Improved Context-Based Adaptive Binary Arithmetic Coding over H.264/AVC for Lossless Depth Map Coding

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

***Abstract*—The depth map, which represents three-dimensional (3D) information, is used to synthesize virtual views in the depth image-based rendering (DIBR) method. Since the quality of synthesized virtual views highly depends on the quality of depth map, we encode the depth map under the lossless coding mode. The original context-based adaptive binary arithmetic coding (CABAC) that was originally designed for lossy texture coding cannot provide the best coding performance for lossless depth map coding due to the statistical differences of residual data in lossy and lossless depth map coding. In this letter, we propose an enhanced CABAC coding mechanism for lossless depth map coding based on the statistics of residual data. Experimental results show that the proposed CABAC method provides approximately 4% bit saving compared to the original CABAC in H.264/AVC.**

***Index Terms*—CABAC, H.264/AVC, lossless, depth map.**

I. Introduction

THREE-DIMENSIONAL video (3DV) and free viewpoint video (FVV) technologies have been studied to expand the user's sensation beyond what is offered by the traditional media. In order to efficiently support 3-D scene representations for 3DV and FVV, the new standard for multiview video coding (MVC) was developed and such multiview video is widely used in various 3DV and FVV systems [1].

In 3DV [2] and FVV systems, the main difficulty in the deployment of 3DV services appears to be the large bandwidth requirements associated with the transport of multiview video. Therefore, recently, instead of using a large number of views, multiview video plus depth (MVD) system [3] is used because the coding of the depth map consumes a small overhead, typically about 10–20%, on the video bitrate [4].

In order to synthesize virtual views, the depth image-based rendering (DIBR) technique using the video and the corresponding depth map images [5] is used. In DIBR, the accuracy of depth map directly affects the quality of synthesized virtual views. As a result, many works have focused on the acquisition of accurate depth map [6].

Manuscript received March 24, 2010; revised July 05, 2010; accepted July 06, 2010. Date of publication July 15, 2010; date of current version August 09, 2010. This work was supported by the MKE (The Ministry of Knowledge Economy), Korea, under the ITRC(Information Technology Research Center) support program supervised by the NIPA(National IT Industry Promotion Agency) (NIPA-2010-(C1090-1011-0003)). The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Panos Nasiopoulos.

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 letter are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/LSP.2010.2059014

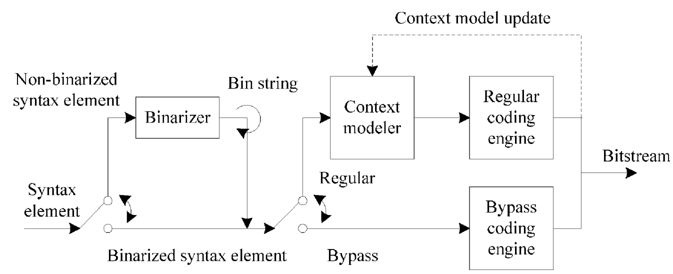

Fig. 1. Encoding structure of CABAC for a single syntax element.

Before 3-D videos are rendered at the receiver side, we transmit regular multiview video as well as their corresponding depths. In MVD system, the depths are encoded by lossy coding and then transmitted. However, since the quality of depth is highly related to the rendering quality of virtual view, the depth map should be encoded by lossless coding.

In lossy depth map coding, residuals represent the quantized transform coefficients. On the contrary, in lossless depth map coding, residuals are not the quantized transform coefficients, but rather the differential pixel values between the original and predicted pixel values. Therefore, the statistics of residuals in lossy and lossless depth map coding are quite different.

In general, the depth map is encoded using a context-based adaptive binary arithmetic coding (CABAC) [7] method. However, since CABAC in the H.264/AVC standard [8] was originally designed for lossy texture coding, it was unable to provide the optimum coding performance for lossless depth map coding. For that reason, we have tried to improve the coding performance of CABAC for lossless depth map coding.

