# An Efficient Table Prediction Scheme for CAVLC in H.264

*Jin Heo, Kwan-Jung Oh, and Yo-Sung Ho*

Department of Information and Communications
Gwangju Institute of Science and Technology (GIST)
1 Oryong-dong, Buk-gu, Gwangju, 500-712, Republic of Korea
{jinheo, kjoh81, hoyo}@gist.ac.kr

**Abstract.** The most recent video coding standard H.264 has adopted context-based adaptive variable-length coding (CAVLC) as the entropy coding tool in the baseline profile. By combining an adaptive variable-length coding technique with context modeling, we can achieve a high degree of redundancy reduction. However, CAVLC in H.264 has a weakness that the correct prediction rate of the variable-length coding (VLC) table is low. In this paper, we propose a new VLC table prediction scheme considering multiple reference blocks; the block at the same position in the previous frame and neighboring blocks of the current block in the current frame, and weighting values considering quantization parameter (QP). Based on the correctness of the VLC table prediction depending on QP values for each reference block, we determine weighting values. Experimental results show that the proposed algorithm increases the correct prediction rate of the VLC table by 7.18 % and reduces bit rates by 0.78 %, compared to the H.264 algorithm.

***Index Terms***—H.264/AVC, CAVLC, VLC table

## 1. INTRODUCTION

H.264 provides high compression efficiency compared to previous video coding standards, such as MPEG-4 and H.263. This efficiency is achieved by using variable block size macroblock modes, multiple reference frames, 1/4 pel motion compensation, deblocking filter, integer transform, and efficient entropy coding techniques, such as context-based adaptive variable-length coding (CAVLC) or context-based adaptive binary arithmetic coding (CABAC) [1] [2].

Variable-length coding (VLC) plays an important role in video coding of nowadays. The main idea of variable-length coding is to minimize the average codeword length. Shorter codewords are assigned to frequently occurring data and longer codewords are assigned to infrequently occurring data [3].

In order to further increase the data compression ratio, H.264 has adopted the CAVLC for entropy coding, where quantized transform coefficients are coded using VLC tables that are switched depending on the values of previous syntax elements [4] [5]. Therefore, the coding efficiency of CAVLC depends on how accurately estimate an appropriate VLC table. However, CAVLC has a drawback that the correct prediction rate of the VLC table is low. It again affects the choice of the VLC table using syntax elements. Consequently, the optimal VLC table is not used for encoding of quantized transform coefficients.

In this paper, we propose a new VLC table prediction scheme using multiple reference blocks and weighting factors to increase the correctness of the VLC table prediction. With the proposed scheme, we can choose the suitable VLC table for encoding of the current 4×4 block.

## 2. CAVLC IN H.264

In this section, we briefly describe the CAVLC scheme in H.264 and explain the VLC table prediction algorithm. The entropy coding represents the lossless part in the advanced video coding (AVC) encoding process. In combination with transformations and quantizations, it can cause significant increase of compression ratio [5].

In H.264 baseline profile, two types of entropy coding are supported. The first entropy coding method uses a single infinite-extent codeword table for all syntax elements except for residual elements. Thus, instead of designing a different VLC table for each syntax element, only the single codeword table is customized according to syntax element statistics. The single codeword table is an exp-Golomb (Exponential Golomb) code with simple and regular decoding properties [6].

Another entropy coding method is CAVLC for encoding of quantized coefficients. CAVLC is designed to take advantage of several characteristics of quantized 4×4 blocks. First, after the quantization of transformed coefficients the distribution of coefficients is sparse in general. CAVLC uses run-level coding to represent strings of zeros compactly. Second, highest frequency non-zero coefficients after the zig-zag scan are usually sequences of ±1 and CAVLC encodes the number of high frequency ±1 coefficients. Third, the number of non-zero coefficients in neighboring blocks is correlated. The number of coefficients is encoded using a VLC table. The choice of VLC table depends on the number of non-zero coefficients in neighboring blocks. Finally, the level of non-zero coefficients is usually higher at the start of the reordered array. Therefore, the choice of VLC tables for the level value depends on recently coded level magnitudes. [7]

As shown in Fig. 1, CAVLC has typically five major steps [8].

