Computational complexity analysis of adaptive arithmetic coding in HEVC video compression standard

Damian Karwowski

Poznan University of Technology,
Chair of Multimedia Telecomunications and Microelectronics
dkarwow@multimedia.edu.pl

Summary. This paper presents computational complexity analysis of adaptive arithmetic coding (CABAC) in the emerging HEVC video compression technology. In particular, computational complexity of individual parts of CABAC (and the whole CABAC entropy codec in the HEVC video decoder) was measured from the point of view of video decoder side. Experiments were done using publically available HM reference software of the HEVC video codec and a set of test video sequences. The range of bitrates that can be processed in real time by CABAC entropy decoder were also evaluated for the considered in the paper implementation of CABAC.

Key words: adaptive arithmetic coding, CABAC algorithm, entropy coding, HEVC video compression

1 Introduction

Contemporary techniques of video compression have been dominated by hybrid coding scheme of images with intra-frame prediction, inter-frame prediction and lossy transform coding of residual data [1, 2, 3]. The well-known and commonly used examples of technologies that use hybrid coding scheme are MPEG-2 [1, 2], H.263 [1, 2] and MPEG-4 AVC/H.264 [1, 2, 3, 5]. The essential part of each hybrid video codec is a block of entropy coding that is used to extra reduce redundancy that exists within residual data in order to further improve the compression performance of a video encoder. Works on entropy coding of a video data resulted in the development of a very efficient Context-based Adaptive Binary Arithmetic Coding (CABAC) algorithm [3, 4, 5, 6]. CABAC technique represents the state-of-the-art solution and has become a part of the MPEG-4 AVC/H.264 video compression technology. The impact of CABAC algorithm on compression performance and computational complexity of a video codec has been already tested well and presented in the literature [3, 4, 5, 6, 7]. What is very important, such works have been only done in the context of the MPEG-4 AVC/H.264 video compression standard.

In the last years intensive works have been carried out aimed at further improving the techniques of hybrid video coding. The works were mainly done in the context of high definition television. As a result of these works the new High Efficiency Video Coding (HEVC) international standard has been announced this year [8]. In the course of work on the new standard, modified version of CABAC algorithm has been elaborated and chosen to be a part of the HEVC video compression technology. It was experimentally proved that the two versions of CABAC (original and modified) have similar compression performance, but complexity of the modified version (used in HEVC) is lower relative to the original one (used in MPEG-4 AVC/H.264). Additionally, compared to previously known solutions for video coding, the new HEVC technology includes many improvements and new coding tools that affect both the compression performance and complexity of a video codec [8]. These changes come down to a more flexible way of splitting the image into coding units, more sophisticated mechanisms of intra- and inter-frame prediction, more advanced techniques of transform coding of residual data and the application of in-loop image filters.

These improvements (made in the parts of a video codec that calculate the residual data and that perform entropy coding of the data) influences the amount of computations that are needed in individual parts of contemporary video codec. Computational complexity analysis and results obtained in the context of older video compression technologies (such as MPEG-4 AVC/H.264) may not be appropriate for the emerging HEVC technology.

At this point the question arises on computational complexity of modified CABAC entropy codec in the framework of the new HEVC video codec. This paper strictly addresses this subject. Additionally, the paper presents detailed computational complexity analysis of individual functional blocks of CABAC entropy codec using representative set of test video sequences. The rest of the paper is organized as follows. Section 2 presents the general concepts of CABAC entropy coding that is used in the HEVC standard. Section 3 describes methodology of experiments on computational complexity analysis of CABAC in HEVC. Section 4 presents obtained results and the last section 5 concludes the paper.

2 HEVC video compression - CABAC entropy coding

Proper evaluation of results presented in this paper requires the knowledge of CABAC entropy codec structure. In this section only the most important elements of CABAC algorithm are presented.

The general structure of the version of CABAC used in HEVC is the same as compared to that used in MPEG-4 AVC/H.264 video codec. The block diagram of CABAC encoder was presented in Figure 1.

