform-bypass [4] lossless mode that employs two main coding processes: entropy coding and prediction in the fidelity range extensions (FREXt) [5]. However, more efficient coding techniques for prediction and entropy coding are still required.

Recently, new intra prediction methods based on *differential pulse-code modulation* (DPCM) were introduced for lossless intra prediction [6], [7]. These methods have been shown to provide better compression performance, and sample-wise DPCM was subsequently adopted as a part of the new draft amendment for the H.264/AVC standard [8].

For lossless intra coding in FREXt, the original sample values are coded by two coding processes, intra prediction and entropy coding, because transform and quantization are not used. In other words, for lossless coding, prediction errors are directly coded by the entropy coder without transform and quantization. On the contrary, for lossy coding, quantized transform coefficients are encoded by the entropy coder. Hence, there are significant statistical differences of residual data between lossy and lossless intra coding. Therefore, the current context-based adaptive variable length coding (CAVLC) [9] in H.264/AVC is not appropriate for lossless intra coding.

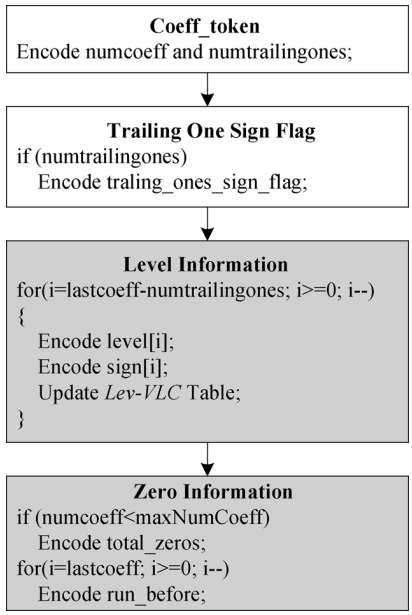

Fig. 1. Encoding structure of CAVLC.

In this letter, we propose a new CAVLC scheme for lossless intra coding in H.264/AVC by modifying the semantics and the decoding process without requiring any other syntax elements in H.264/AVC. We changed coding methods for the syntax elements of residual data considering their statistical characteristics. As a result, the basic CAVLC tools originally designed for lossy intra coding are extended to lossless intra coding. In the proposed CAVLC scheme, we modified three main coding processes of the conventional CAVLC in level and zero coding parts. Experimental results show that the proposed scheme improves coding performance, compared with the H.264/AVC.

II. Context-Based Adaptive Variable Length Coding

The encoding structure of CAVLC for residual data in each 4×4 subblock is depicted in Fig. 1.

The detailed coding procedure of CAVLC is as follows.

Step 1: Both the total number of non-zero coefficients (*numcoeff*) and the number of trailing ones (*numtrailingones*) are encoded using a combined codeword (*coeff_token*). A trailing one is one of up to three consecutive non-zero coefficients at the end of the scan of non-zero coefficients having an absolute value equal to 1.

Manuscript received May 19, 2009; revised August 06, 2009. First published September 04, 2009; current version published October 28, 2009. This work was supported by the MKE, Korea, under the ITRC support program supervised by IITA-2009-(C1090-0902-0017).

The authors are with the School of Information and Mechatronics, Gwangju Institute of Science and Technology (GIST), Gwangju 500-712, Korea (e-mail: jinheo@gist.ac.kr; hoyo@gist.ac.kr).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2009.2031710

TABLE I
THRESHOLD VALUES FOR SELECTING THE "*LEV-VLC*" TABLE

Lev-VLC tables	Threshold values
Lev-VLC0	0
Lev-VLC1	3
Lev-VLC2	6
Lev-VLC3	12
Lev-VLC4	24
Lev-VLC5	48
Lev-VLC6	> 48

Step 2: The sign of each trailing one is encoded using one bit codeword in reverse order (*trailing_ones_sign_flag*).

Step 3: The absolute level value of each remaining non-zero coefficient (*abs_level*) is encoded in reverse order using one of the seven predefined *Lev-VLC* tables (Table I). The sign information is encoded in the same way as in Step 2. Selection of the *Lev-VLC* table is dependent on the magnitude of the previously encoded level.