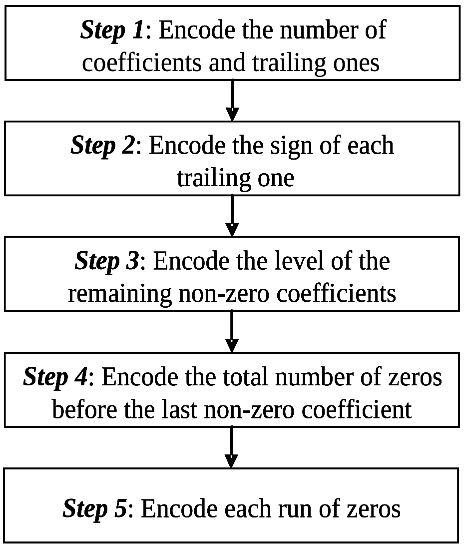



**Fig. 1.** Flowchart of the CAVLC scheme

In Step 1, trailing ones indicate the number of coefficients with absolute value equal to 1 at the end of the scan. The number of non-zero coefficients and the number of trailing ones are coded using a combined codeword, where one of four VLC tables is used based on the number of non-zero coefficients of neighboring blocks.

In Step 2, trailing ones only need the sign specification since they are all equal to ±1. Thus, the sign is encoded with a single bit (0 = +, 1 = −) for each trailing one in reverse order, starting with the highest frequency trailing one.

In Step 3, the level (sign and magnitude) of each remaining non-zero coefficient in the current 4×4 block is encoded in reverse order, starting with the highest frequency and working back towards the DC coefficient. The code for each level is made up of a prefix and a suffix. The length of the suffix is adapted depending on the magnitude of each successive encoded level by choosing among six tables.

In Step 4, the number of all zeros preceding the highest frequency non-zero coefficient of each 4×4 block is encoded.

In Step 5, the number of zeros preceding each non-zero coefficient (*run_before*) is encoded in reverse order. A *run_before* parameter is encoded for each non-zero coefficient, starting with the highest frequency.

In CAVLC, VLC tables for syntax elements for the current block are switched depending on previously coded syntax elements. These improve the coding efficiency compared to schemes that use a single VLC table.

At the first step, there are four choices of VLC tables used in encoding of both the total number of non-zero coefficients and the number of trailing ±1 values in the current 4×4 block. Four VLC tables are Num-VLC0, Num-VLC1, Num-VLC2, and Num-FLC (3 variable-length code tables and a fixed length code table). The choice of VLC table depends on the number of non-zero coefficients in upper block ($N_U$) and left block ($N_L$) as shown in Fig. 2.

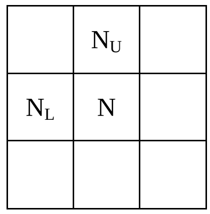



**Fig. 2.** Left and upper blocks of the current 4×4 block

If upper and left blocks are both available, a parameter $N$ is calculated as

$$N = round(N_U + N_L)/2. \qquad (1)$$

where $N$ represents the number of predicted non-zero coefficients in the current block. $N_U$ and $N_L$ are the number of non-zero coefficient of upper and left blocks, respectively. If any block is not available, $N$ is set to 0. By using parameter $N$, we choose the VLC table from Table 1 with which we encode transformed coefficients in the current block. The choice of VLC table adapts to the number of coded coefficients in neighboring blocks [4].

**Table 1.** Choice of VLC table

| N | VLC table for parameter |
|---|---|
| 0, 1 | Num-VLC0 |
| 2, 3 | Num-VLC1 |
| 4, 5, 6, 7 | Num-VLC2 |
| 8 or above | Num-FLC |

However, H.264 has a drawback that correctness of the VLC table prediction is low in some cases, such as only one reference block is available or when any block is not available or the area at the current block is complex such as boundaries of moving objects. Figure 3 shows the correct prediction rate of the VLC table. As you can see in Fig. 3, the correct prediction rate of the VLC table for five test sequences such as "Foreman", "News", "Container", "Silent", and "Carphone" is approximately 55 % on average. In order to improve prediction accuracy, we propose an efficient VLC table prediction scheme by using multiple reference blocks and weighting factors considering QP.