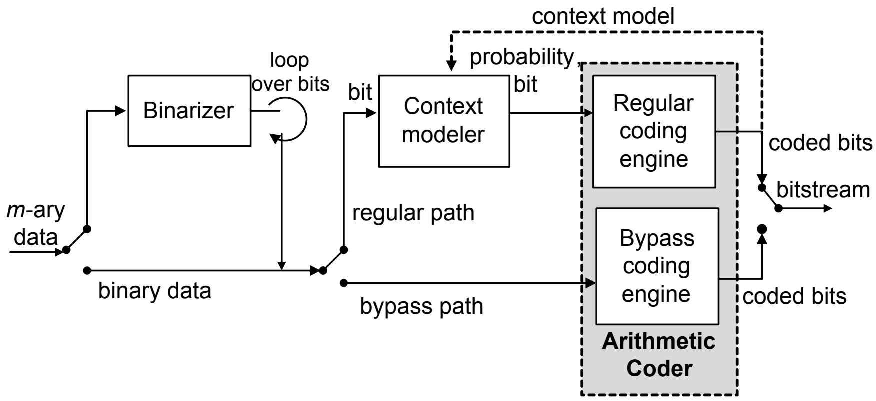

Fig. 1. Block diagram of CABAC entropy encoder.

Characteristic feature of an algorithm is using the binary arithmetic encoder core (instead of m-ary core), the so-called M-encoder, which was highly optimized toward speed [4, 5, 6, 7]. The binary encoder core is able to process symbols from alphabet A={0, 1} - in notation of CABAC the symbols are considered as least probable symbol (LPS) or most probable symbol (MPS).

Due to application in CABAC of the binary arithmetic encoder core, all m-ary symbols must be mapped into string of binary symbols at the first stage of CABAC encoding. This operation is realized in the binarizer. It must be stated, that the way in which binarizer works strongly affects the number of resulted binary symbols, which in turn translates into performance and complexity of the entropy encoder. It was the motivation to use several different binarization schemes in CABAC in order to account different statistics of individual syntax element coded in the HEVC encoder. In general, binarization techniques are used that are based on unary coding, Exp-Golomb coding and Golomb-Rice coding [4, 5, 6, 9]. In this way, this part of CABAC encoder is very similar to adaptive variable-length coding, but data statistics modeling together with arithmetic coding of binary symbols is additionally realized in order to increase compression performance of entropy encoder.

Data within string of binary symbols (data at the output of a binarizer block) still exhibit some statistical redundancy. This redundancy is additionally reduced in CABAC by the use of data statistics modeling and arithmetic encoding of symbols. From compression performance point of view these are also important parts of entropy encoder. It is known, that the 'quality' of data statistics estimation (accuracy of probabilities of symbols that are estimated in the context modeler block) strongly affects compression efficiency of arithmetic encoding. Experiments proved different character of data associated with individual syntax elements and even varied nature of binary data within bit planes of binarized words for a given syntax element. Therefore, estimation of probabilities should be realized independently not only on the level of individual syntax elements, but also independently on the level of bit planes in a binarized word. It was the motivation to split the input stream of binary symbols into a big number of individual binary sub-streams, for which the probabilities of symbols are calculated independently. This refers to sub-streams of data that are coded in the so-called regular coding mode (see Figure 1). In relation to solution used in CABAC of MPEG-4 AVC/H.264 standard, in CABAC of HEVC the number of regular sub-streams was highly reduced in order to simplify the mechanism of data statistics estimation. Total number of 205[1] regular sub-streams was introduced in CABAC of HEVC (in contrast to 460 regular sub-streams used in CABAC of MPEG-4 AVC/H.264). The way in which binary symbols are assigned to individual regular sub-streams determines the efficiency of entropy encoder. In CABAC it is done based on context information (i.e. values of neighboring binary symbols that have been coded previously), wherein actual context pattern (the location of neighboring symbols that are taken into account) is different for individual syntax elements. For some binary symbols uniform probability distribution is assumed and no conditional probabilities are calculated for such symbols. It refers to the so-called bypass coding mode and bypass sub-stream of symbols (see Figure 1). Relative to the CABAC of MPEG-4 AVC/H.264 greater percentage of symbols is classified as bypass symbols in the CABAC of HEVC.

In general the process of probabilities estimation makes a bulk of computations in adaptive entropy codec. Therefore, CABAC calculates probabilities in a simplified way using pre-defined Finite State Machines (FSMs). The number of FSMs corresponds to the number of regular sub-streams defined in CABAC - it means that a given FSM calculates probabilities of symbols that come from individual sub-stream. Each FSM consists of 64 states which correspond to particular 64 values of probabilities for LPS symbol and 64 values of probabilities for MPS symbol. Probability of a symbol is finally used by the arithmetic encoder core.