1) If (*numcoeff* >10 && *numtrailingones* ==3)

Initialize $Lev\text{-}VLC1$.

Otherwise,

Initialize $Lev\text{-}VLC0$.

2) Encode *abs_level* in reverse order.
3) Encode the sign of the non-zero coefficient.
4) If the magnitude of the current encoded coefficient is larger than a predefined threshold value in Table I, increment the *Lev-VLC* table.

Step 4: The number of all zeros before the last non-zero coefficient (*total_zeros*) is encoded.

Step 5: The number of consecutive zeros preceding each non-zero coefficient (*run_before*) is encoded using a selected VLC table based on *run_before* and *zerosleft*, where *zerosleft* indicates the number of zeros that has not been encoded, starting with the highest frequency, with two exceptions:

1) If there are no *zerosleft* to encode, processing is terminated.
2) Processing is terminated to encode *run_before* for the final (the lowest frequency) non-zero coefficient.

III. PROPOSED METHOD

In this section, we describe a new CAVLC scheme for lossless intra coding by reflecting the statistical properties of residual sample values. In Fig. 1, the gray-shaded processes are modified and removed in the proposed scheme for lossless intra coding; further details are described in Sections III-A and B.

A. Level Coding

In level coding, the absolute level value of each non-zero coefficient (*abs_level*) is encoded adaptively based on the selected *Lev-VLC* table from among the seven predefined *Lev-VLC* tables (*Lev-VLC0* to *Lev-VLC6*) in reverse scanning order. Each *Lev-VLC* table is designed to efficiently encode *abs_level* in a specified range of *abs_level*, as indicated in Table I. Selection of the *Lev-VLC* table for level coding in CAVLC is based on the expectation that *abs_level* is likely to increase toward the low frequency regions. Hence, selection of the *Lev-VLC* table monotonically increases according to the previously encoded *abs_level*.

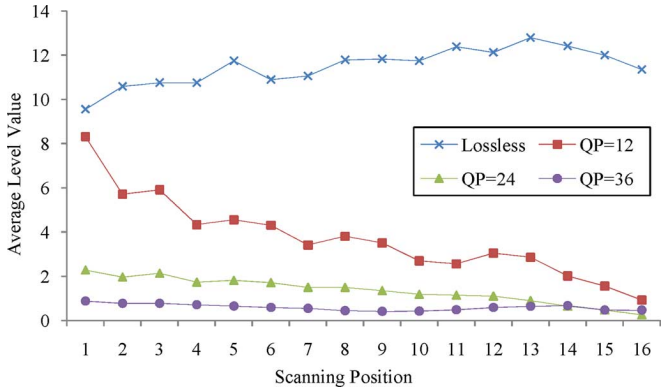

Fig. 2. Distribution of average absolute level values according to the scanning position ("Tempete," CIF).

TABLE II
NEW THRESHOLD VALUES FOR SELECTING THE "*LEV-VLC*" TABLE

Lev-VLC tables	Best coding range for *abs_level*	Average threshold values	New threshold values
Lev-VLC0	1	1	1
Lev-VLC1	1, 2	1.5	2
Lev-VLC2	1 – 5	3	3
Lev-VLC3	2 – 11	6.5	7
Lev-VLC4	4 – 23	13.5	14
Lev-VLC5	8 – 47	27.5	28
Lev-VLC6	> 15	-	-

However, *abs_level* in lossless coding is independent of the scanning position, as shown in Fig. 2. Therefore, we designed an adaptive method for *Lev-VLC* table selection that can decrease or increase according to the previously encoded *abs_level*.

In lossy coding, CAVLC typically determines the smallest *Lev-VLC* table in a range of possible *Lev-VLC* tables based on the assumption that the next *abs_level* to be coded is going to be larger. However, since the next *abs_level* does not necessarily increase at lower frequencies in lossless coding, we cannot assume that the next *abs_level* is larger than the current *abs_level*. Therefore, theoretically, a new threshold value for each *abs_level* should be selected as the average threshold value in lossless coding because we cannot predict whether or not the next *abs_level* will increase.

In Table II, we represent new threshold values for selecting the corresponding *Lev-VLC* table. In the proposed level coding, selection of the *Lev-VLC* table can increase or decrease according to the previously encoded *abs_level*.