II. Context-Based Adaptive Binary Arithmetic Coding

The encoding process of CABAC consists of four coding steps: binarization, context modeling, binary arithmetic coding, and probability update. The block diagram for encoding a single syntax element in CABAC is depicted in Fig. 1.

In the first step, a given nonbinary valued syntax element is uniquely mapped to a binary sequence (bin string). When a binary valued syntax element is given, the first step is bypassed. In the regular coding mode, each binary value (bin) of the binary sequence enters the context modeling stage, where a probability model is selected based on the previously encoded syntax elements. Then, the arithmetic coding engine encodes each binary value with its associated probability model. Finally, the selected context model is updated according to the actual coded a binary value.

Fig. 2 illustrates the CABAC encoding structure for a 4×4 subblock of the quantized transform coefficients. First, the syntax element *coded_block_flag* is encoded with a one bit

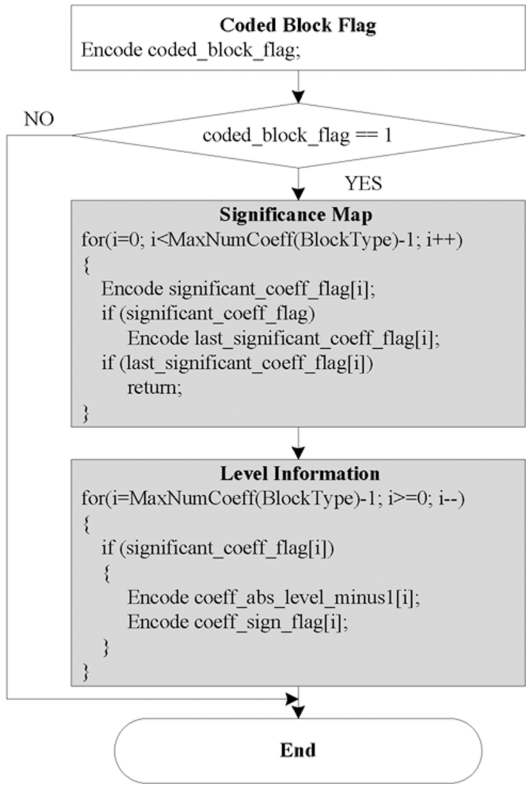

Fig. 2. Encoding structure of CABAC for residual data coding.

symbol for each subblock. If *coded_block_flag* is zero, since no further information is encoded for the subblock, the coding process for the subblock is terminated. Otherwise, significance map and level information are sequentially encoded.

If *coded_block_flag* indicates that a subblock has the significant coefficients, a binary valued significance map is encoded. For each coefficient, a one bit symbol *significant_coeff_flag* is encoded in scanning order. If *significant_coeff_flag* is one, a further one bit symbol *last_significant_coeff_flag* is encoded. This syntax element indicates if the current significant coefficient is the last coefficient inside the subblock or not.

After the encoded significance map determines the locations of all significant coefficients inside a subblock, the values of the significant coefficients are encoded by using two syntax elements: *coeff_abs_level_minus1* and *coeff_sign_flag*. The syntax element *coeff_sign_flag* is encoded by a one bit symbol while the *Unary/0th order Exp-Golomb* (UEG0) binarization method is used for encoding the values of *coeff_abs_level_minus1* representing the absolute value of the level minus 1. The values of the significant coefficients are encoded in reverse scanning order.

III. Proposed Method

In this section, we describe a new CABAC scheme for lossless depth map coding based on the statistical characteristics of residual sample values. In Fig. 2, the gray-shaded processes are modified in the proposed method for lossless depth map coding.

