Evaluation of Beyond-HEVC Entropy Coding Methods for DCT Transform Coefficients

Han Zhang [1,2], Li Song [1,2], Xiaokang Yang [1,2], Zhengyi Luo [3]

[1] *Institute of Image Communication and Network Engineering, Shanghai Jiao Tong University*
[2] *Cooperative Medianet Innovation Center, Shanghai, China*
[3] *School of Electronics and Information Engineering, Shanghai University of Electric Power*
[1,2] {zetallica, song_li, xkyang}@sjtu.edu.cn, [3] lzy@shiep.edu.cn

***Abstract*—Entropy coding, which acts as one of the most important compression tools in video coding standard, had been improved step by step for HEVC. There are also several advanced methods which provide better performance than current solutions of HEVC proposed during the standardization of HEVC. However, these methods are all tested in different conditions. Comprehensive evaluation of these advanced methods under a common scenario is desired to indicate where the potential improvement of entropy coding may come from for next generation video codec. In this paper, we first introduce several advanced entropy coding methods for DCT transform coefficients, which aim to improve CABAC performance from two aspects – context modeling and probability updating. Then some modifications based on these original ones are presented. Comprehensive comparison of these methods is conducted under common test conditions. Besides, some combined methods of these two aspects are also tested. Experimental results show that all individual approaches can achieve coding gain and two new combined methods can reduce the BD-Rate up to 1.7%, 1.2% and 1.0% on common test sequences and 1.4%, 1.0% and 1.1% on 4K sequences under all intra, random access and low delay configurations, respectively.**

***Index Terms*—entropy coding, transform coefficient, CABAC, context modeling, probability updating, HEVC**

I. INTRODUCTION

With the growing popularity of Ultra High-Definition video (UHD, 3840×2160 resolution) and the development of video applications with higher quality and resolutions, an increasing challenge is posed in video transmission and storage. Thus, next-generation video coding technologies superior to those of HEVC [1] still deserve continuous efforts. Preparatory work of next generation video coding standard carried out by the Joint Video Exploration Team (JVET) of ITU-T Video Coding Experts Group (VCEG) and ISO/IEC Moving Picture Experts Group (MPEC) is underway. The first version of test model Joint Exploration Test Model 1 (JEM1) is available, where Context-Based Adaptive Binary Arithmetic Coding (CABAC) [2] inherited from HEVC is still used.

As a new form of entropy coding technology, CABAC was first introduced in H.264/AVC and became the only entropy coding scheme in HEVC due to its superior performance. However, the data dependencies used to improve CABAC coding efficiency make it difficult for parallelization and often lead to a throughput bottleneck. Thus, both coding efficiency and throughput improvement techniques were investigated during the standardization of HEVC [3]. Syntax elements related to transform coefficients account for a significant portion of the total number of coded bits, thus CABAC methods for coefficients have to be carefully designed to balance the overall performance [4]. Because of the tradeoff between coding efficiency and throughput/complexity, some methods providing better performance than current solutions of HEVC were ignored. However, at current stage of next-generation video coding standard, coding efficiency should be the primary consideration. Thus, these advanced methods can be regarded as the source of potential improvement of entropy coding for next-generation video codec. While some of these advanced methods have been adopted by JEM, there are still many methods were tested in other conditions. Thus, comprehensive evaluation of these methods under common scenario is still desired to show the potential improvement roadmap.

In this paper, we first introduce several advanced entropy coding methods for DCT transform coefficients, which aim to improve CABAC performance from two aspects – context modeling and probability updating. More specifically, methods utilizing local neighbor template to determine context model index of syntax elements *significant_coeff_flag, coeff_abs_level_greater1_flag* and *coeff_abs_level_greater2_flag* are introduced. Meanwhile we also choose some techniques which can give more accurate probability estimation for arithmetic coding engine of CABAC. Additionally, some modifications based on these original methods are also presented. Then, comprehensive comparison of these methods is conducted under common test conditions. Besides, some combined methods of these two aspects are also tested. Experimental results show all individual approaches perform better than current solutions of HEVC and two new combined methods can achieve remarkable coding gain at the cost of increased complexity and memory consumption.

II. REVIEW OF TRANSFORM COEFFICIENT CODING WITH CABAC IN HEVC

The regular mode of CABAC consists of three elementary steps. A non-binary syntax element is firstly mapped to an unique binary string at the binarization stage. Secondly, context models along with the related probability states for current coded bins are determined by predefined modeling

978-1-5090-5316-2/16/$31.00 © 2016 IEEE

r les at the context modeling stage. After c rrent bin is coded, the probability state is pdated sing a probability transition table with 64 states. The binary arithmetic coding engine – M coder works as the final stage[5]. In M coder, the interval s bdivision step of arithmetic coding is replaced by a look- p table approach.