In Fig. 2, we can observe that the last scanned *abs_level* is quite different in lossy and lossless coding. In lossy coding, level encoding starts with *Lev-VLC0* or *Lev-VLC1* because the last scanned *abs_level* is likely to be small; however, in lossless coding, the last scanned *abs_level* is not small enough to use either *Lev-VLC0* or *Lev-VLC1*. Through our extensive experiments on lossless coding with various video sequences, we find that the average values of the last scanned *abs_level* in the chroma DC subblock and other subblocks are approximately 6.43 and 11.29, respectively. Therefore, we initialize

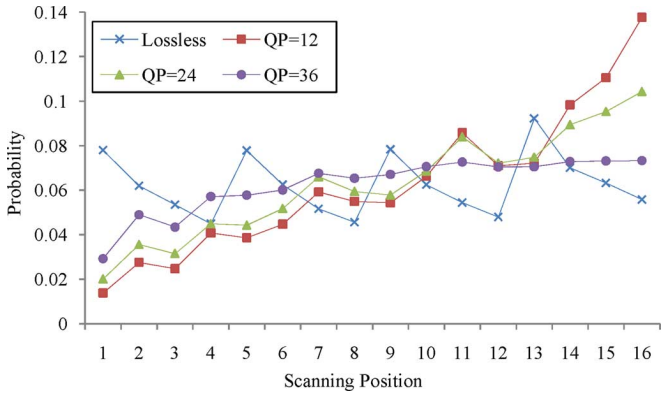

Fig. 3. Occurrence probability distribution of zero coefficients according to the scanning position ("Tempete," CIF).

Scanning position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Coefficient level	0	3	-2	0	1	0	0	0	0	0	0	0	0	0	0	0
Coding order	*abs_level*				*zerosleft*						*run_before*					
1	1				2						1					
2	-2				1						0					
3	3				1						1					

Fig. 4. Syntax element *run_before* coding in CAVLC.

Scanning position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Coefficient level	0	3	-2	0	1	0	0	0	0	0	0	0	0	0	0	0
Coding order	*abs_level*				*all_zeros*						*run_before*					
1	3				13						1					
2	-2				12						0					
3	1				12						1					

Fig. 5. Syntax element *run_before* coding in the proposed method.

the *Lev-VLC* tables as *Lev-VLC3* and *Lev-VLC4* for each case. The modified level coding method is as follows:

1) If the current level is in a chroma DC subblock

$$\text{Initialize } Lev\text{-}VLC3.$$

Otherwise,

$$\text{Initialize } Lev\text{-}VLC4.$$

2) Encode *abs_level* in reverse order.
3) Encode the sign of the non-zero coefficient.
4) Update *Lev-VLC* table by considering the previously encoded *abs_level* and new threshold value for each *Lev-VLC* table.

B. Zero Coding

In zero coding, two syntax elements for the zero coefficient, how many zeros are and where each zero is in the subblock, are encoded. After the total number of zero coefficients located before the last non-zero coefficient (*total_zeros*) is encoded, the number of consecutive zero coefficients located before each non-zero coefficient (*run_before*) is encoded using a selected VLC table. The VLC table is chosen based on *run_before* and *zerosleft*, where *zerosleft* indicates the number of zeros that has not yet been encoded before each non-zero coefficient starting from the last non-zero coefficient.

In lossy coding, *run_before* is encoded along the backward scanning path to increase coding efficiency because subblocks typically contain many zeros in the high frequency regions, as shown in Fig. 3. However, the probability distribution of zero coefficients in lossless coding is independent of the scanning position.

In lossless coding, we cannot assume that the occurrence probability of zero coefficients will decrease along the backward scanning path. Thus, we change *run_before* encoding order, i.e., we encode *run_before* along the forward scanning path, thereby eliminating the need to encode *total_zeros*. The reason for encoding *total_zeros* in lossy coding is that we do not know the location of the last non-zero coefficient. However, since *run_before* encoding process starts from the low frequency regions in lossless coding, the scanning position of the last non-zero coefficient (*lastcoeff*) does not need to be considered; therefore, we can remove *total_zeros* coding process.

