IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received March 12, 2021, accepted April 20, 2021, date of publication April 26, 2021, date of current version May 6, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3075875

Precise Probability Estimation of Symbols in VVC CABAC Entropy Encoder

DAMIAN KARWOWSKI

Faculty of Computing and Telecommunications, Poznan University of Technology, 60-965 Poznań, Poland

e-mail: damian.karwowski@put.poznan.pl

This work was supported by the Ministry of Education and Science, Poland.

ABSTRACT The new version of the Context-based Adaptive Binary Arithmetic Coding (CABAC) that is used in the latest Versatile Video Coding (VVC) technology is the state-of-the-art in the field of entropy compression of video data. This paper gives an in-depth analysis of possible, further extensions of the CABAC technique for even more accurate estimation of the probabilities of coded data symbols. The paper considers four author's examples of such extensions. With respect to these examples, the paper shows the practical limitations of further increasing the efficiency of the newest version of CABAC used in VVC. Results show that for the considered methods of improvements the limit of gain in compression is up to 0.2073% on average, when compressing the VVC video data.

INDEX TERMS Probability estimation, entropy coding and arithmetic coding, VVC CABAC, video compression.

I. INTRODUCTION

Today, as much as 80% of all the data that is transmitted in IT networks represents video. Despite this, the percentage of video data continues to further increase. To make it possible to use video data in a such a large scale, it must be compressed before transmission or storage. The compression of video is performed by a video encoder.

Currently, hybrid video encoders have the greatest practical use [1]–[7]. These encoders can represent video data in a very efficient manner by describing this data with specially designed syntax elements (instead of direct encoding of video samples) (see Fig. 1). Nevertheless, the syntax elements data usually still exhibit some statistical redundancy. Therefore, to reduce this redundancy entropy coding is always used as the final stage of video compression. Entropy coding is a lossless coding and its application in a video encoder further increases the compression performance of video data [8]. Thus the high importance of this encoding technique in video compression. Entropy coding is the subject of interest in this work.

In connection with the above entropy coding has been the object of intensive scientific studies for years. Such works have especially been pursued in the context of video data compression. When referring to a high percentage of video data usage, it is obvious that the improvement of compression performance of this type of data by even a fraction of percent gives very large savings of the bandwidth, when viewed in terms of total traffic in IT networks. This conclusion explains the high need for such research. A milestone in this research was the development of the Context-based Adaptive Binary Arithmetic Coding (CABAC) technique in the early 2000s [9]. This technique is a very sophisticated method of applying arithmetic coding [10], [11], [16], [34]–[38] for entropy compression of syntax elements data in a video encoder. The CABAC technique ranks among the most efficient entropy encoders for video data. This is why it has found practical application in the three most advanced video compression technologies, developed

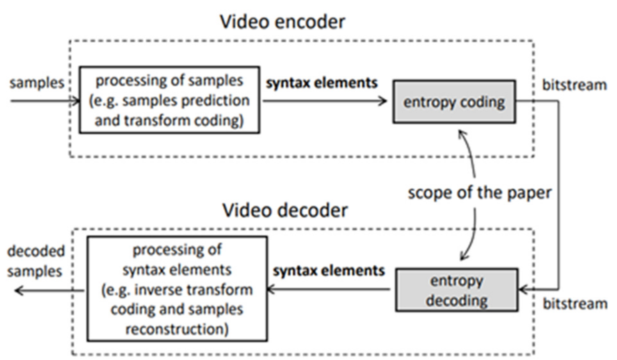

FIGURE 1. A simple block diagram of a video encoder and video decoder. Entropy coding is of interest to this paper.

The associate editor coordinating the review of this manuscript and approving it for publication was Gangyi Jiang.

in the last two decades. Those are: 1) H.264/MPEG-4 AVC (Advanced Video Coding) [2], [3] that is currently widely used in television and Internet, 2) H.265/MPEG-H HEVC (High Efficiency Video Coding) [6], [7], which is the successor of AVC, and 3) the newest Versatile Video Coding (VVC) technology [12] that has been jointly developed by the ISO/IEC JTC 1/SC 29/W G11 Moving Picture Experts Group (MPEG) and ITU-T SG16 Q6 Video Coding Experts Group (VCEG) [13]. It must be stated here, that the latter VVC video compression technology (that has emerged in the middle of 2020) contains the most efficient version of the CABAC technique [14], which was developed in the past few years through numerous improvements of versions of CABAC used before in the H.264/MPEG-4 AVC [9] and H.265/MPEG-H HEVC technologies [15].