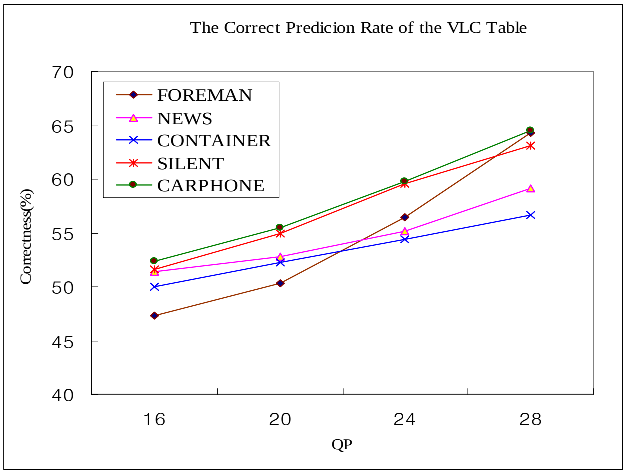



**Fig. 3.** Correct prediction rate of the VLC table

In Table 2, we compare resulting bit rates between the H.264 CAVLC and ideal CAVLC. Ideal CAVLC has 100 % correctness of VLC table prediction. As you can see, if we use the correct VLC table for the current 4×4 block, we can reduce the bit rate by approximately 3 %, compared to the CAVLC in H.264. Table 2 means that if we can increase correctness of the VLC table prediction for each test sequence, we can increase the coding efficiency.

**Table 2.** Comparison of correctness and bit rates

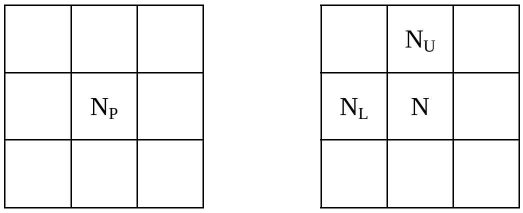
| QP | CAVLC | | Ideal CAVLC | |
|---|---|---|---|---|
| | Correctness (%) | Bit Rate (kbps) | Correctness (%) | Bit Rate (kbps) |
| 16 | 47.32 | 842.13 | 100 | 818.57 |
| 20 | 50.32 | 451.64 | 100 | 435.41 |
| 24 | 56.47 | 233.73 | 100 | 226.43 |
| 28 | 64.31 | 115.52 | 100 | 112.92 |

## 3. PROPOSED ALGORITHM

In CAVLC, overall coding efficiency of a given sequence highly depends on how to select the proper VLC table. The proposed algorithms try to find the suitable VLC table. In this section, we describe two proposed algorithms such as multiple reference blocks and weighting factors.

### 3.1 Multiple Reference Blocks

The VLC table prediction scheme in H.264 selects a VLC table by considering left and upper blocks of the current 4×4 block. However, as mentioned before, it is difficult to select the VLC table exactly in some cases. Such as when only the left or upper block is available, when no blocks are available for reference, and when the area is complex such as boundaries of moving objects. Therefore, we propose a new VLC table prediction scheme using the block at the same position in the previous frame and available reference blocks; the upper block and the left block in the current frame as shown in Fig. 4.




(a) Previous frame (b)Current frame

**Fig. 4.** Multiple reference blocks

A parameter $N$ is calculated by

$$N = round(N_U + N_L + N_p)/3. \qquad (2)$$

From Eq. (1), we can obtain Eq. (2). $N_P$ represents the number of non-zero coefficients at the same position 4×4 block of the current block in the previous frame. By using the parameter $N$ from the Eq. (2), we can choose the proper VLC table for the current block from Table 1.

### 3.2 Determination of Weighting Values

We need to determine weighting values considering correlation between the reference and current blocks. In order to determine proper weighting values, we use correctness of VLC table prediction of each reference block; the upper block, the left block, and the previous block.

Table 3 shows correctness of VLC table prediction for each reference block.

**Table 3.** Correctness of the VLC table prediction

| QP | Current frame | | Previous frame |
|---|---|---|---|
| | Left block (%) | Upper block (%) | Same position block (%) |
| 16 | 39.76 | 36.74 | 54.34 |
| 20 | 46.94 | 43.18 | 57.99 |
| 24 | 54.95 | 51.07 | 62.16 |
| 28 | 62.08 | 59.18 | 67.22 |

In Table 3, while correctness of two reference blocks of the current frame is low, correctness of the block of the previous frame is higher than that of two reference blocks. Also, correctness of two reference blocks of the current frame is similar to each other. Based on this observation, we propose new weighting values for each reference block as follows

$$N = round(\alpha \cdot (N_U + N_L) + \beta \cdot N_p). \qquad (3)$$

where $\alpha$ and $\beta$ represent the weighting values of each reference block ($2\alpha+\beta=1$).

