An Optimized Probability Estimation Model for Binary Arithmetic Coding

Jing Cui[1], Shanshe Wang[2], Nan Zhang[3] and Siwei Ma[4]

[1]School of Electrical & Computer Engineering
Seoul National University, Seoul, Korea
[3]School of Biomedical Engineering, Capital Medical University, Beijing, China
[2,4]School of Electronic Engineering & Computer Science
Peking University, Beijing, and [4]Peking University Shenzhen Graduate School, Shenzhen, China

[1]Jingcui106@gmail.com [2]sswang@jdl.ac.cn [3]zhangnan@ccmu.edu.cn [4]swma@pku.edu.cn

***Abstract*— In this paper, we analyze the binary arithmetic coding of High Efficiency Video Coding (HEVC) and the second generation of audio and video coding standard (AVS2). Then an optimized probability estimation scheme is proposed for arithmetic coder. The proposed scheme is incorporated into the HEVC reference software (HM 16.0) and AVS2 reference software (RD 10.1). Experimental results demonstrate that the proposed scheme can efficiently improve the coding efficiency of entropy coding. The rate-distortion (R-D) performance gain can be up to 0.21% for AVS2 and 0.30% for HEVC respectively.**

***Index Terms*— HEVC, AVS2, Context-based Binary Arithmetic Coding (CBAC), Context-based Adaptive Binary Arithmetic Coding (CABAC), probability estimation**

I. INTRODUCTION

High efficiency video coding (HEVC) [1] is the latest video coding standard developed by JCT-VC (Joint Collaborative Team on Video Coding). Meanwhile, in China, AVS group developed the second generation of audio and video coding standard (AVS2) [2]. Compared to the previous video coding standards, such as H.264/AVC [3] and AVS1 [4], coding performance of HEVC and AVS2 has been significantly improved.

To improve the coding performance, both HEVC and AVS2 adopted binary arithmetic coding. What is different is that HEVC adopted Context-based Adaptive Binary Arithmetic Coding (CABAC) [5] and AVS2 adopted Context-based Binary Arithmetic Coding (CBAC) [6]. These two schemes are both based on information entropy theory [7] and bring much coding performance improvement.

For the Binary Arithmetic Coding, the main procedure mainly includes binarization, context modelling, binary arithmetic coding and parameters updating. The binarization process in current video standard employs the similar model-probability distribution [8]. As for the context modelling, the selection of probability model is referred as the context modelling and the probability estimation is introduced to estimate the probability of current symbol valued from 0 to 1[9]. In addition, the probability estimation model also affects the probability update and range subdivision in the binary arithmetic coding engine where the recursive interval subdivision, parameters derivation, and relative updating are performed based on the principle of arithmetic coder. Therefore more compelling probability updating model is needed especially for the motive signal source. Many research works focus on how to optimize the binary arithmetic coding. In [10], the proposed "virtual sliding window" method provided a more outstanding compression rate compared with look-up table index based entropy coder in [9]. Currently, the virtual sliding window technique is widely explored in HEVC. An integrated window sizes technique is introduced in [11] ~ [13], which gives a higher precision estimation model with around 0.8% performance improvement in HEVC. In [14], a counter-based window sizes scheme is proposed and brings about 0.9% BD-rate saving. Therefore for probability estimation, the smaller window size of each probability model in the beginning of the sequence can improve the R-D performance considerably and the changeable window size tends to be more effective. The entropy coder CBAC used in AVS2 made the similar affords to design an adaptive probability estimation model to improve R-D performance, although it causes computation complexity increase. In this paper, we proposed a more effective scheme for probability estimation in CBAC and we also incorporate our proposed scheme as the entropy coder into HEVC to improve the CABAC.

The rest of the paper is arranged as follows. In Section II, probability estimation models in CABAC and CBAC are shown briefly. Section III proposed our optimization scheme. The experimental results both on AVS2 and HEVC are presented in Section IV. Finally, Section V concludes paper.