A. Significance Map Coding

In lossy depth map coding, the occurrence probability of a significant coefficient is likely to be decrease as the scanning position increases because residual data are the quantized transform coefficients. Therefore, the significant coefficient tends to be located at the earlier scanning position. In this case,

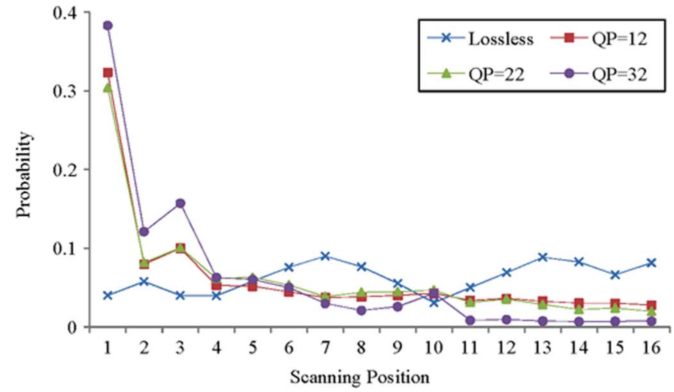

Fig. 3. Occurrence probability distribution of nonzero coefficients according to the scanning position (Breakdancers_Depth_View4).

Scanning position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Transform coefficient level	9	0	-5	3	0	-7	4	0	8	-11	-6	0	3	1	0	0
significant_coeff_flag	1	0	1	1	0	1	1	0	1	1	1	0	1	1		
last_significant_coeff_flag	0		0	0		0	0		0	0	0		0	1		

Fig. 4. Example of significance map coding for lossy coding.

Scanning position	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
Differential pixel value	9	0	-5	3	0	-7	4	0	8	-11	-6	0	3	1	0	0
significant_coeff_flag	1	0	1	1	0	1	1	0	1	1	1	0	1	1	0	0

Fig. 5. Example of significance map coding for lossless coding.

last_significant_coeff_flag plays an important role in the early termination of significance map coding.

In lossless depth map coding, residual data do not represent the quantized transform coefficients, but rather the differential pixel values. Therefore, the statistics of residual data from lossy and lossless depth map coding are quite different. In lossless depth map coding, the occurrence probability of a significant coefficient is independent of the scanning position, as shown in Fig. 3. From extensive experiments on lossless depth map coding, we observed that significance map coding is likely to be terminated at the end of the scanning position. In this case, it is meaningless to encode *last_significant_coeff_flag* indicating the position of the last significant coefficient. Therefore, we remove *last_significant_coeff_flag* coding process and directly encode all *significant_coeff_flags* at the scanning position from 1 to 16.

Fig. 4 represents an example of significance map coding in CABAC for lossy coding when the scanning position of the last significant coefficient is 14; the gray-shaded *significant_coeff_flag* and *last_significant_coeff_flag* are encoded in significance map coding.

However, since we removed *last_significant_coeff_flag* in lossless coding, *significant_coeff_flag* is unconditionally encoded up to the last scanning position. Fig. 5 indicates an example of significance map coding for lossless coding. All gray-shaded *significant_coeff_flags* are encoded in the proposed significance map coding, as shown in Fig. 5.

B. Binarization for Level Coding

In level coding, the UEG0 binarization method is applied to the absolute values of the quantized transform coefficients (*abs_level*). UEG0 binarization is specified by the cutoff value $S = 14$ for the *truncated unary* (TU) prefix part and the order $k = 0$ for the *Exp-Golomb* (EGk) suffix part. The structure of UEG0 binarization is only efficient for small *abs_level*; for larger *abs_level*, adaptive modeling has limited the functionality.

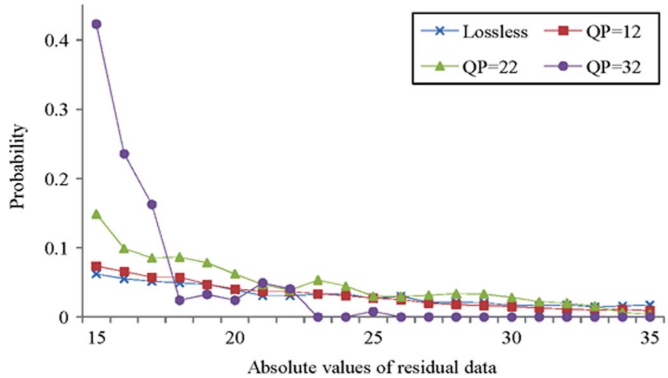

Fig. 6. Occurrence probability distribution of residual data between 15 to 35 (Breakdacners_Depth_View4).