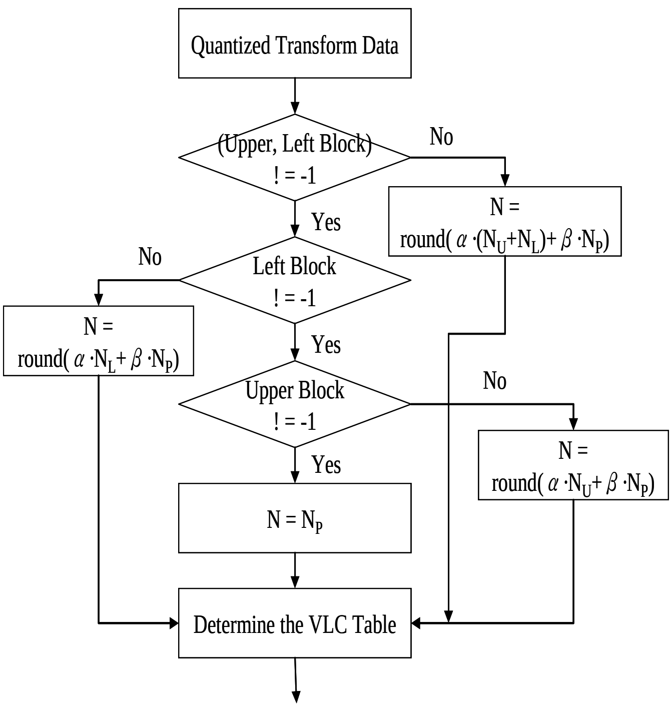



**Fig. 5.** Flowchart of the proposed algorithm

Figure 5 shows the flowchart of the proposed VLC table prediction algorithm. If both upper and left blocks ($N_U$ and $N_L$) are available, we should consider three reference blocks (the upper block, the left block, and the previous block). If only the upper block ($N_U$) or the left block ($N_L$) is available, we consider two reference blocks (either the upper block or the left block and the previous block). If neither is available, we consider just the previous block. By using the parameter $N$ obtained from each method, we choose the proper VLC table for the current block from Table 1.

The difference of correctness between two reference blocks of the current frame and the block of the previous frame is large when the QP is low. Therefore, the weighting values are determined by considering QP. In order to evaluate the influence of weighting values considering QP, we experiment 100 frames from “Foreman” sequence in QCIF format (176×144) and QP is set to 20. The frame coding structure is IPPP…P. The search range is ±16 [5].

**Table 4.** Improvement of correctness and bit rate

| $\beta$ | Improvement of correctness (%) | Number of saving bits (bits) |
|---|---|---|
| 0.60 | +8.36 | 16,664 |
| 0.67 | +9.23 | 17,696 |
| 0.71 | +9.67 | 17,720 |
| 0.75 | +9.90 | 17,512 |
| 0.78 | +9.97 | 17,232 |

Table 4 represents improvement of correctness and saving bits depending on weighting values. As you can see in Table 4, when $\beta$ is 0.71, coding efficiency is the best. Therefore, if we select the proper weighting value for each sequence, we can reduce the amount of encoding bits. We set weighting values which show the best coding efficiency for several test sequences by the data driven approach.

In order to derive general weighting values for several test sequences, we investigate the best weighting value. Table 5 shows general weighting values considering QP.

**Table 5.** General weighting values considering QP

| QP | $\alpha$ | $\beta$ |
|---|---|---|
| 0 ~ 16 | 0.125 | 0.75 |
| 17 ~ 28 | 0.145 | 0.71 |
| 29 ~ 51 | 0.165 | 0.67 |

## 4. EXPERIMENTAL RESULTS

In order to evaluate performance of the proposed algorithm, we use first 100 frames from five test video sequences (Foreman, News, Container, Silent, and Carphone) in QCIF format (176×144). JM 9.5 is used to conduct the experiments [9]. In the motion estimation, one reference frame is enabled with the maximum search range ±16. The frame coding structure is IPPP…P. We have tested for various quantization parameters (16, 20, 24, and 28) [10].