In lossy coding, the VLC table used to encode *run_before* is selected based on *run_before* and *zerosleft*. In Fig. 4, we indicate an example of *run_before* coding in CAVLC when *lastcoeff* is 5; the gray-shaded *run_befores* are encoded. In this example, it is not necessary to encode *run_before* corresponding to $abs\_level = 3$ (scanning position number $= 2$) because the current *abs_level* is the final (the lowest frequency) non-zero coefficient. This condition is satisfied based on the second exception rule of Step 5 in Section II.

Fig. 5 represents an example of *run_before* coding in the proposed method; the VLC table used to encode *run_before* is chosen based on *run_before* and *all_zeros*, where *all_zeros* indicates the number of zeros that has not yet been encoded after each non-zero coefficient starting from the DC coefficient. In this case, the conventional rule for *run_before* coding is generally applied to each *run_before*, with one exception: the coding order is changed. If the conventional rules are applied to this example, each *run_before* corresponding to $abs\_level = 3 \text{ and } -2$ is encoded, except for $abs\_level = 1$. If we decode this encoding result, we can obtain the final outputs: 0, 3, −2, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0. However, this decoding data is not equal to the original data; hence, we should modify the conventional rule for *run_before* coding to handle this problem.

Since we change the coding order from backward scanning to forward scanning and do not encode *total_zeros*, we should encode *run_before* corresponding to the final (the highest frequency) non-zero coefficient in lossless coding. Thus, in Fig. 5, *run_before* corresponding to $abs\_level = 1$ (scanning position number $= 5$) should be encoded; in all other cases, no matter which *run_before* has been encoded, there is no alteration to *run_before* coding method. Finally, the gray-shaded *run_befores* are encoded in lossless coding, as shown in Fig. 5. The final two exceptions for *run_before* coding in lossless coding are as follows.

1) If there are no *all_zeros* to encode, processing is terminated.
2) Processing is terminated after encoding *run_before* for the final (the highest frequency) non-zero coefficient.

TABLE III
ENCODING PARAMETERS

ProfileIDC	244 (High 4:4:4)
IntraPeriod	1 (only intra coding)
QPISlice	0 (lossless)
SymbolMode	0 (CAVLC)
QPPrimeYZeroTransformBypassFlag	1

In lossy coding, we use the VLC table to encode *run_before* in the range from 0 to 14 with code lengths from 1 to 11 because *run_before* for the final (the lowest frequency) non-zero coefficient is not encoded. However, in lossless coding, *run_before* for the final (the highest frequency) non-zero coefficient should be encoded; hence, we assigned a new codeword for $run\_before = 15$ with $\text{code length} = 12$ and $\text{codeword} = 1$ to avoid ambiguity at the decoder.

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to evaluate coding efficiency of the proposed CAVLC scheme, we have performed intensive experiments on several video sequences of the YUV 4:2:0 format with QCIF (176×144), CIF (352×288), and HD (1920×1080) resolutions. We have implemented our proposed CAVLC scheme in the H.264/AVC reference software version JM 13.2 [10]. Table III represents the encoding parameters for lossless intra coding.

Note that the proposed CAVLC scheme was applied to H.264/AVC lossless intra coding by modifying the semantics and the decoding processes without adding any other syntax elements to the H.264/AVC standard. In our experiments, we have compared the well-known lossless coding technique, context-based adaptive lossless image coding (CALIC) [11] with our proposed CAVLC scheme. For this comparison, we used CALIC source code downloaded from the CALIC site [12]. We have compared their performances in terms of bit-rate percentage differences and compression ratio differences, with respect to H.264/AVC, which were defined by (1) and (2).

$$\Delta\, Saving\ Bits(\%) = \frac{Bitrate_{H.264/AVC} - Bitrate_{Method}}{Bitrate_{H.264/AVC}} \times 100 \quad (1)$$

$$Compression\ Ratio = \frac{Original\ image\ size}{Bitrate_{Method}}. \quad (2)$$

From Table IV, we can observe that the proposed CAVLC scheme is superior to both CALIC and the current H.264/AVC FREXt high profile.

