IMPROVED CONTEXT-ADAPTIVE ARITHMETIC CODING IN H.264/AVC

Damian Karwowski, Marek Domański

Poznań University of Technology, Chair of Multimedia Telecommunication and Microelectronics
Polanka 3, 60-965 Poznań, POLAND
phone: + (48) 61 665 39 00, fax: + (48) 61 665 38 99, email: dkarwow@multimedia.edu.pl, domanski@et.put.poznan.pl
web: www.multimedia.edu.pl

ABSTRACT

In the paper, an improved Context-based Adaptive Binary Arithmetic Coding (CABAC) is proposed for applications in video compression. The improved CABAC is characterized by higher coding efficiency as compared to that of the original CABAC as standardized in ISO MPEG-4 AVC (Advanced Video Coding) and ITU-T H.264. In the proposal, the increase of compression ratio is obtained by application of more exact data statistics estimation based on well-known data modeling algorithm: Context-Tree Weighting (CTW). The improved CABAC codec has been tested in the framework of AVC video codec. The extensive experiments show that application of the modified AVC yields 1.2% - 5.4% bitrate reduction. Computational effort of the modified AVC decoder is about 5-10% and 15-25% greater relative to the original AVC decoder for SD and HD video sequences, respectively.

1. INTRODUCTION

For high compression efficiency, entropy coding is always used at the output stage of hybrid video encoders. Entropy coding reduces statistical redundancy that exists within data streams that represent transform coefficients of residual signal, motion data and control information.

Currently, in video encoders, the state-of-the-art entropy coding technique is Context-based Adaptive Binary Arithmetic Coding (CABAC) [1] that has been incorporated into the new international standard called Advanced Video Coding (AVC) [6]. As compared to other entropy coders hitherto used in video compression, CABAC technique exploits more efficient arithmetic coding and far more sophisticated mechanisms of data statistics modeling. Currently, CABAC technique is distinguished by the highest compression ratio among entropy coders used in digital video compression.

Very high compression efficiency of CABAC is mainly a result of application of an advanced technique for data statistics estimation. The goal of this paper is to prove that compression performance of CABAC may be further improved by use of even more sophisticated data statistics modeling. The paper presents in detail a proposed modified CABAC codec together with new mechanisms of data statistics modeling based on the well-known Context-Tree Weighting (CTW) method [2, 3, 7, 12].

This paper reports results obtained in continuation of previous works of the authors [8-11]. In [8,9] an older version of modified CABAC codec has been described. As compared to authors' previous works, this paper describes a modified CABAC codec with three significant modifications that improve compression performance and reduce computational effort. These modifications are the following:

- More sophisticated method of context trees initialization;
- Adaptation of the CTW-based context modeller to fast binary arithmetic codec (M-codec) applied in the original CABAC [1, 5];
- Improved mechanism of data statistics estimation that reduces computational complexity of the codecs previously proposed by the authors

In order to verify the ideas, a complete AVC video codec with modified CABAC codec has been implemented and used in experimental tests. All experiments have been performed by production of compressed bitstreams and decoding of video. For the modified CABAC codec, implementation correctness has been verified by comparison of video clips decoded from the standard bitstreams and bitstreams produced by the AVC encoder with the modified CABAC encoder.

The compression ratio of the presented improved CABAC encoder has been thoroughly tested with various test video sequences. The results have been compared to the efficiency of the original CABAC encoder. Both adaptive entropy codecs have been tested within the framework of H.264/AVC video codec. Complexity of the modified AVC video decoder (with the improved CABAC) has also been tested against the complexity of the original AVC video decoder.

2. STANDARD CABAC ALGORITHM

Firstly we are going to review some features of the standard adaptive arithmetic CABAC encoder (see Figure 1).

In order to obtain high throughput of a CABAC codec, fast core of binary arithmetic codec (M-codec) has been used [1, 5]. Application of binary arithmetic codec core created necessity of mapping all non-binary valued syntax elements into a string of binary symbols. In CABAC encoder, this task is performed by the binarizer. The mapping algorithms that are used in the binarizer strongly influence coding efficiency as well as complexity of entropy codec. Therefore, CABAC takes advantage of adaptive binarization with several different binarization schemes for all coded syntax elements.