**Table 6.** VLC Table prediction correctness

| Test Sequence | QP | Correctness (%) | |
|---|---|---|---|
| | | H.264 | Proposed scheme |
| Foreman | 16 | 47.32 | 55.27 |
| | 20 | 50.32 | 60.09 |
| | 24 | 56.47 | 65.38 |
| | 28 | 64.31 | 70.92 |
| News | 16 | 51.37 | 58.78 |
| | 20 | 52.76 | 60.45 |
| | 24 | 55.19 | 63.34 |
| | 28 | 59.12 | 66.09 |
| Container | 16 | 49.99 | 57.91 |
| | 20 | 52.27 | 61.97 |
| | 24 | 54.45 | 62.99 |
| | 28 | 56.63 | 63.62 |
| Silent | 16 | 51.58 | 56.55 |
| | 20 | 54.93 | 59.92 |
| | 24 | 59.62 | 63.46 |
| | 28 | 63.12 | 67.09 |
| Carphone | 16 | 52.34 | 59.09 |
| | 20 | 55.51 | 63.71 |
| | 24 | 59.81 | 67.50 |
| | 28 | 64.48 | 70.97 |

Table 6 compares correctness of the proposed and the H.264 algorithm. As you can see, the proposed method provides higher correctness of VLC table prediction. For "Foreman" sequence, we have increased correctness of VLC table prediction up to 9.77 % compared to the H.264.

Figure 6, Fig. 7, and Fig. 8 illustrate correctness curves for sequences "Foreman", "News", and "Silent". Figure 6 shows the best case and Fig. 7 represents the middle case and Fig. 8 shows the worst case among the results. From figures, we observe that correctness curves of the proposed algorithm are located upper than correctness curves of H.264.