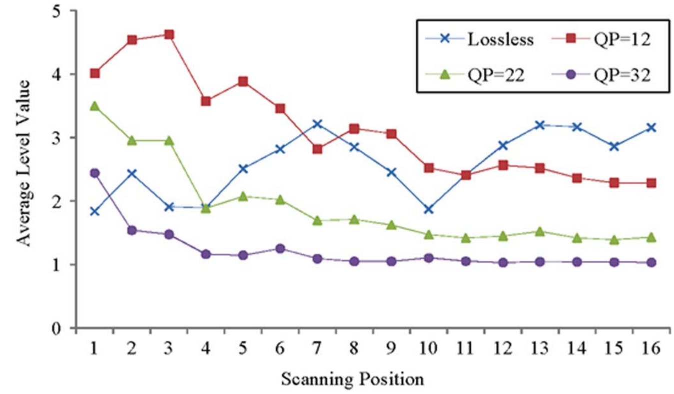

Fig. 7. Distribution of average absolute level values according to the scanning position (Beer_garden_Depth_View5).

TABLE I
BLOCK TYPES WITH NUMBER OF COEFFICIENTS AND ASSOCIATED CONTEXT CATEGORY (*CTX_CAT*)

Block Type	MaxNumCoeff	Context Category
Luma DC block for Intra16×16	16	0
Luma AC block for Intra16×16	15	1
Luma block for Intra4×4 Luma block for Inter	16	2
U- and V-Chroma DC block	4	3
U- and V-Chroma AC block	15	4

TABLE II
AVERAGE NUMBER OF CONSECUTIVE TRAILING ONES

Sequence	QP			
	0 (Lossless)	12	22	32
Breakdancers	1.439709	0.847259	1.029962	0.950915
Ballet	1.376442	0.747477	0.854036	0.633731
Beer_garden	2.921325	0.702819	0.866761	0.324139
Mobile	0.825232	0.302705	0.220352	0.317946
Newspaper	1.436985	0.382456	0.623383	0.477589

The statistics of the differential pixel values in lossless coding is different from the statistics of *abs_level* in lossy coding. In lossy coding, the occurrence probability of *abs_level* is highly skewed on small *abs_level*. However, in lossless coding, the distribution of the differential pixel values is quite wide, as shown in Fig. 6. Therefore, UEG0 binarization designed for lossy coding is not appropriate for lossless coding.

In order to encode the large differential pixel values efficiently, theoretically, the larger cutoff value for the TU prefix part in UEG0 binarization is required. UEG0 binarization with a large cutoff value is convergent to unary binarization. In unary binarization, for each unsigned integer differential pixel value $x \geq 0$, the unary codeword consists of x "1" bits plus a terminating "0" bit.

C. Context Modeling

Each model can be identified by the unique context index because the entity of probability models used in CABAC can be arranged in a linear fashion. The context index for syntax element of residual data (γ_S) is specified by

$$\gamma_S = \Gamma_S + \Delta_S(\text{ctx\_cat}) + \chi_S \tag{1}$$

where Γ_S represents the *context index offset*, which is defined as the lower value of the range of a given syntax element S, and χ_S denotes the *context index increment* of a given syntax element S. $\Delta_S(\text{ctx\_cat})$ is the *context category* (*ctx_cat*) dependent offset Δ_S and is determined based on the block types, as shown in Table I.

In lossless coding, since transform and quantization are not performed, all coefficients within a subblock do not separate from dc and ac coefficients. As a result, block types corresponding to $\text{context category} = 2$ are only used in lossless coding. Therefore, we can fix the context category value in the context modeling stage for *significant_coeff_flag* and *coeff_abs_level_minus1*.

In lossy coding, the context index increment for *significant_coeff_flag* (χ_{SIG}) was designed based on the expectation that the occurrence probability of a significant coefficient is likely to decrease as the scanning position increases, as shown in Fig. 3. Therefore, χ_{SIG} is determined according to the scanning position.