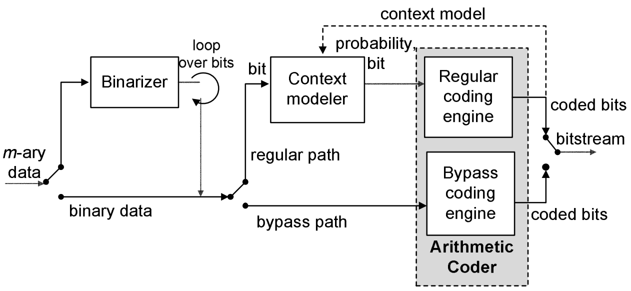

Figure 1 – Block diagram of CABAC encoder.

The core of binary arithmetic codec encodes binary symbols with respect to the probabilities of their occurrence in the video data stream. The probabilities of binary symbols are estimated by the context modeler. The way of calculating these probabilities has a great influence on compression ratio of entropy encoder. Therefore, in order to obtain accurate adaptation to the current signal statistics, 399 individual finite-state machines are used by the context modeler (this is only for the case of a transform calculated in 4×4 blocks) – see Figure 2.

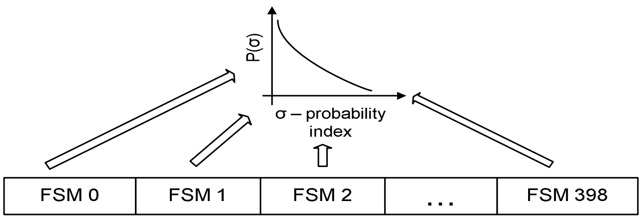

Figure 2 – Definition of finite-state machines in CABAC context modeler.

An individual finite-state machine calculates probabilities of symbols for an individual context. The context depends on the currently coded syntax element (e.g. *mb_type*, *mb_qp_delta* etc.) and its values in neighboring blocks within an image. In this way CABAC realizes two levels of adaptation to the current signal statistics.

The mechanisms of binarization and data statistics estimation are quite advanced and they constitute the major part of the whole CABAC codec (about 90% of whole CABAC implementation) [10].

Nevertheless, the main limitation of CABAC, that negatively affects its compression ratio, is related to only one predefined probability distribution used by all 399 finite-state machines (see Figure 2). Moreover, the same predefined probability distribution is used for higher bitrates as well as for lower bitrates. This simplification surely reduces the ability of CABAC to precisely adapt to current signal statistics. Therefore, an extremely interesting research problem is related to the question about possibility of inceasing compression ratio of CABAC by application of even more exact mechanisms of data statistics estimation.

3. IMPROVED STATISTICAL MODELING IN MODIFIED CABAC

The main idea of improved CABAC algorithm is to replace the existing (i.e. used in original CABAC) methods of data statistics modeling by more advanced mechanisms of data statistics estimation. The idea is to use more sophisticated and more accurate technique of conditional probabilities estimation in the context modeler. Previous works [8,9] of the authors have yielded a conclusion that it can be done by use of the method of Context-Tree Weighting (CTW) [2, 3]. In a modified CABAC codec, other functional blocks i.e, the binarizer and the core of binary arithmetic codec remain unchanged with respect to the original CABAC. The general structure of the proposed improved CABAC encoder has been presented in Figure 3.

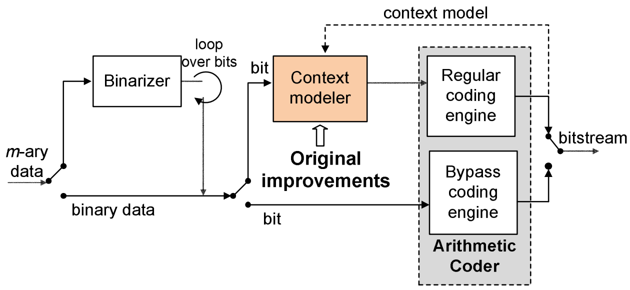

Figure 3 – Block diagram of improved CABAC encoder.

In our proposal, each of the 399 finite state machines has been replaced with a binary context tree of depth $D = 8$ (see Figure 4).