II. PROBABILITY ESTIMATION MODELS IN CABAC AND CBAC

Probability estimation plays a vital role in Binary Arithmetic Coding. Probability estimation model is introduced by adapting the internal state of the coder to the underlying source statistics. Such feature enhances the compression efficiency and it has been adopted into various entropy coding schemes such as M coder, PIPE. However, HEVC and AVS2 adopted different probability estimation models to support their respective arithmetic coding engine. In this section, we will briefly introduce nature of probability estimation. Then probability estimation models for HEVC and AVS2 are presented respectively.

A. Probability Estimation Model

For probability estimation model, one of the most frequently used formulas is shown as following (1):

$$p_\delta(i) = \alpha \cdot y(i) + (1-\alpha) \cdot p_{\delta-1}(i) \quad (1)$$

where *i* is "0" or "1" which denotes that current bin is most probability symbol (MPS) or the least probability symbol (LPS), respectively. $y(i)$ is 0 if the current symbol is MPS and is 1 otherwise. The value of δ denotes the probability state. The scaling factor α which adjusts adaptation speed indicates that how many in-prior encoded bins N_{bin} ($\alpha \leftarrow 1/N_{bin}$) are needed to estimate probability distribution for the coming bins. In some standards, in-prior encoded bins N_{bin} is presented by 2^{cw} through using sliding window factor *cw*. Two smoothing functions, Exponent Mesh and Uniform Mesh [12][13], can be utilized for probability updating. The Exponent Mesh employs the map function $p_\delta = 0.5(1-\alpha)^\delta$, where δ is the quantized state and indicates the discrete probability value covered interval domain (0, 0.5). For Uniform Mesh, the probability is presented as $p_\delta = P_\delta / 2^k$. P_δ is the scaled probability ranged between 0 and 2^k where *k* is an integer that denotes the scaling bit depth. Thus, the speculative computation for probability estimation is addressed by shift or/and addition operation ($P_\delta \leftarrow P_{\delta-1} \pm \{\Delta, (P_{\delta-1} >> cw)\}$) for LPS and MPS, respectively. Here, Δ is the increment of the Uniform Mesh.

According to required variables α, p_δ and δ in equation (1), methods to implement probability estimation are summarized in TABLE 1. Practically, each standard considers the tradeoff among the computation complexity, memory requirements and the estimation accuracy when exploring the method to balance all the variables and seeks maximum advantage.

TABLE 1
THE MODEL VARIABLES FOR PROBABILITY ESTIMATION

variable	models	formula	note
α	static	$1/N$	[1]
	adaptive	$1/2^{cw}$	[2]
p_δ	exponent mesh	$p_\delta = 0.5(1-\alpha)^\delta$	[1][13]
	uniform mesh	$p_\delta = P_\delta / 2^k$	[2][13]
δ	table-based	$nextState[\delta-1]$	[1]
	scale-based	$P_{\delta-1} \pm \{\Delta, (P_{\delta-1} >> cw)\}$	[2]

B. Probability Estimation Model in CABAC of HEVC

In CABAC of HEVC, two FSMs are employed to realize probability estimation procedure. Specifically, for LPS case, recalling the feature of Exponent Mesh, the probability distribution tends to be more intensive when it is closer to 0. Thus there will be an approximate boundary for the minimum probability $p_{min.}$ since the difference between two probabilities is smaller. Also, the maximum probability $p_{max.}$ is introduced to limit the up boundary if necessary. Interestingly, the window size *N* in this case can be derived by equation (2):

$$N = \lceil 1/(1-(p_{\min}/p_{\max})^{1/\delta_{\max}}) \rceil \quad (2)$$

where, $\delta_{\max.}$ is the maximum index number of state indices of FSM. Also, the scaling factor α can be described as ($\alpha \leftarrow (p_{\min.}/p_{\max.})^{1/\delta_{\max.}}$). This classical probability estimation model assigns low and up probability boundaries as 0.01875 and 0.5, respectively. Therefore, the window size *N* is about 19.69 and scaling factor α is around 0.95, simultaneously. The probability modelling and updating is performed with a 126-state indexed by the LPS or MPS as equation (3):