In HEVC, transform coefficients are processed in 4×4 s bblock, also referred to as coefficient gro p (CG). The following three syntax elements are coded with the CABAC reg lar mode [6]: *significant_coeff_flag* (indicating whether a coefficient is non-zero), *coeff_abs_level_greater1_flag* (indicating whether the absol te val e of a coefficient is greater than 1), *coeff_abs_level_greater2_flag* (indicating whether the absol te val e of a coefficient is greater than 2).

1) significant_coeff_flag (SIG): For a 4×4 TB (Transform Block), the context model index depends on the position of c rrent scanning order within the TB. For TB larger than 4×4, context model selection is based on whether there are non-zero coefficients in neighboring right and lower CGs and on the scanning position within the c rrent CG.
2) coeff_abs_level_greater1_flag (ALG1): Firstly, a context set is selected from six sets. Then the context model within each set is selected from fo r models according to the following form lation:

$$ctxInc = \begin{cases} 0 & NumG1 > 0 \\ 1 + min\left(2, NumT1\right) & otherwise \end{cases} \quad (1)$$

where N mT1 denotes the acc m lated n mber of absol te levels eq al to 1 along the reverse scanning order within the CG and N mG1 denotes the acc m lated n mber of levels greater than 1 .
3) coeff_abs_level_greater2_flag (ALG2): The same context modeling approach is applied as ALG1 except each context set contains only one model.

III. Advanced transform coefficient entropy coding techniques

In this section, we present several advanced coefficient related CABAC methods proposed d ring the standardization of HEVC firstly. Then some modified approaches based on the original proposed ones are also introd ced. These methods can be divided into two categories – context modeling and probability pdating. The symbols SIG, ALG1 and ALG2 described in section II will be sed in the following as well.

A. Context Modeling

As mentioned in section II, the context index of SIG in HEVC is determined based on the c rrent coefficient's position within a TB. In order to exploit the correlation between adjacent pixels, an index selection method based on previo s coded bins in a local template is presented in [7]. The template can be depicted in Fig.1, where x denotes the c rrent scan position, x_i with $i\epsilon[0, 4]$ denotes the neighbo rs covered by the template. This approach tilizing the s m of absol te levels of neighboring coefficients and is good for generalization.

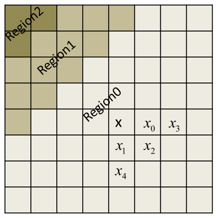


Fig. 1. Template sed in coefficient context modeling

Recently, Chen *et al.* proposed a more sophisticated approach where the context indices of SIG, ALG1 and ALG2 are determined by the s m of corresponding syntax element val e lying in the template as depicted in Fig.1 [8]. The context index selection process can be expressed as follows:

$$ctx_{SIG} = min\left(numSig, 5\right) \quad (2)$$

$$ctx_{ALG1} = min\left(sumGrt1, 4\right) + 1 \quad (3)$$

$$ctx_{ALG2} = min\left(sumGrt2, 4\right) + 1 \quad (4)$$

where *numSig* denotes the n mber of non-zero coefficients in the local template, *sumGrt1* and *sumGrt2* refer to the n mber of coefficients with absol te level greater than 1 and 2, respectively.

The DCT transform coefficients have different statistical behaviors at different freq ency zones. To f t re exploit these statistical characteristics, different context models are sed at different freq ency zones. To this end, one TU may be splitted into p to three regions with fixed splitting method as ill strated in Fig.1 for SIG context model selection. While regions for ALG1 and ALG2 are set as Fig.2(b). An offset is defined to disting ish different regions.

$$OF_{SIG} = \begin{cases} f\left(x, y, 6, 2\right) + f\left(x, y, 6, 5\right) & luma \\ f\left(x, y, 6, 2\right) & chroma \end{cases} \quad (5)$$

The additional offset of ALG1 and ALG2 applies to the l ma component only.

$$OF_{ALG} = f\left(x, y, 5, 3\right) + f\left(x, y, 5, 10\right) \quad (6)$$

where

$$f\left(x, y, n, t\right) = \begin{cases} n & \text{if } x + y < t \\ 0 & \text{if } x + y \geq t \end{cases} \quad (7)$$

Th s the final context indexes of these three syntax elements are calc lated as:

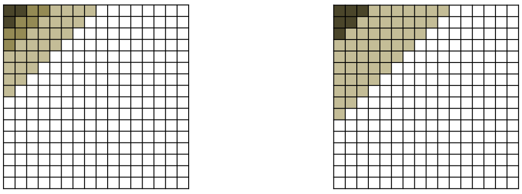
$$ctxidx = ctx + OF \quad (8)$$

The above method is *QC-CTX*.