3 Complexity analysis of CABAC in HEVC - methodology

Main goal of the paper was to perform detailed complexity analysis of individual functional blocks of CABAC entropy codec when operating in the framework of HEVC video codec. The starting point to research was the publicly available reference software of the HEVC video codec, known as HM software [10]. In particular, version 3.2 (HM 3.2) of the software was used. As a matter of fact the newer versions of HM software are available now (HM 10.0 at time of writing this paper), nevertheless the general structure of CABAC algorithm has not been changed (since HM 3.2 software version) and results obtained with older version of the software may be regarded as reliable.

[1] This is the number of regular sub-streams used in the HM 10.0 version of the HEVC reference software.

Experiments were done from the point of view of HEVC video decoder. It is known that the most computationally complex blocks of CABAC are data statistics modeling and arithmetic encoding [7]. These parts are the same in both the encoder and the decoder. Therefore, complexity results of CABAC decoder correspond well to that of the CABAC encoder side.

The HM software was modified by the author in order to be able to capture the execution time of the following parts of CABAC: 1) binary arithmetic decoder core operating in the regular mode (regular mode of the M-decoder), 2) binary arithmetic decoder core operating in the bypass mode (bypass mode of the M-decoder), 3) de-binarization of data and context modeling for binary symbols. Additionally, the execution time of the whole CABAC entropy decoder was tested in the HEVC decoder. Execution times of mentioned parts of CABAC (and the whole CABAC entropy decoder) were measured using RDTSC Pentium processor instruction. This instruction counts the number of processor cycles that elapsed during execution of considered fragment of a program code. Finally, the frequency of use the bypass mode in CABAC codec was explored for a wide range of target bitrates. All experiment were done using the following scenario:

- HD and full HD test video sequences were used: *Station* (1920x1080, 25Hz), *RiverBed* (1920x1080, 25Hz), *PoznańStreet* (1920x1088, 25Hz), *Balloons* (1024x768, 30Hz), *ChinaSpeed* (1024x768, 30Hz), *SliceEditing* (1280x720, 30Hz);
- The IBBPBBPBBP structure of group of pictures (GOP) was used;
- Experiments were done for different values of quantization parameter (QP): QP=22, 27, 32, 37, that correspond to quality of reconstructed video from excellent (QP=22) to poor (QP=37).

All tests were done on Intel Core i7 950 platform (3.07GHz clock frequency, 8MB cache memory) equipped with 12GB of RAM and working under Microsoft Windows 7 system.

4 Complexity of CABAC in HEVC - results

In the course of experiments complexity of individual parts of CABAC decoder was measured, as clearly pointed out in the previous section. Detailed results achieved for different test video sequences and wide range of bitrates (250-31000 kbps) were presented in Table 1.

Results revealed that de-binarization of syntax elements and context modeling of binary symbols make a bulk of computations in CABAC. Depending on the bitrate of encoded bitstream it is 50% to 70% of the total computations in entropy codec (it was 62% in average). Such a large amount of computations for this part of entropy codec results from main features of applied algorithms. And so, de-binarization of syntax elements is very similar to traditional Huffman decoding with necessity of bit-by-bit processing of data. Binary symbols are finally arithmetically decoded taking into account the statistics of symbols. The algorithm of context modeling is very irregular in CABAC and large amount of computations must be done to calculate probabilities of symbols. In particular, calculation of context information is the most time consuming part of the context modeler block. All this affects complexity of CABAC codec to a large extent.

The core of binary arithmetic coding has been greatly accelerated due to application of the M-codec core. Additionally, some of symbols are coded using bypass coding mode where the computationally complex stage of symbols' probabilities estimation is skipped. All mentioned simplifications of the core have reduced its complexity significantly - the M-decoder core makes 27% to 47% of the total computations in CABAC (it was 37% in average).

Share of computations in different parts of CABAC strongly depends on the target bitrate. As achieved in experiments, the higher the bitrate the higher amount of computations are associated with arithmetic decoder core. Such an observation has been noticed for each of test video sequences (see Table 1). In author's opinion, the answer lies in the number of symbols coded in bypass mode. Lower values of QP parameter (that correspond to higher bitrate and better quality of reconstructed video) increases the number of symbols that are coded in the bypass mode. For this coding mode there is no context modeling of symbols, which increases percentage contribution of arithmetic decoder core in the whole CABAC noticeably.