$$\delta_{prob} \leftarrow Mps\, ?\, nextState[mps][\delta] : nextState[lps][\delta] \quad (3)$$

where, δ_{prob} denotes context variable state which is stored in the context memory for the next access. *Mps* is another context variable which denotes the bin used this context is the most significant symbol or not. The array *nextState[ ][ ]* is defined for the state transition FSM with index δ ranged from 0 to 63.

C. Probability Estimation Model in CBAC of AVS2

The probability estimation in AVS2 is performed with logarithm addition and shift operation for CBAC algorithm. The Uniform Mesh and speculation computation are used for the probability up-date with multiplication free. The scaling factor for CBAC is defined as ($\alpha \leftarrow 1/2^{cw}$) with adaptive *cw* chose from 3, 4 and 5 according to the engine execution counter *cycno* for each context model. Specifically, at the beginning several iterations, a smaller scaling factor is used and it will stabilize at 5 after 2 iterations. In addition, the implementation of the probability estimation adopts the Uniform Mesh where the scaled probability is represented as the corresponding *LgPmps* with *k*-bit resolution. Here, *LgPmps* denotes the scaled absolute value of *log2*(*Pmps*) with *Pmps* valued from (0.5, 1). Hence, the factor *k* defined in Uniform Mesh function indicts the resolution (bit-depth) of *LgPmps*, theoretically. The scaled MPS probability *LgPmps* is described as equation (4):

$$LgPmps = 2^{bitDepth} \times |\log_2 P_{mps}| \quad (4)$$

where bit depth *bitDepth* is 10-bit and *Pmps* valued from (0.5, 1). Then we can achieve two boundary values, i.e., (0, 1024), for the *LgPmps* calculation in the arithmetic coding process. Thus the probability transition can be mapped into a scaled integer range. Then the estimation updating model employed in AVS2 can be fulfilled in the equation (5):

$$LgPmps = \begin{cases} LgPmps - (LgPmps >> cw) & if\ mps \\ LgPmps + \Delta & if\ \ lps \end{cases} \quad (5)$$

where *cw* is the sliding window factor as described before, Δ is the increment of the *LgPmps* once encoding one bin based on the Uniform Mesh for case that the symbol is LPS, which is also relative to the *cw* and the bit depth of the *LgPmps*.

III. Proposed Optimized probability estimation model For Arithmetic Coding Engine

Probability estimation is a crucial step in arithmetic coding of CBAC as illustrated in Fig. 1. It has much influence on the final coding performance. In CBAC, context variables includes 10-bit *LgPmps*, 1-bit *valMps*, and 2-bit *cycno*. Once the regular arithmetic coding is finished, the context variables will be updated including *LgPmps* speculation, *valMps* conversion (if necessary), and *cycno* marking. Even through this adaptation increases the computation complexity, the coding performance of CBAC tends to be competitive compared with CABAC. In this section, based on the mechanism in CBAC, we propose an optimized probability estimation model with well-regulated scheme to improve coding efficiency.

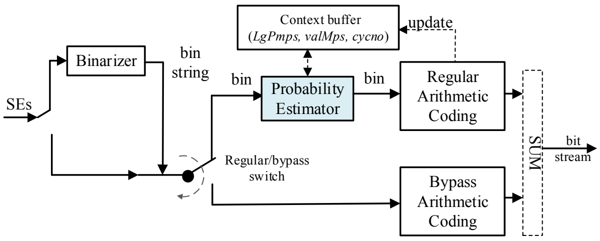

Fig.1 The flowchart of CBAC encoder

Referring to the analysis of Uniform Mesh in above section, it can be concluded that the scaled probability *LgPmps* is valued within a scaled domain as (0, $2^{bitDepth}$) in theory. Thus the probability estimation can be performed by addition or subtraction, and shift within integer data domain. Considering that the estimation error of probability of MPS near to 0.5 tends to be more considerable than that close to 1 where the difference between two symbols is marginal, we deign a feasible data domain, called (Thr_{LgPmps}, $Init_{LgPmps}$), for probability estimator of the CBAC. Thr_{LgPmps} denotes the low boundary that the scaled probability *LgPmps* can reach. $Init_{LgPmps}$ is the initial value assigned to each context model at the beginning of new slice.