A simpler scheme is proposed in *I383* [9], where context selection of ALG1 and ALG2 re ses the n mber of significant coefficients *numSig* to simplify the context model selection process. Additionally, ALG1 and ALG2 se the same context model for one coefficient.

$$ctx_{ALG1} = ctx_{ALG2} = min\left(numSig, 5\right) \quad (9)$$

And the context selection of SIG is the same as *QC-CTX*.



(a) (b)

Fig. 2. Additional context model index offset. (a) splitting method for SIG. (b) splitting method for ALG1 and ALG2.

In order to make the best use of different statistical characteristics of transform coefficients for higher resolution videos. For SIG context selection, one TU is splitted into up to four regions regardless of TU size as depicted in Fig.2(a). The splitting method for ALG1 and ALG2 remains the same as Fig.2(b). This modified method is called *I383-updating1* in this paper. The offset of SIG for luma TU is modified as:

$$OF_{SIG} = f(x, y, 6, 2) + f(x, y, 6, 4) + f(x, y, 6, 8) \quad (10)$$

To further improve the performance, the number of absolute level greater than 1 *sumGrt1* is directly mapped to a context model index of ALG1 if the number of coefficient with absolute level greater than 1 (*sumGrt1*) in the template is not zero. Otherwise, *numSig* is utilized as in *I383* to derive the context index. Additionally, ALG1 still shares the same context models with ALG2 for context model number reduction. *I383-updating2* is used for short.

B. Probability Updating

As mentioned in section II, a probability transition table with 64 states is used to complete the probability updating step in HEVC. The probability states are associated with model probability values p ranging in the interval $[0.01875,\ 0.5]$. Thus, in order to obtain a more accurate probability value, an intuitional idea is use more states to represent the probability interval. In the JCT-VC proposal *I276* [10], the state number of the probability transition table is extended to 112.

The states used in this table-based method are precalculated using exponential mesh but often associated with quantization error. To avoid the quantization error, a new probability updating technology which calculates the real probability value directly was proposed in [11]:

$$P_{new} = \begin{cases} P_{old} + \left(\left(2^k - P_{old} \right) \gg W_i \right) & input\ '1' \\ P_{old} - \left(P_{old} \gg W_i \right) & input\ '0' \end{cases} \quad (11)$$

where k is the bit precision to represent probability value, *input* is the current coded bin, P_{new} and P_{old} denote the probability after and before updating. The parameter W_i works like a sliding window with size 2^{W_i}, which determines the probability adaptation speed. A small W_i results in a fast adaptation but may be sensitive to random jitter, while a large W_i model converges slowly but the model is stable after reaching the optimal value.

Alshin *et al.* proposed to utilize a two-parameter probability updating model called *mCABAC* with $W_0 = 4$ and $W_1 = 8$ [11], where the updated probability is calculated as average:

$$P_{new} = (P_{0,new} + P_{1,new} + 1) \gg 1 \quad (12)$$

Note that this two-parameter model is only used when the number of probability updating exceeds a threshold. Otherwise a model of short window size is utilized due to the fast adaptation speed.

A context adaptive probability update speed method *QC-WIN* [8] was proposed recently. This approach also takes advantage of the probability updating model in (11), but with a dynamic updating speed, where each context model may be assigned with a different parameter W_i ranging from 4 to 7.

These three methods only improve the probability updating process. The range table, which is still used for interval subdivision of arithmetic coding engine needs to be modified according to the new probability states or probability value.

IV. Experimental Results

To conduct a comprehensive and fair evaluation, all the methods mentioned in Section III have been implemented in the test model HM-14.0. Simulations are performed under common test conditions as described in [12]. Coding efficiency is evaluated using BD-Rate [13] and the complexity is measured in terms of encoding time.