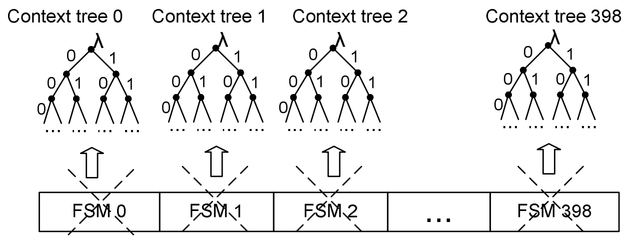

Figure 4 – Context trees in improved context modeller.

The choice of depth D of context trees has been experimentally investigated by authors. The conclusion was that depth $D = 8$ is a good compromise between compression ratio and complexity of the improved CABAC codec [8, 9, 10]. Higher depths result in negligible increase of compression performance but substantial increase of codec complexity. Context trees are used to store information about symbol statistics of binary source data. In the improved context modeler, context trees keep statistics of binary symbols that appeared in video data stream for a given context. For a given context, probabilities of binary symbols 0 and 1 are calculated with CTW method based on the statistics of symbols saved in context trees. The way of choosing a proper context (and context tree thereby) is the same as the way of choosing a proper finite stae machine in the original CABAC. Furthermore, uniform probability distribution is assumed for some binary symbols likewise in the original CABAC.

In general, the improved context modeler generates values of conditional probabilities in a significantly greater set of numbers as compared to the original CABAC. The problem is that the core of fast binary arithmetic codec (M-codec) is not able to work properly with this extended set of probabilities (it has been adopted to work properly with a limited set of only 128 predefined values of probabilities). Therefore, each probability calculated with the improved context modeler has to be mapped to a value from smaller set of 128 predefined probabilities. Criterion of minimization of absolute difference between two probabilities has been assumed by authors in implementation of the improved CABAC.

4. INITIALIZATION OF CONTEXT TREES

The algorithm of context tree initialization influences the compression ratio of the improved CABAC encoder. In [8, 9] a prototype architecture of the modified CABAC codec has been tested with two relatively simple mechanisms of context trees initialization. The current version of the improved CABAC codec has been adapted to work with more advanced mechanism of context trees initialization that is based on QP- and slice-dependent context initialization used in the original CABAC. The slice- and QP-dependent contexts initialization of the original CABAC sets the initial probabilities for 0 and 1 symbols for each of 399 defined contexts. In the current version of improved CABAC, the counters of the number of 0 symbol and the number of 1 symbol in roots of 399 context trees are set to values that allow for obtaining of original CABAC initial probabilities for 0 and 1 symbols. The counters of 0 and 1 symbol in remaining nodes of all 399 context trees are initialized to zero. Initialization of context trees is made at the beginning of each slice, as it takes place in the original CABAC.

5. EXPERIMENTAL RESULTS

The proposed improved context modeler with authors' mechanism of data statistics estimation has been implemented and embedded into the structure of CABAC within the reference software JM 10.2 [4] of AVC video codec. In order to obtain reliable experimental results, both video encoder and video decoder have been implemented. In this way, the improved AVC video codec has been obtained. The coding efficiency of the improved AVC video encoder has been explored and compared against the coding efficiency of the original AVC. Video encoders (improved AVC and original AVC) have been compared in the following scenario of experiments:

- Four HD test video sequences have been used: *CITY*, *CREW*, *NIGHT*, *BIGSHIPS* (spatial resolution: 1280x720, temporal resolution: 60 Hz);
- The IBBPBBP… structure of group of pictures (GOP) has been assumed;
- Experiments have been done for a wide range of quantization parameter (QP) values, from QP = 20 to QP = 44 with step 6. The QP parameter is responsible for the quality of decoded video sequence. For a given QP value, experiments have been done by encoding and decoding of 298 frames of each test sequence;
- Both the improved and the original AVC encoders have been tested with rate-distortion optimization switched off. In this way, the entropy encoders can be directly compared because they do not affect the control mechanism of video encoder (two sequences decoded with the improved and the original AVC decoders are identical and only sizes of encoded bitstreams are different).

Experimental results showed better compression ratio of the improved AVC encoder in comparison to the original AVC video encoder. Average experimental results obtained for *CITY*, *CREW*, *NIGHT* and *BIGSHIPS* test video sequences have been presented in Figure 5 (for I-slices only, P-slices only, B-slices only and for whole test sequences). Figure 6 presents results obtained for individual test sequences.