For the initial value, it is assigned as in CBAC as follow,

$$Init_{LgPmps} = 2^{bitDepth} - \tau \qquad (6)$$

where τ is valued as 0 or 1. For the threshold value Thr_{LgPmps}, it is represented by (7):

$$Thr_{LgPmps} = 2^{bitDepth} \cdot |\log_2(1-\hat{p}_{\min,plps})| \qquad (7)$$

where $\hat{p}_{\min,pls}$ is the statistical result of minimum LPS probability which can be obtained through the similar method used for the CABAC in [5]. In theory, it is a statistical result.

Based on the provided scheme, the scaled probability *LgPmps* can be transited within the feasible domain with the uniform increment each iteration in the LPS case. However, note that the adaptive scaling factor *cw* is introduced in CBAC where the sliding window size will be changed along with context variable *cycno* marking, thus the uniform increment will also adaptively change and the adaptive uniform increment $\tilde{\Delta}$ is defined as equation (8):

$$\tilde{\Delta}_{bitDepth,cw} = 2^{bitDepth} \times |\log_2(1-2^{-cw})| \qquad (8)$$

Therefore, the proposed probability estimation model can be modified with the following equation (9):

$$LgPmps = \begin{cases} \max((LgPmps - LgPmps >> cw), Thr_{\lg pmps}) & if\ mps \\ LgPmps \geq 1024 ? (2^{bitDepth+1} - LgPmps) : (LgPmps + \tilde{\Delta}) & if\ lps \end{cases} \qquad (9)$$

In implementation, parameters adjustments including *cw*, *bitDepth*, Thr_{LgPmps}, and $Init_{LgPmps}$ are necessary in order to find out the best scheme. Then the overall schedule for the probability estimation can be illustrated as Fig.2.

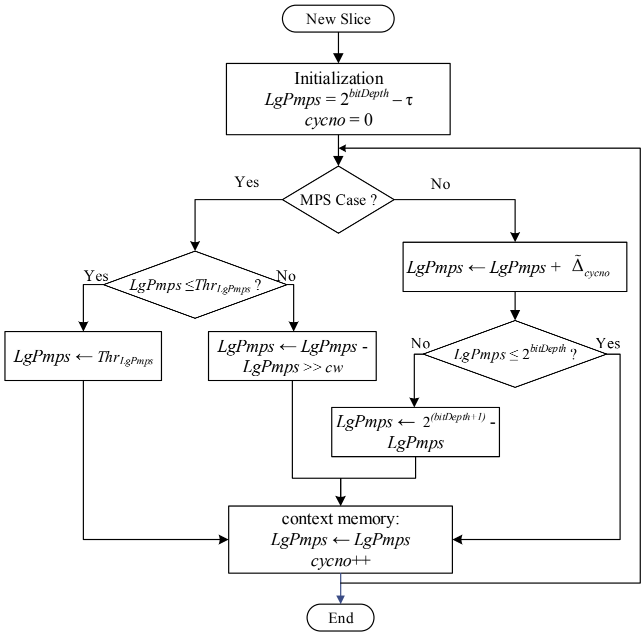

Fig.2 The proposed probability estimation scheme for each context model.