The CABAC technique is therefore the starting point for the most advanced works carried out in recent years in the domain of entropy compression of video data. It is also the case with this paper. In the studies presented here, the most sophisticated version of CABAC used in the VVC video coding was used as the starting point. The aim of the paper is to point out the possible further technical improvements of the CABAC technique together with the author's analysis of potential benefits in compression efficiency. The paper considers four such author's improvements.

II. RATIONALE, LITERATURE REVIEW AND SPECIFIC GOALS OF THE WORK

A. KEY ISSUES OF THE CABAC TECHNIQUE

The CABAC entropy encoder [9], [15] performs the lossless compression of syntax elements data that are produced by earlier stages of video compression. In the case of hybrid video compression those syntax elements in general represent three types of information: 1) motion data (information on motion in a video), 2) residual data from video samples prediction (information on prediction error of video samples), and 3) control data (information on the coding mode of blocks of video frames). The efficient compression of the mentioned data is very difficult. It is due several reasons. The statistical nature of each of these three types of data is generally different. Within a given group of data, statistics of data symbols are dependent on the content of a video, so they can change from video frame to video frame, and even within a single frame. The configuration of the video encoder (e.g. encoding with high or low quality of encoded video) also affects the data statistics. All these factors require the use of complex mechanisms of data processing in the entropy encoder. And this is indeed the case with the CABAC technique. In particular, it concerns the method of computing the probabilities for the data symbols.

To do it precisely in CABAC, the syntax elements are mapped to strings of binary symbols – the so-called bins (see Fig. 2). The mapping is realized by the binarizer – the first functional block of CABAC. For this purpose, the binarizer uses specially selected methods of ''translating'' the syntax elements into bins (several different binarization schemes). In the next step (probability modelling) the stream of bins is split into a high number of separate sub-streams in order to gather together those bins which show similar statistical properties. In the newest version of CABAC that is used in the VVC video encoder there may be as many as nearly 1100 independent sub-streams of bins! The goal of this split is to calculate the probabilities of bins independently for ''statistically similar'' parts of the stream (so, independently for the individual sub-streams). The premise for the method of splitting the stream of bins into smaller sub-streams is the type of a syntax element, its value in neighboring image blocks (relative to the block that is currently being encoded) and the position of the bin in the binarized word. So, this part of CABAC is really very complex. In the final step of probability modelling, CABAC estimates the probability of each bin, when taking into consideration dependencies that can be seen between the individual bins (within a sub-stream). The probabilities are finally used by an arithmetic encoder core to perform the compression of bins. Therefore, the way that these probabilities are calculated strongly affects the efficiency of arithmetic encoding. For this reason, this step of CABAC technique is the scope of this paper. The topic of bins' probabilities estimation will be continued below.

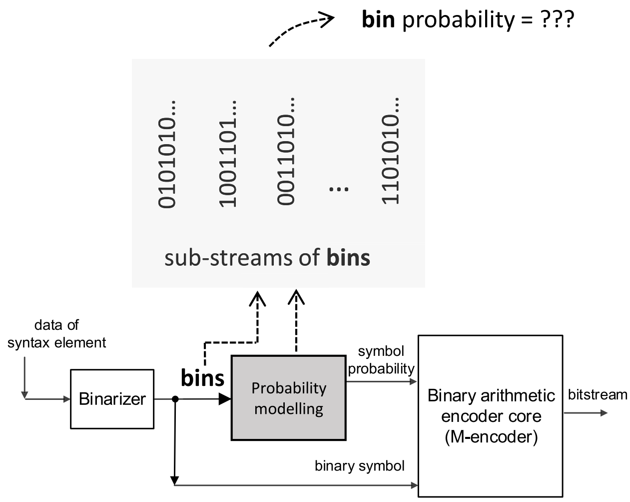

FIGURE 2. General block diagram of the CABAC technique.

B. CABAC TECHNIQUE–BIN PROBABILITY ESTIMATION

In the first versions of CABAC (the ones used before in the H.264/MPEG-4 AVC [9] and H.265/MPEG-H HEVC [15] video encoders) the probabilities of bins were calculated when assuming the ''exponential aging'' model of data statistics [16]. According to this model, the probability that the current bin will be 0 or 1 depends on the values of previous bins that were encoded before.