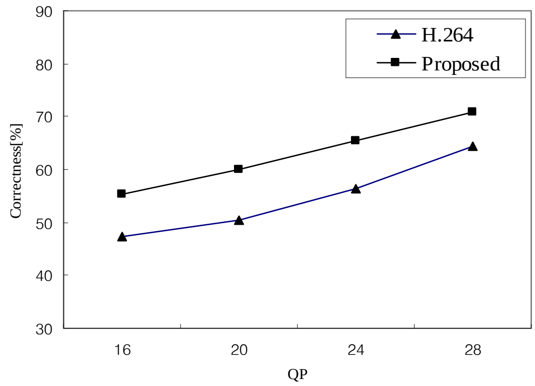



**Fig. 6.** Correctness curve for "Foreman" sequence

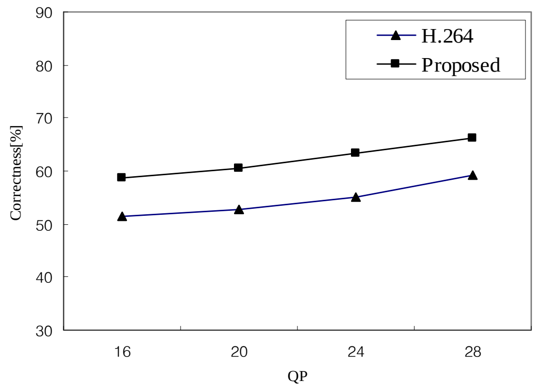



**Fig. 7.** Correctness curve for "News" sequence

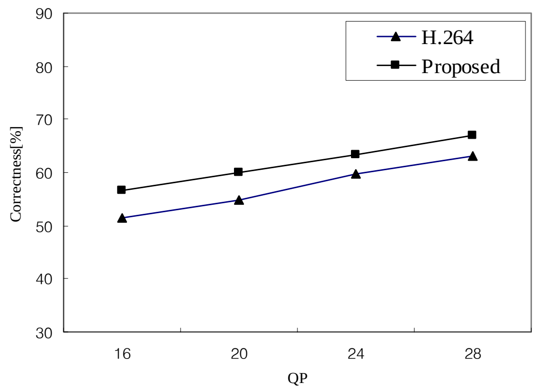



**Fig. 8.** Correctness curve for "Silent" sequence

**Table 7.** Comparison of bit rates

| Test Sequence | QP | Bit Rate (kbps) | |
|---|---|---|---|
| | | H.264 | Proposed scheme |
| Foreman | 16 | 842.13 | 833.52 |
| | 20 | 451.64 | 446.33 |
| | 24 | 233.73 | 231.47 |
| | 28 | 115.52 | 114.84 |
| News | 16 | 312.42 | 309.68 |
| | 20 | 199.44 | 197.60 |
| | 24 | 125.07 | 124.03 |
| | 28 | 76.07 | 75.65 |
| Container | 16 | 343.25 | 339.40 |
| | 20 | 174.80 | 172.69 |
| | 24 | 84.01 | 83.17 |
| | 28 | 40.19 | 39.87 |
| Silent | 16 | 350.01 | 348.15 |
| | 20 | 215.77 | 214.73 |
| | 24 | 134.51 | 134.14 |
| | 28 | 82.85 | 82.69 |
| Carphone | 16 | 590.37 | 585.80 |
| | 20 | 340.48 | 337.33 |
| | 24 | 194.82 | 193.21 |
| | 28 | 105.72 | 105.14 |

As you can see in Table 7, the proposed algorithm improves coding efficiency over the H.264 CAVLC scheme. For the "Container" sequence, we have reduced bit rates by up to 1.21 %. We observe that the increase of correctness for VLC table prediction provide better performance improvement with the proposed algorithm.

## 5. CONCLUSION

In this paper, we have proposed a new prediction scheme for estimating a proper VLC table in H.264. The proposed algorithm uses two methods to find the proper VLC table. First method uses multiple reference blocks; upper, left, and previous block. Second method uses weighting factors. Weighting values are determined by considering QP and correlation between the current and reference blocks. Experimental results show that the proposed algorithm increases correctness of VLC table prediction by 7.18 % and reduces bit rates by 0.78 % on average, compared to H.264.


## ACKNOWLEDGMENTS

This work was supported by the Ministry of Information and Communication (MIC) through the Realistic Broadcasting Research Center (RBRC) at Gwangju Institute of Science and Technology (GIST).


## REFERENCES


1. ITU-T Rec. H.264 | ISO/IEC 14496-10 AVC.: Draft ITU-T Recommendation and Final Draft International Standard of Joint Video Specification. JVT Document JVT-G050 (2003)
2. Marpe, D., Schwarz, H., Wiegand, T.: Context-based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard. IEEE Transactions on Circuits and Systems for Video Technology. Volume 13 (2003) 620-636
3. Chang, H.-C., Lin, C.-C., Guo, J.-J.: A Novel Low-Cost High-Performance VLSI Architecture for MPEG-4 AVC/H.264 CAVLC Decoding. IEEE Interna-tional Symposium on Circuits and Systems. Volume 6 (2005) 6110-6113
4. Sullivan, G., Wiegand, T.: Video Compression-From Concepts to the H.264/AVC Standard. Proceedings of the IEEE. Volume 93 (2005)
5. Amer, I., Badawy, W., Jullien, G.: Towards MPEG-4 PART 10 System on Chip: A VLSI Prototype for Context-based Adaptive Variable Length Coding (CAVLC). IEEE Workshop on Signal Processing Systems. (2004) 275-279
6. Wiegand, T., Sullivan, G., Bjontegaard, G., Luthra, A.: Overview of the H.264/AVC Video Coding Standard. IEEE Transaction on Circuit and System for Video Technology. Volume 13 (2003)
7. Richardson, I. E. G.: H.264 and MPEG-4 Video Compression. Wiley (2003) 201-207
8. Bjontegaard, G., Lillevold, K.: Context-adaptive VLC (CVLC) coding of coefficients. JVT Document JVT-C028., Fairfax, Virginia (2002)
9. Sullivan, G., Bjontegaard, G.: Recommended Simulation Common Conditions for H.26L Coding Efficiency Experiments on Low-Resolution Progressivescan Source Material. ITU-T Q.6/16, Doc. VCEG-N81 (2001)
10. JVT Reference Software JM 9.5, available online at: http://iphome.hhi.de/suehring/tml/download/jm_old/jm95.zip