IV. Experiment Results

To verify the coding efficiency of proposed optimized probability estimation model, experiments are conducted on HM 16.0 and RD 10.1 respectively. *Bjøntegaard* Delta (BD) rate is utilized to evaluate the coding efficiency of proposed scheme. The bit depth *bitDepth* is assigned as 9, $\hat{p}_{\min,pls}$ is about 0.0382, τ is set as 1, and the final sliding factor *cw* is set as 5. Note that *cw* is determined by the *cycno* marking and we assign the value of *cw* along with different *cycno* and syntax element type. Until *cycno* increases up to 3, *cw* is assigned as 5 for each context model for all syntax elements.

A. R-D Performance in AVS2

To test the coding efficiency of the proposed scheme on AVS2, some video sequences with higher resolution are tested as illustrated in TABLE 2. The testing configuration is random-access. The quantization parameters are set to be 27, 32, 38 and 45 according to common test condition of RD10.1 [15]. TABLE 2 shows the experiment results of proposed scheme with RDOQ off. It can be seen that the maximum coding gain can be up to 0.49% with 0.21% on average. TABLE 3 shows the coding performance improvement achieved by the proposed scheme with RDOQ on. The average coding gain is 0.19%.

TABLE 2
THE CODING EFFICIENCY FOR THE PROPOSED SCHEME WITH RDOQ-OFF

Test Sequence	Resolution	Y	U	V
Traffic	2560*1600	-0.20%	-0.24%	-0.71%
Pku-girls	3840*2160	-0.10%	-0.94%	-0.73%
PeopleOnStreet	2560*1600	-0.21%	-0.88%	-1.43%
BQTerrace	1080p	-0.49%	-0.82%	-0.42%
beach	1080p	-0.07%	-8.48%	-9.65%
taishan	1080p	-0.13%	-0.25%	-0.62%
kimono	1080p	-0.10%	-0.21%	-0.47%
cactus	1080p	-0.28%	-1.43%	-0.70%
BasketballDrive	1080p	-0.29%	-0.67%	-0.54%
Average		**-0.21%**	**-1.55%**	**-1.70%**

TABLE 3
THE CODING EFFICIENCY FOR THE PROPOSED SCHEME WITH RDOQ ON

Test Sequence	Resolution	Y	U	V
Traffic	2560*1600	-0.23%	-0.39%	-0.76%
Pku-girls	3840*2160	-0.16%	-0.44%	-0.42%
PeopleOnStreet	2560*1600	-0.23%	-0.73%	-0.89%
BQTerrace	1080p	-0.19%	-0.82%	-0.95%
beach	1080p	-0.14%	-7.37%	-6.00%
taishan	1080p	-0.15%	-0.53%	-0.45%
kimono	1080p	-0.21%	0.14%	-0.36%
cactus	1080p	-0.24%	-1.09%	0.40%
BasketballDrive	1080p	-0.13%	-0.47%	-0.62%
Average		**-0.19%**	**-1.30%**	**-1.12%**

B. *Performance Improvement in HEVC*

To verify coding efficiency of proposed arithmetic coding scheme, we also incorporate the proposed scheme into the HEVC software HM 16.0. Firstly, we use the CBAC engine to replace the CABAC engine in HEVC. Note that only the entropy coder engine is replaced with matched context variables and engine variables, and there is no change in binarization process. Then, the proposed scheme is employed to test efficiency of HM encoder. The same video sequences are encoded with a random-access configuration for four QPs as 22, 37, 32, and 37 according to HEVC common test [16]. TABLE 4 shows the experimental results. It can be seen that there is about 0.30% BD-rate reduction on average. The maximum coding gain can be over 0.5%.

V. CONCLUSION

In this paper, we propose an optimized probability estimation model for arithmetic coding. With continuous adjusting and optimizing parameters of the model, the proposed probability estimation scheme can achieve a considerable performance enhancement both on AVS2 and HEVC.

ACKNOWLEDGMENT

This work was supported in part by the National High-tech R&D Program of China (863 Program, 2015AA015903), National Natural Science Foundation of China (61322106, 61272255), the Scientific Research Common Program of Beijing Municipal Commission of Education (KM201310025009), and Shenzhen Peacock Plan, which are gratefully acknowledged.