However, in lossless coding, the occurrence probability of a significant coefficient is independent of the scanning position, as shown in Fig. 3. Therefore, in lossless coding, we do not consider χ_{SIG} in the context index decision for *significant_coeff_flag* (γ_{SIG}). As a result, γ_{SIG} is given by

$$\gamma_{\text{SIG}} = \Gamma_{\text{SIG}} + \Delta_{\text{SIG}}(\text{ctx\_cat} = 2). \tag{2}$$

In order to determine the context index increment for *coeff_abs_level_minus1* (χ_{Coeff}), we use two adequately designed sets of context models: one for the first bin (bin $\text{index} = 0$) and another one for the remaining bins (bin indices 1 to 13) of the UEG0 prefix part. In lossy coding, at the end of the scanning position, *abs_level* is likely to observe the occurrence of successive ± 1 called trailing ones. In addition, *abs_level* is going to be larger as the scanning position decreases, as shown in Fig. 7. Based on these observations, χ_{Coeff} is determined according to the accumulated number of encoded trailing ones (*NumT1(i)*) and the accumulated number of encoded levels with absolute value greater than one (*NumLgt1(i)*), where i means the scanning position.

Table II represents the average number of consecutive trailing ones according to the quantization parameter (*QP*). In lossless coding, the number of consecutive trailing ones is larger than that in lossy coding. Moreover, the average absolute differential pixel value in lossless coding is independent of the scanning position, as shown in Fig. 7. Based on these

TABLE III
ENCODING PARAMETERS

ProfileIDC	244 (High 4:4:4)
Coding Structure	IPPP ··· P
QPISlice and *QPPSlice*	0 (Lossless)
SymbolMode	1 (CABAC)
QPPrimeYZeroTransformBypassFlag	1

observations, we designed a new context modeling method for *coeff_abs_level_minus1* in lossless coding.

Since the number of consecutive trailing ones turns out to be relatively larger than that in lossy coding, we modify the upper bound of *NumT1(i)* from 3 to 5. Therefore, for encoding the first bin of the UEG0 prefix part, the corresponding context index $(\gamma_{\text{Coeff}}(i, \text{bin\_index} = 0))$ is determined by

$$\begin{aligned}\gamma_{\text{Coeff}}&(i, \text{bin\_index} = 0)\\ &= \Gamma_{\text{Coeff}} + \Delta_{\text{Coeff}}(\text{ctx\_cat} = 2)\\ &\quad + \chi_{\text{Coeff}}(i, \text{bin\_index} = 0)\end{aligned} \tag{3}$$

$$\begin{aligned}\chi_{\text{Coeff}}&(i, \text{bin\_index} = 0)\\ &= \begin{cases} 6, & \text{if NumLgt1}(i)>1\\ \min(5, \text{NumT1}(i)), & \text{otherwise.}\end{cases}\end{aligned} \tag{4}$$

In lossy coding, decision of the context index increment for the remaining bins of the UEG0 prefix part $(\chi_{\text{Coeff}}(i, \text{bin\_index}))$ is based on the expectation that *abs_level* is likely to increase at the low frequencies. However, the differential pixel value in lossless coding is independent of the scanning position, as shown in Fig. 7. Therefore, we do not use $\chi_{\text{Coeff}}(i, \text{bin\_index})$ in the context index decision for the remaining bins $(\gamma_{\text{Coeff}}(i, \text{bin\_index}))$. Finally, $\gamma_{\text{Coeff}}(i, \text{bin\_index})$ is calculated by

$$\gamma_{\text{Coeff}}(i, \text{bin\_index}) = \Gamma_{\text{Coeff}} + \Delta_{\text{Coeff}}(\text{ctx\_cat} = 2). \tag{5}$$

IV. EXPERIMENTAL RESULTS AND ANALYSIS

In order to examine efficiency of the proposed method, we performed experiments on several depth map sequences. We implemented our proposed method in the H.264/AVC reference software version JM 13.2 [9]. Table III shows the encoding parameters for the reference software.

The proposed method consists of three parts.

1) Method I: Proposed significance map coding.
2) Method II: Proposed binarization for level coding + proposed context modeling.
3) Method III: Method I + Method II.