TABLE IV
COMPARISON OF COMPRESSION RATIO AND SAVING BITS

Image	Total encoding frame	Image size (bits)	Method	Total bits (bits)	Compression ratio	Saving bits (%)
Foreman (QCIF)	150 frames	45619200	H.264/AVC	21119040	2.1601	0
			CALIC	25880400	1.7627	-22.5453
			Proposed	20075952	2.2723	**4.9391**
Mobile (QCIF)	150 frames	45619200	H.264/AVC	29793808	1.5312	0
			CALIC	34205400	1.3337	-14.8071
			Proposed	26613808	1.7141	**10.6734**
Tempete (CIF)	150 frames	182476800	H.264/AVC	97856096	1.8647	0
			CALIC	112462200	1.6226	-14.9261
			Proposed	90709024	2.0117	**7.3037**
Flowergarden (CIF)	150 frames	182476800	H.264/AVC	103308112	1.7663	0
			CALIC	113932800	1.6016	-10.2845
			Proposed	93099064	1.9600	**9.8821**
Crowdrun (HD)	100 frames	2488320000	H.264/AVC	1177625784	2.1130	0
			CALIC	1499731200	1.6592	-27.3521
			Proposed	1135423808	2.1915	**3.5836**
Parkrun (HD)	100 frames	2488320000	H.264/AVC	1210958808	2.0548	0
			CALIC	1638843600	1.5183	-35.3344
			Proposed	1152238192	2.1596	**4.8491**
Average			H.264/AVC		1.9150	0
			CALIC		1.5830	-20.8749
			Proposed		2.0515	**6.8718**

V. CONCLUSION

In this letter, we proposed an improved context-based adaptive variable length coding (CAVLC) scheme for lossless intra coding. Considering the statistics of residual pixel values obtained from lossless intra prediction, we modified the encoding mechanism of CAVLC. Experimental results show that the proposed CAVLC scheme provides approximately 6.8% bit saving, compared with the H.264/AVC FREXt high profile.

REFERENCES

[1] *Advanced Video Coding for Generic Audiovisual Services*, ITU-T Rec. H.264 and ISO/IEC 14496-10, May 2003.

[2] T. Wiegand, G. J. Sullivan, G. Bjøntegaard, and A. Luthra, "Overview of the H.264/AVC video coding standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 560–576, Jul. 2003.

[3] G. J. Sullivan and T. Wiegand, "Video compression-from concepts to the H.264/AVC standard," *Proc. IEEE*, pp. 18–31, Jan. 2005.

[4] S. Sun, Intra Lossless Coding and QP Range Selection ISO/IEC JTC1/SC29/WG11 and ITU-T Q.6/SG16 Joint Video Team, May 2002, Doc. JVT-C023.

[5] G. J. Sullivan, P. Topiwala, and A. Luthra, "The H.264/AVC advanced video coding standard: Overview and introduction to the fidelity range extensions," in *SPIE Conf., Special Session on Advances in the New Emerging Standard: H.264/AVC*, Aug. 2004.

[6] Y.-L. Lee, K.-H. Han, and G. J. Sullivan, "Improved lossless intra coding for H.264/MPEG-4 AVC," *IEEE Trans. Image Processing*, vol. 15, no. 9, pp. 2610–2615, Sep. 2006.

[7] N. Krishnan, R. K. Selvakumar, P. Vijayalakshmi, and K. Arulmozhi, "Adaptive single pixel based lossless intra coding for H.264/MPEG-4 AVC," in *IEEE ICCIMA '07*, Dec. 2007, vol. 3, pp. 63–67.

[8] H. Yu, Draft Text of H.264/AVC Advanced 4:4:4 Profile Amendment to ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC ISO/IEC JTC1/SC29/WG11 and ITU-T Q.6/SG16 Joint Video Team, Oct. 2005, document JVT-Q209.

[9] I. E. G. Richardson, *H.264 and MPEG-4 Video Compression*. Hoboken, NJ: Wiley, 2003.

[10] Joint Video Team [Online]. Available: http://iphome/hhi.de/shehring/tml/download/old_jm/jm13.2.zip

[11] X. Wu and N. Memon, "Context-based, adaptive, lossless image coding," *IEEE. Trans. Commun.*, vol. 45, no. 4, pp. 437–444, Apr. 1997.

[12] [Online]. Available: http://newsgroups.derkeiler.com/Archive/Comp/comp.compression/2006–02/msg00152.html