TABLE 4
THE CODING EFFICIENCY FOR THE PROPOSED SCHEME

Sequence	Resolution	Y	U	V
Traffic	2560*1600	-0.27%	-0.95%	-0.57%
Pku-girls	3840*2160	-0.29%	-1.75%	-0.96%
PeopleOnStreet	2560*1600	-0.43%	-1.66%	-1.45%
BQTerrace	1080p	-0.57%	-1.19%	-0.99%
beach	1080p	-0.11%	-1.21%	-0.97%
taishan	1080p	-0.27%	-1.03%	-1.49%
Kimono	1080p	-0.23%	-1.04%	-0.74%
cactus	1080p	-0.15%	-1.58%	-1.48%
BasketballDrive	1080p	-0.39%	-0.73%	-0.76%
Average		**-0.30%**	**-1.17%**	**-1.06%**

REFERENCES

[1] Bross. B, Han. W.-J, Ohm. J.-R., Sullivan. G. J, and Wiegand. T, “High efficiency video coding (HEVC) text specification draft 10,” Joint Collaborative Team on Video Coding (JCT-VC) of ITU-T SG 16 WP 3 and ISO/IEC JTC 1/SC 29/WG 11, document JCTVC-L1003, Geneva, Switzerland, Jan. 2013.

[2] Information Technology – High Efficiency Media Coding – Part2: Video, Chinese GB/T201503, 2015.

[3] Join Video Team (JVT) ISO/IEC MPEG, ITU-T VCEG: Final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14496-10 AVC). JVT-G050, 2003.

[4] Information Technology – High Efficiency Media Coding – Part2: Video, Chinese GB/T200602, 2006.

[5] Marpe D, Schwarz H, Wiegand T. “Context-based adaptive binary arithmetic coding in the H. 264/AVC video compression standard”. Circuits and Systems for Video Technology, IEEE Transactions on, 2003, 13(7): 620-636.

[6] Zhang L, Wang Q, Zhang N, et al. “Context-based entropy coding in AVS video coding standard” in Signal Processing: Image Communication, 2009, 24(4): 263-276.

[7] Hankerson D C, Harris G A, Johnson Jr P D. “Introduction to information theory and data compression.” CRC press, 2003.

[8] Sze, V., & Marpe, D. “Entropy Coding in HEVC”. In High Efficiency Video Coding (HEVC) (pp. 209-274). Springer International Publishing, 2014.

[9] Said A. “Introduction to arithmetic coding-theory and practice”. Hewlett Packard Laboratories Report, 2004.

[10] Belyaev E, Gilmutdinov M, Turlikov A (2006) Binary arithmetic coding system with adaptive probability estimation by “virtual sliding window” in IEEE tenth international symposium on consumer electronics (ISCE ’06), pp 1–5, 2006.

[11] Alshin A, Alshina E, Park J H. “High precision probability estimation for CABAC” in Visual Communications and Image Processing (VCIP), 2013. IEEE, 2013: 1-6.

[12] Alshin A, Alshina E, Park. I "CEI (subset B): Multi-parameter probability up-date for CABAC," Document of Joint Collaborative Team on Video Coding, JCTVC-G0764, November 2011.

[13] Alshina E, Alshin A (2011) Multi-parameter probability up-date for CABAC, Joint Collaborative Team on Video Coding (JCT-VC), Document JCTVC-F254, Torino, July 2011

[14] Stegemann J, Kirchhoffer H,Marpe D,Wiegand T (2011) Non-CE1: counterbased probability model update with adapted arithmetic coding engine, Joint Collaborative Team on Video Coding (JCT-VC), Document JCTVC-G547, Geneva, Nov. 2011

[15] AVS-P2 common test condition, AVS-N2020, 2014.

[16] Common HM Test Conditions and Software Reference Configurations, JCTVC-L1100, ITU-T/ISO/IEC Joint Collaborative Team on Video Coding (JCT-VC), Jan. 2013.