However, the impact (on the probability value of the current bin) of each of the previously encoded bins is not the same - the influence of bins that immediately precede the currently encoded bin is stronger than that of bins that had appeared long ago. To take this into consideration, the α parameter is used in CABAC, which influences the number N of previously encoded bins that are actually used to estimate the probability of the current bin, according to the dependency $N = 1/\alpha$. It translates into the speed with which the symbol probability values are updated as a result of changing the value of the coded bins (the so-called adaptation speed). The α parameter is used in the mathematical formula of determining the probability of the new bin. If the new bin is the least probable symbol (LPS) the formula takes form:

$$p_{new}\left(LPS\right) = \alpha \cdot y\left(bin\right) + \left(1-\alpha\right) \cdot p_{old}\left(LPS\right) \quad (1)$$

where:

$p_{new}\left(LPS\right)$ is the estimated probability of the LPS symbol,

$p_{old}\left(LPS\right)$ is the probability of the LPS symbol calculated earlier,

$y\left(bin\right)$ is equal to 0 if the current bin is an MPS symbol, and equal to 1 otherwise.

In the older versions of CABAC [9], [15] formula (1) was independently used for each sub-stream of bins when assuming a fixed value of $\alpha \approx 1/19.69$ [17]. So, more or less, the number of 19.69 previously encoded bins were taken into account when estimating probability of the new bin.

C. BIN PROBABILITY ESTIMATION IN CABAC–LATEST IMPROVEMENTS

Nevertheless, it is obvious that the use of one fixed value of α in each of the sub-streams is far from optimal. It has been proven in scientific research. Better results of compression efficiency of bins can be obtained when using multi-parameter probability estimation instead of one using a single parameter only. The authors of [18], [19] have proposed an improvement of CABAC in which two different values of α are used (α_1 and α_2), that correspond to slow as well as fast speed of adaptation of probabilities. This way, two different values of probabilities are calculated for a bin, p_{α_1} and p_{α_2}, which are finally weighted together (see Equation 2) with fixed weighting factors w_1 and w_2 equal to 1/2, to produce the probability of a bin.

$$p\left(bin\right) = w_1 \cdot p_{\alpha_1} + w_2 \cdot p_{\alpha_2} \quad (2)$$

where: $w_1 + w_2 = 1$

The values of the two parameters were still fixed ($\alpha_1 = 1/16$ and $\alpha_2 = 1/256$) and the same for each of the CABAC sub-streams. Despite this simplification, this improvement helped increase compression efficiency of video data by 0.5%-0.8% when using it in the HEVC video encoder [18].

These results have motivated further works on the improvement of the probability estimation of symbols in the CABAC technique. The above-mentioned two-probability model update was further extended in order to use specific, independent values of parameters α_1 and α_2 for each of the CABAC sub-streams [14]. In practice, each of the sub-stream uses one specific value for parameter α_1 from the set $\{1/4, 1/8, 1/16, 1/32\}$ and one value of α_2 from the second set $\{1/32, 1/64, 1/128, 1/256, 1/512\}$. The proper values of parameters α_1 and α_2 for each of the sub-stream were determined experimentally.

The version of the CABAC algorithm that uses this improvement is the state-of-the-art in the field of probability estimation of symbols in contemporary arithmetic encoders [14]. This newest version of CABAC has become part of the latest international video compression technology, named Versatile Video Coding (VVC), and is called VVC CABAC in this paper [12]. For this reason VVC CABAC makes the starting point for studies presented in this paper.

D. BIN PROBABILITY ESTIMATION IN CABAC–OTHER LITERATURE PROPOSALS

In the literature, other works on improving CABAC (that those referenced in the previous sub-section) can also be found. However, the idea of the improvement is much different from that highlighted above. Additionally, they are concerned with the older versions of CABAC that found practical applications in coding technologies preceding VVC video technique (i.e., in H.264/MPEG-4 AVC [9] or H.265/MPEG-H HEVC). Numerous such works by the author can be mentioned here [20]–[23], in which the Context-Tree Weighting (CTW) method [24], [25] was originally used to estimate the probabilities of data symbols more precisely. Recently, another idea of improving CABAC has also been proposed by the author [26]. In the latter case, the basis of the proposal is to use the Cauchy optimization method, in order to further minimize the number of bits that are observed at the output of the CABAC entropy encoder. Those extensions of CABAC have led to a noticeable reduction of the video data stream by 0.6% to 1.2%, making these proposals competitive to other literature proposals [e.g., 27].