Table I shows all of those individual methods can achieve coding gain on common test sequences (class A – class F). For the context modeling part, more accurate statistic characteristic can be used when utilizing the correlation between adjacent pixels. Thus, these three advanced context modeling methods can make a significant performance improvement. *QC-CTX* has larger coding gain than *I383-u2* because a rice parameter derivation method is also integrated [8]. It can be seen that nearly 0.3% BD-Rate reduction can be achieved by only enlarging the probability transition table due to the more elaborate probability states. The directly calculated probability value without quantization error is more accurate than value mapped from transition table, thus, the other two probability updating methods can achieve high coding gain.

In addition to the individual approaches, the coding gain of combination of these two aspects is remarkable. With the integration of an initialization method proposed in [8] in *QC-CTX+QC-WIN*, a new combined method *QC-CABAC* is obtained. The BD-Rate reduction with *QC-CABAC* can be up to 1.7%, 1.2% and 1.0% under three configurations respectively. Note that the bitrate reduction is obtained at the cost of nearly 30% complexity increase. In addition, we also evaluate the performance of *mCABAC+I383-u2* and *QC-CABAC* on 4K videos as shown in Table II. The coding gain of *mCABAC+I383-u2* is nearly 0.2% better than *QC-CABAC* on average due to the more careful TU splitting method for SIG context model selection.

The memory consumption is evaluated by the number of context models and the number of total bits used to save the probability related tables which are shown in Table III and Table IV. *QC-CTX* consumes the maximum number of context models since the context model of SIG varies with TU

TABLE I
BD-RATE RESULTS ON COMMON TEST SEQUENCES RELATIVE TO HM 14.0

Category	Tool	BD-Rate[%]											
		All Intra				Random Access				Low Delay			
		Y	U	V	Time	Y	U	V	Time	Y	U	V	Time
Context Modeling	QC-CTX[8]	-0.9%	-0.5%	-0.7%	107%	-0.6%	0.0%	-0.1%	104%	-0.2%	0.1%	-0.1%	103%
	I383-u1	-0.5%	-0.2%	-0.3%	107%	-0.3%	0.0%	0.0%	103%	-0.1%	0.1%	-0.1%	106%
	I383-u2	-0.7%	-0.5%	-0.5%	105%	-0.5%	-0.2%	-0.2%	105%	-0.2%	0.1%	-0.1%	105%
Probability Updating	QC-WIN[8]	-0.6%	-0.2%	-0.1%	113%	-0.3%	0.5%	0.4%	115%	-0.1%	0.8%	0.7%	102%
	mCABAC [11]	-0.7%	-1.1%	-0.8%	123%	-0.5%	-0.4%	-0.3%	118%	-0.4%	-0.2%	-0.2%	106%
	I276[10]	-0.4%	-0.4%	-0.3%	101%	-0.2%	-0.2%	0.1%	95%	-0.2%	0.4%	0.2%	97%
Combined Method	QC-CTX +QC-WIN[8]	-1.7%	-0.8%	-0.8%	123%	-1.0%	0.3%	0.2%	109%	-0.5%	0.9%	0.7%	111%
	QC-CABAC [8]	**-1.7%**	-0.8%	-0.8%	136%	**-1.2%**	-0.3%	-0.2%	117%	**-1.0%**	**-0.4%**	**-0.5%**	124%
	mCABAC+ I383-u2	-1.4%	**-1.5%**	**-1.3%**	132%	-1.1%	**-0.7%**	**-0.6%**	126%	-0.7%	**-0.4%**	-0.2%	127%

TABLE II
BD-RATE RESULTS ON 4K VIDEOS

Seqs	Tool	BD-Rate[%]		
		AI	RA	LDB
Traffic Flow	QC-CABAC	-1.0%	-0.8%	-0.8%
	mCABAC+u2	-1.4%	-0.9%	-1.1%
Rushhour	QC-CABAC	-1.5%	-1.3%	-1.3%
	mCABAC+u2	-1.5%	-1.4%	-1.2%
Runners	QC-CABAC	-1.2%	-1.2%	-1.1%
	mCABAC+u2	-1.5%	-1.3%	-1.2%
Campfire Party	QC-CABAC	-0.9%	0.4%	-0.6%
	mCABAC+u2	-1.2%	0.0%	-0.7%
Fountains	QC-CABAC	-1.3%	-1.2%	-0.7%
	mCABAC+u2	-1.6%	-1.3%	-1.4%
Average	QC-CABAC	-1.2%	-0.8%	-0.9%
	mCABAC+u2	**-1.4%**	**-1.0%**	**-1.1%**

[1] *mCABAC+u2* represents *mCABAC + I382-u2*

size. The increased number of *I383-u2* can be explained by the use of both *numSig* and *sumGrt1* in the model generation of ALG1. As for the table size cost, although *QC-WIN* and *mCABAC* do not require a probability transition table, the interval subdivision table is much larger than that in HEVC. In contrast, *I276* extends the transition table from 64 states to 112 states but with reduced size of range table. Thus the memory consumption of these two tables is similar to those of HEVC.