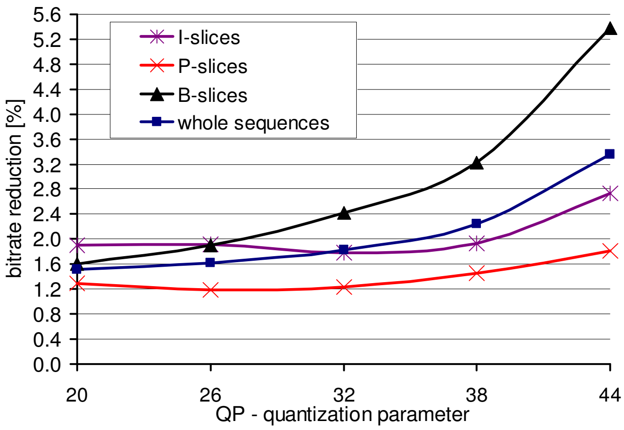

Figure 5 – Average bitrate reduction due to application of the improved AVC encoder instead of the original AVC encoder.

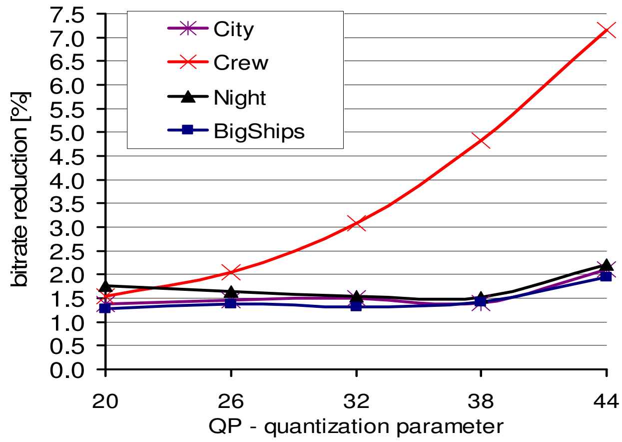

Figure 6 – Bitrate reduction using the improved AVC encoder compared to the original AVC encoder – calculated for individual test video sequences.

The obtained bitrate reduction is strongly dependent on the value of QP parameter. The best results have been observed for lower bitrates that correspond to higher values of QP. Different results have been achieved for individual slice types. The worst results have been obtained for P-slices (1.2% - 2.0% bitrate reduction) and the best results have been observed for B-slices (1.6% - 5.4% bitrate reduction). On average, bitrate reduction of 1.6% - 3.4% has been achieved for test video sequences. The gap in coding efficiency between the improved and the original AVC video encoders is also different for individual test sequences (see Figure 6). The best results have been obtained for *CREW* sequence. Therefore, efficiency of both improved and original AVC encoders depends on the content of test video sequence.

6. COMPLEXITY OF THE MODIFIED CABAC

Higher compression ratio of improved AVC codec relative to the original AVC codec is burdened with higher complexity of both improved encoder and improved decoder in comparison to the original ones. Of course, higher complexity of the improved AVC codec is a result of application of more sophisticated context modeler in CABAC. In order to test the influence of application of improved context modeler on complexity of whole AVC video decoder, some experiments have been done. In comparison to video encoders, a video decoder is much often used by the end users. Therefore, complexity tests have been done for AVC decoders. As a matter of fact, preliminary complexity tests of the prototype version of improved AVC decoder have been already presented in [8]. Nevertheless, complexity tests presented in this paper concern the current version of improved AVC with three significant modifications (from the complexity point of view) relative to the prototype improved AVC:

- Current version of improved AVC works with more advanced mechanism of context trees initialization;
- Different core of arithmetic codec is used in the current version of improved AVC;
- Mechanism of context trees update has been algorithmically optimized in presented improved AVC.