The solution in CABAC of HEVC that reduces complexity of the entropy codec substantially (relative to CABAC of MPEG-4 AVC) is increasing the frequency of using the bypass mode (in average, 21% of symbols are coded in this mode in HEVC - see Table 2). Table 2 presents details for the percentage of bypass bins in the encoded stream and comparison of complexity of arithmetic decoder (AD) core when operating in the regular and bypass mode. In average, complexity of the bypass AD core is 80% of the complexity of the regular AD core. This comparison concerns arithmetic decoder cores only (bypass and regular cores) and does not take into account the computationally complex stage of context modeling that must be additionally done in the case of regular mode. Additionally, other complexity differences were noticed for the lower and the higher bitrates (see Table 2). It may be a result of different actions in re-normalization part of AD core for cases of uniform (bypass mode) and non-uniform (regular mode) probability distributions of symbols.

The optimized CABAC decoder is 8% of computations in the whole HEVC decoder (see Table 1). Table 3 presents the averaged (over test sequences) results of complexity of CABAC codec per one binary symbol.

CABAC decodes a binary symbol in almost 190 processor cycles, with 120 processor cycles needed for de-binarization and context modeling part of CABAC. For a processor operating with 3GHz clock frequency it gives a possibility of real-time CABAC decoding of streams with bitrate up to 16Gbps. It should be keep in mind that this result applies only to algorithmically optimized sequential version of the software CABAC decoder. Additional oppor-

Table 1. Percentage contribution of arithmetic decoder (AD) core (M-decoder) and de-binarization (de-bin) and context modeling in CABAC decoder. Contribution of CABAC decoding in the whole HEVC decoding time.

	HEVC decoder time (**original HEVC)**		
Test sequence	% of AD core in CABAC	% of de-bin and context modeling in CABAC	% of CABAC in HEVC
Station2			
QP=22 (3781 kbps)	38.91	61.09	7.36
QP=27 (1430 kbps)	35.14	64.86	3.93
QP=32 (707 kbps)	31.66	68.34	2.46
QP=37 (361 kbps)	27.86	72.14	1.68
River Bed			
QP=22 (30874 kbps)	41.93	58.07	24.91
QP=27 (15880 kbps)	40.47	59.53	15.07
QP=32 (8041 kbps)	38.71	61.29	9.69
QP=37 (3858 kbps)	36.91	63.09	6.18
Poznań Street			
QP=22 (3050 kbps)	38.99	61.01	7.69
QP=27 (1574 kbps)	36.13	63.87	4.77
QP=32 (808 kbps)	32.66	67.34	3.44
QP=37 (425 kbps)	28.71	71.29	2.51
Balloons			
QP=22 (1742 kbps)	37.58	62.42	6.64
QP=27 (807 kbps)	35.25	64.75	4.40
QP=32 (433 kbps)	32.76	67.24	3.02
QP=37 (251 kbps)	29.93	70.07	2.26
China Speed			
QP=22 (7007 kbps)	42.88	57.12	20.06
QP=27 (3675 kbps)	41.07	58.93	14.32
QP=32 (1827 kbps)	39.01	60.99	9.48
QP=37 (926 kbps)	36.79	63.21	6.09
Slice Editing			
QP=22 (2153 kbps)	47.17	52.83	14.59
QP=27 (1577 kbps)	45.11	54.89	11.32
QP=32 (1154 kbps)	42.49	57.51	8.55
QP=37 (844 kbps)	39.54	60.46	6.37
Average	**37.402**	**62.598**	**8.200**

tunities exist to parallelize computations in CABAC in the case of hardware implementation [7].

Table 2. Computational complexity of bypass coding engine (relative to regular coding engine) in binary arithmetic decoder core of CABAC. Percentage contribution of bins coded in bypass mode in the HEVC video decoder.