TABLE III
NUMBER OF CONTEXT MODEL

	SIG		ALG1		ALG2		Total
	Y	Cb/Cr	Y	Cb/Cr	Y	Cb/Cr	
HEVC	27	15	16	8	4	2	**72**
QC-CTX	54	12	16	6	0	0	**88**
I383-u1	24	12	19	7	0	0	**62**
I383-u2	24	12	34	12	0	0	**82**

TABLE IV
TABLE SIZE COST (NUMBER OF BITS)

	HEVC	QC-WIN	mCABAC	I276
Transition Table	64x6	0	0	112x7
Range Table	64x4x8	512x64x9	512x64x9	28x8x8
Sum	**2432**	**294912**	**294912**	**2576**

V. CONCLUSION

Comprehensive evaluation of several representative beyond-HEVC entropy coding methods is presented in this paper. These methods aim to improve the coding performance in terms of two aspects – context modeling and probability updating. All of these methods give more accurate probability estimation, which is desired by the binary arithmetic coding engine. Simulation results demonstrate that all individual approaches can achieve coding gain and two new combined methods can reduce the BD-Rate up to 1.7%, 1.2% and 1.0% on common test sequences and 1.4%, 1.0% and 1.1% on 4K videos under three configurations respectively. Besides, it's also shown that the remarkable improvement can be obtained at the cost of an acceptable increase in coding complexity and memory consumption.

ACKNOWLEDGMENT

This work was supported by NSFC (61521062, 61527804), and Shanghai Zhangjiang national independent innovation demonstration zone development fund (201501-PD-SB-B201-001).

REFERENCES

[1] G. J. Sullivan, J.-R. Ohm, W.-J. Han, and T. Wiegand, "Overview of the high efficiency video coding hevc standard," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 22, no. 12, pp. 1649–1668, 2012.

[2] 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 Techn.*, vol. 13, no. 7, pp. 620–636, 2003.

[3] V. Sze and M. Budagavi, "High throughput cabac entropy coding in hevc," *Circuits and Systems for Video Technology, IEEE Transactions on*, vol. 22, no. 12, pp. 1778–1791, 2012.

[4] V. Sze and D. Marpe, "Entropy coding in hevc," in *High Efficiency Video Coding (HEVC)*. Springer, 2014, pp. 209–274.

[5] D. Marpe and T. Wiegand, "A highly efficient multiplication-free binary arithmetic coder and its application in video coding," in *International Conference on Image Processing*, vol. 2. IEEE, 2003, pp. II–263.

[6] J. Sole, R. Joshi, N. Nguyen, T. Ji, M. Karczewicz, G. Clare, F. Henry, and A. Duenas, "Transform Coefficient Coding in HEVC," *IEEE Trans. Circuits Syst. Video Techn.*, vol. 22, no. 12, pp. 1765–1777, 2012.

[7] T. Nguyen, D. Marpe, and T. Weigand, "Non-CE11: Proposed Cleanup for Transform Coefficient Coding," *Doc. JCTVC-H0228*, February, 2012.

[8] J. Chen, W. J. Chien, M. Karczewicz, X. Li, H. Liu, A. Said, and *et al.*, "Further improvements to HMKTA-1.0," *Doc. VECG-AZ07*, 2015.

[9] W. J. Chien, J. Chen, J. Sole, and M. Karczewicz, "Template-based context modeling for coefficient coding," *Doc.JCTVC-I0383*, April,2012.

[10] J. Stegemann, H. Kirchhoffer, C. Bartnik, D. Marpe, M. Siekmann, and T. Wiegand, "Updated design of the probability estimator and the coding engine of the M Coder," *Document JCTVC-I0276*, April, 2012.

[11] A. Alshin and E. Alshina, "Multi-parameter probability up-date for CABAC," *Document JCTVC-F254*, July, 2011.

[12] F. Bossen, "Common test conditions and software reference configurations," *Doc. JCTVC-L1100*, January, 2013.

[13] G. Bjontegaard, "Calcuation of average psnr differences between rd-curves," *Doc. VCEG-M33 ITU-T Q6/16,*, April, 2001.