Complexity of the improved AVC decoder has been measured by the effort of the processor. Experiments have been done in the following scenario:

- Two test video sequences have been used in tests: *CREW* and *NIGHT*. Experiments have been done for QP parameter values changing from 20 to 44 with step equal to 6. In this way experiments on complexity of the improved AVC decoder have been done for a wide range of bitrates from excellent (QP = 20) to very poor subjective quality of decoded video (QP = 44);
- Tests have been done on Intel Core 2 Duo E6600 platform (2.4 GHz, 4MB of Level 2 memory cache) with 2 GB of RAM under 32-bit Windows XP with Service Pack 2 operation system.

Realtive increase of the total decoding time of the improved AVC with respect to the total decoding time of the original AVC has been presented in Figure 7 for *CREW* and *NIGHT* test sequences.

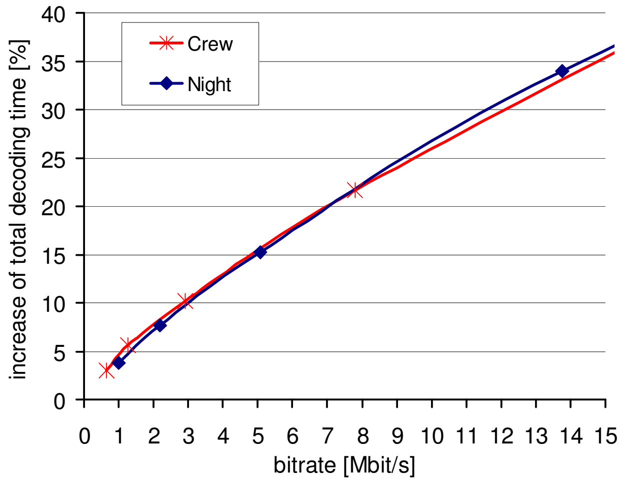

Figure 7 – Increase of total decoding time of the improved AVC decoder relative to total decoding time of the original AVC decoder.

From experimental results it is clear that the target bitrate of encoded video sequence has a significant impact on increase of total decoding time for the improved AVC decoder. For useful range of bitrates (less than 8 - 10 Mbit/s for HD sequences) maximum relative increase of total decoding time for improved AVC is below 25% relative to the original AVC decoder. For useful range of bitrates for standard definition (SD) sequences (less than 3 Mbit/s), the relative increase of total decoding time due to application of improved AVC decoder is below 10%.

7. FURTHER IMPROVEMENTS

The mechanism of context tree initialization and the frequency of resetting of contexts to default values influence compression ratio of the improved AVC encoder. Presented in this paper version of improved AVC codec performs context tree initialization each time before a new slice, likewise in the original AVC codec. For the improved AVC codec, the authors have also investigated the idea of extended slices where one slice contains one or more consecutive pictures of the same type. It must be stated that such an approach is not compliant with AVC video coding standard. In the considered structure of GOP (IBBPBBP…), a given sequence contains regular slice with an I-picture, regular slice with a P-picture and the extended slice with two consecutive B-pictures. Because of much smaller numbers of bits in B-picture representations, such an approach is reasonable. In this approach, context trees are initialized at the beginning of a regular slice or at the beginning of an extended slice. So, in the case of the extended slice, context trees are initialized only before the first B-picture from a pair (see assumed structure of GOP). In this way, the second B-picture from pair takes advantage of data statistics collected in the previous B-picture.

In Figure 8 presented are experimental results for improved AVC encoder with modified mechanism of context trees initialization from Section 4.

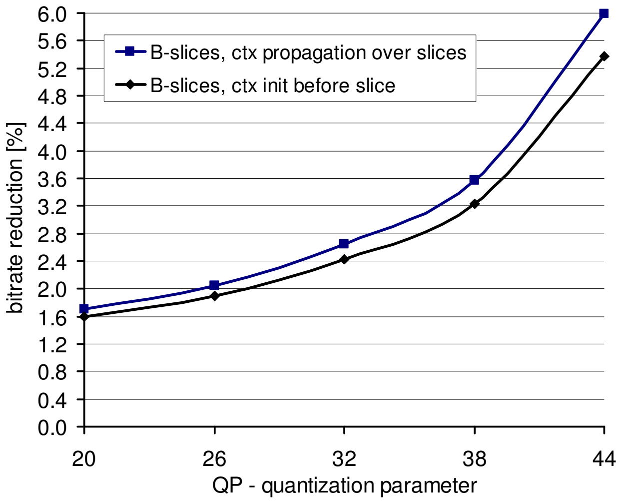

Figure 8 – Influence of frequency of context initialization on compression ratio of the improved AVC video encoder. The results are relative to the standard AVC codec.