	HEVC decoder time (**original HEVC**)	
Test sequence	% Complexity of AD bypass core (relative to AD regular core)	% of bypass bins
Station2		
QP=22 (3781 kbps)	80.774	15.336
QP=27 (1430 kbps)	75.597	16.806
QP=32 (707 kbps)	69.769	16.711
QP=37 (361 kbps)	64.408	16.539
River Bed		
QP=22 (30874 kbps)	91.694	20.566
QP=27 (15880 kbps)	87.792	21.127
QP=32 (8041 kbps)	84.424	19.978
QP=37 (3858 kbps)	82.225	19.157
Poznań Street		
QP=22 (3050 kbps)	81.265	21.402
QP=27 (1574 kbps)	77.541	19.241
QP=32 (808 kbps)	73.560	17.383
QP=37 (425 kbps)	68.388	16.395
Balloons		
QP=22 (1742 kbps)	80.579	19.739
QP=27 (807 kbps)	76.816	20.384
QP=32 (433 kbps)	72.447	20.008
QP=37 (251 kbps)	68.287	19.267
China Speed		
QP=22 (7007 kbps)	83.085	24.641
QP=27 (3675 kbps)	79.820	22.582
QP=32 (1827 kbps)	76.554	20.552
QP=37 (926 kbps)	73.441	18.621
Slice Editing		
QP=22 (2153 kbps)	95.150	37.816
QP=27 (1577 kbps)	91.981	32.573
QP=32 (1154 kbps)	85.671	27.123
QP=37 (844 kbps)	78.258	21.901
Average	**79.147**	**21.077**

Table 3. Complexity of CABAC (in processor cycles) for one binary symbol. Average over 6 test sequences.

	Complexity of CABAC in HEVC **(per 1 binary symbol)**			
	whole CABAC decoder time [**processor ticks**]	de-bin and context modeling time [**processor ticks**]	AD regular core time [**processor ticks**]	AD bypass core time [**processor ticks**]
Average over test sequences	**188**	**119**	**72**	**57**

5 Conclusions

CABAC entropy decoding makes significant part of total computations in the process of HEVC video decoding. For the considered in the paper implementation of the HEVC decoder it is 8% (in average) of total decoding time, but this value strongly depends on the value of bitrate and increases with the increase of the bitrate. The content of the video sequence also affects exact percentage contribution of entropy decoding in the whole video decoder.

Results revealed that de-binarization of data and context modeling of binary symbols make a bulk of computations in CABAC. Depending on the bitrate of encoded bitstream it is 50% to 70% of the total computations performed in entropy decoder (it was 62% in average). The core of binary arithmetic decoder makes 37% of total computations in CABAC in average.

Using of bypass coding mode for selected binary symbols reduces complexity of CABAC entropy decoder significantly (see section 4 for detailed discussion). Depending on the bitrate and content of a video sequence, 15%-37% of binary symbols are processed in this coding mode (21% in average). In the bypass mode, complexity of arithmetic decoder core is about 80% of complexity of the core operating in the regular mode.

In experiments, almost 190 processor cycles were needed to process one binary symbol with CABAC. It allows to process in real-time encoded streams with bitrates up to 16Mbps.

Acknowledgment

Research project was supported by The National Centre for Research and Development, Poland. Grant no. 0889/R/T02/2010/10.

References

1. M. Domański, Obraz cyfrowy, Wydawnictwa Komunikacji i Lacznosci, 2011.
2. J. W. Woods, Multidimensional Signal, Image, and Video Processing and Coding, Academic Press, 2012.

3. I. E. G. Richardson, The H.264 Advanced Video Compression Standard. Second Edition, Wiley 2010.
4. D. Marpe, H. Schwarz, and T. Wiegand, Context-Based Adaptive Binary Arithmetic Coding in the H.264/AVC Video Compression Standard, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 13, No. 7, pp. 620-636, July 2003.
5. Special issue on H.264/AVC video coding standard, IEEE Trans. Circuits and Systems for Video Technology, Vol. 13, July 2003.
6. X. Tian, T. M. Le, Y. Lian, Entropy Coders of the H.264/AVC Standard, Springer-Verlag, 2011.
7. D. Karwowski, Advanced Adaptation Algorithms of Arithmetic Coding in Hybrid Video Compression, Doctoral Dissertation, Pozna University of Technology, 2008.
8. G. J. Sullivan, J. R. Ohm, W. J. Han, T. Wiegand, Overview of the High Efficiency Video Coding (HEVC) standard, IEEE Transactions on Circuits and Systems for Video Technology, Vol. 22, No. 12, pp. 1649 - 1668, December 2012.
9. T. Nguyen, M. Winken, D. Marpe, H. Schwarz, T. Wiegand, Reduced-complexity entropy coding of transform coefficient levels using a combination of VLC and PIPE, JCTVC- D336, JCT-VC Meeting, Daegu, January 2011.
10. HEVC Test Model, HM 3.2, http://hevc.hhi.fraunhofer.de