Our proposed method was applied to lossless depth map coding by modifying the semantics and decoding processes, but not requiring any changes in the syntax elements of the H.264/AVC standard. In our experiments, we compared coding performance with bit-rate percentage differences:

$$\Delta\text{Saving Bits}(\%) = \frac{\text{Bitrate}_{\text{H.264/AVC}} - \text{Bitrate}_{\text{Method}}}{\text{Bitrate}_{\text{H.264/AVC}}} \times 100. \tag{6}$$

In Table IV, we indicate that the proposed method improves coding performance by approximately 4% bit saving in lossless depth map coding, compared to the original CABAC in the H.264/AVC standard.

TABLE IV
COMPARISON OF SAVING BITS

Sequence (Original Size (bits))	Method	Total Bits (bits)	Saving Bits (%)
Breakdancers,	Original	29855552	0
1024×768,	Method I	29122856	2.45414
100 frames	Method II	29540424	1.05551
(629145600)	Method III	**28815952**	**3.48210**
Ballet,	Original	28193248	0
1024×768,	Method I	27612040	2.06151
100 frames	Method II	27755216	1.55368
(629145600)	Method III	**27180568**	**3.59192**
Beer_garden,	Original	127650800	0
1920×1080,	Method I	122000584	4.42631
100 frames	Method II	126896960	0.59055
(1658880000)	Method III	**121264240**	**5.00315**
Mobile,	Original	2891920	0
720×540,	Method I	2861352	1.05701
100 frames	Method II	2795824	3.32291
(311040000)	Method III	**2757296**	**4.65518**
Newspaper,	Original	30917872	0
1024×768,	Method I	30486536	1.39510
100 frames	Method II	30470920	1.44561
(629145600)	Method III	**30042152**	**2.83241**

V. CONCLUSION

In this letter, we propose an improved context-based adaptive binary arithmetic coding (CABAC) for lossless depth map coding. Considering the statistical differences in residual data between lossy and lossless depth map coding, we modified the CABAC encoding mechanism based on the modified significance map coding, binarization for differential sample value, and context modeling. Experimental results demonstrate approximately 4% bit saving improvement compared to the current H.264/AVC CABAC.

REFERENCES

[1] A. Smolic, K. Mueller, P. Merkle, C. Fehn, P. Kauff, P. Eisert, and T. Wiegand, "3D video and free viewpoint video-technologies, applications and MPEG standards," in *ICME*, Jul. 2006, pp. 2161–2164.

[2] A. Redert, M. O. Beeck, C. Fehn, W. IJsselsteijn, M. Pollefeys, L. V. Gool, E. Ofek, I. Sexton, and P. Surman, "Attest: Advanced three-dimensional television system technologies," in *Proc. Int. Symp. 3D Data Processing Visualization and Transmission*, Nov. 2002, pp. 313–319.

[3] "Applications and requirements on 3D video coding," Feb. 2009, MPEG Video Group, MPEG Doc. N10358.

[4] H. M. Ozaktas and L. Onural*, Three-Dimensional Television Capture, Transmission, Display*. Berlin, Germany: Springer, 2008.

[5] C. Fehn, "Depth-image-based rendering (DIBR), compression and transmission for a new approach on 3-D TV," in *Proc. SPIE Conf. Stereoscopic Displays and Virtual Reality Systems*, Jan. 2004, vol. 5291, pp. 93–104.

[6] C. L. Zitnick, S. B. Kang, M. Uyttendaele, S. Winder, and R. Szeliski, "High-quality video view interpolation using a layered representation," in *Proc. SIGGRAPH'04*, Aug. 2004, pp. 600–608.

[7] D. Marpe, H. Schwarz, and T. Wiegand, "Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 13, no. 7, pp. 620–636, Jul. 2003.

[8] T. Wiegand, G. J. Sullivan, G. Bjontegaard, 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.

[9] [Online]. Available: http://iphome/hhi.de/shehring/tml/download/old_jm/jm13.2.zipJoint Video Team, Reference Software Ver. 13.2