Experimental results proved the possibility of further improving compression ratio for B-pictures when using the idea of extended slice that reduces the problem of context dilution. Higher bitrate savings have been observed for lower bitrates.

8. CONCLUSIONS

Presented in the paper experimental results proved that application of more sophisticated mechanisms of adaptation of arithmetic coding increases compression ratio of contemporary adaptive arithmetic coders used in video compression. Experiments have been done in the context of the state-of-the-art CABAC algorithm that is used in the newest AVC video coding standard. The authors have presented the new architecture of improved CABAC codec that uses proposed original technique of data statistics estimation. The proposed improved CABAC codec outperforms the original CABAC codec by 1.2% - 5.4% when working within the framework of AVC video encoder.

The authors answered also another important question, how much the application of more advanced mechanisms of the conditional probabilities estimation influences complexity of the improved AVC video decoder. Algorithmically optimized implementation of the improved AVC decoder has been used for tests. Obtained results showed that total decoding time for improved AVC is approximately 5-10% and 15-25% greater relative to the original AVC decoder for SD and HD video sequences, respectively.

ACKNOWLEDGMENT

The work was supported by the public founds as a research project in year 2009.

REFERENCES

[1] 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.

[2] F. M. J. Willems, Y. M. Shtarkov, and Tj. J. Tjalkens, "The Context-Tree Weighting Method: Basic Properties". *IEEE Transactions on Information Theory*, Vol. 41, No. 3, pp. 653-664, May 1995.

[3] F. M. J. Willems, "The Context-Tree Weighting Method: Extensions". *IEEE Transactions on Information Theory*, Vol. 44, No. 2, pp. 792-797, March 1998.

[4] H.264/AVC software coordination site – http://bs.hhi.de/~suehring/tml.

[5] D. Marpe, G. Marten, and H. L. Cycon, "A Fast Renormalization Technique for H.264/MPEG4-AVC Arithmetic Coding". *51st Internationales Wissenschaftliches Kolloquium Technische Universitat Ilmenau*, September 2006.

[6] ISO/IEC 14496-10, Generic Coding of Audio-Visual Objects, Part 10: Advanced Video Coding, March 2006.

[7] F. M. J. Willems, and Tj. J. Tjalkens, "Complexity Reduction of the Context-Tree Weighting Algorithm: A study for KPN research". *Technical Report EIDMA-RS.97.01, Euler Institute for Discrete Mathematics and its Applications*, Eindhoven University of Technology, 1997.

[8] D. Karwowski, "Improved Arithmetic Coding in H.264/AVC Using Context-Tree Weighting and Prediction by Partial Matching". *European Signal Processing Conf. EUSIPCO 2007*, pp. 1270-1274, September 2007, Poznań, Poland.

[9] D. Karwowski, and M. Domański, "Improved Arithmetic Coding In H.264/AVC Using Context-Tree Weighting Method". *Picture Coding Symposium PCS 2007*, November 2007, Lisboa, Portugal.

[10] D. Karwowski, "Advanced Adaptation Algorithms of Arithmetic Coding in Hybrid Video Compression". *Doctoral Dissertation*, Poznań University of Technology, 2008.

[11] D. Karwowski, M. Domański, ITU-T VCEG SG 16 Q.6, Doc. VCEG-AI15, "Improved Context-based Adaptive Binary Arithmetic Coding using Context-Tree Weighting", *35rd VCEG Meeting*, Berlin, Germany, 16-18 July 2008.

[12] D. Salomon, *Data Compression. The Complete Reference – 4th Edition*, Springer-Verlag, 2006.

[13] D. Hong, M. van der Schaar, B. Pesquet – Popescu, "Arithmetic Coding with Adaptive Context-Tree Weighting for the H.264 Video Coders", *Visual Communications and Image Processing 2004*, Vol. 5308, pp. 1226-1235, January 2004.

[14] M. H. Firooz, M. S. Sadri, "Improving H.264/AVC Entropy Coding Engine Using CTW Method", *Picture Coding Symposium*, April 2006.