Just recently, solutions that use artificial neural networks have started to be popular. In the case of arithmetic coding of data such studies should be considered as preliminary. Although they are promising, they only relate to single, selected syntax elements [28]–[30]. Therefore, in practice, it is still difficult to assess the real impact of these solutions on the efficiency of the whole entropy encoder.

E. SPECIFIC GOALS OF THE PAPER

The studies presented in this paper concern the newest version of CABAC, called VVC CABAC, that has been pointed out in sub-section C. The goal of this paper is to provide an in-depth analysis of possible, further extensions of the method of estimating probabilities of bins in the VVC CABAC technique. The paper presents four author's examples of such extensions, which are to be presented in the next sections of the paper. With respect to these examples, the paper aims to show the practical limitations of further increasing the efficiency of the VVC CABAC. This efficiency is understood as the ability of the entropy encoder to reduce the number of bits that describe video data in the VVC video encoder.

III. AUTHOR'S VARIANTS OF EXTENDING THE VVC CABAC TECHNIQUE

The advantage of VVC CABAC (over the older versions of CABAC) results from using two different dedicated values of the α parameter (α_1 and α_2) in each of the sub-streams. To achieve the goal of the paper, the author has considered four variants that could potentially further extend the VVC CABAC technique. The basis of the first variant (called VVC CABAC_0.5+) is to apply such two dedicated values α_1 and α_2 in each of the sub-streams, which are optimal (in terms of compression efficiency) for the content of the encoded test video sequence. Therefore, a dedicated set of α_1 and α_2 parameters is used for each video sequence. The second variant (named VVC CABAC+) extends this idea further by introducing dedicated, optimal values of α_1 and α_2 not only for the encoded video, but also for a given slice type (I, B0, B1, B2, B3 types of slice).

In these variants, the same as in the VVC CABAC technique, the probability of a bin is just a simple average of the two probabilities calculated for the values α_1 and α_2. The third and the fourth variants further extend the earlier ones that the obtained two probabilities are specially weighted together with the use of dedicated weighting factors w_1 and w_2. Those weighting factors have been independently selected for each of the sub-streams to be optimal (from the compression efficiency point of view) for the encoded sequence (third variant) or for the encoded sequence and type of slice (fourth variant). The obtained third variant of the VVC CABAC extension is called VVC CABAC_0.5++ in this paper, while variant four of the extension of VVC CABAC is named VVC CABAC++.

For the four solutions cited above, the optimal α_1 and α_2 parameters are selected from those that are in the sets {1/4, 1/8, 1/16, 1/32} and {1/32, 1/64, 1/128, 1/256, 1/512} respectively (these are original sets of α parameters, used in the VVC CABAC technique), when calculating the entropy for a set of sub-streams data that were previously produced by the VTM 7.2 software for a given test sequence. Depending on the variant of the VVC CABAC extension, the entropy of video data is calculated independently for each of the sub-streams, slice types and possible weighting factors w_1 and w_2 that come from the set {0, 0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9, 1}. Values α_1 and α_2 and values of weighting factors w_1 and w_2 that minimize entropy of compressed data are considered the optimal ones. The compression efficiency capabilities of the four above-mentioned variants of the VVC CABAC extension is the subject of analysis in the next, experimental part of the paper.

IV. ANALYSIS OF POSSIBILITIES OF IMPROVING THE VVC CABAC TECHNIQUE

A. METHODOLOGY OF EXPERIMENTS

1) MAIN ASSUMPTIONS

The goal of the experiments was to assess what are the possibilities of increasing the compression efficiency of the VVC CABAC technique when the previously presented variants of improvements are applied in this technique. For this purpose the author has prepared three separate software packages. They were implemented in the C/C++ programming language. The first package is the software that extracts individual sub-streams of bins that are produced by the VVC CABAC within the successive video frames, when operating in the VVC video encoder. The second software tool calculates the cost (expressed in bits) of entropy compression of the sub-streams data when using a given author's variant of the VVC CABAC extension, and selects the optimal parameters (from the viewpoint of compression efficiency) for the considered variant of the VVC CABAC extension. The third piece of software implements a given variant of the VVC CABAC extension (together with its optimal parameters that were determined with the second software) and measures the compression efficiency of a set of bins which were previously gathered with the first software. At this point, it is assumed that the extended VVC CABAC technique, which is the subject of experiments, already uses optimal values of its parameters (determined on the basis of the data being compressed). In practical application, two separate phases of data processing should be taken into account: the so-called learning phase, in which, based on the set of training data, optimal parameter values are determined, and the appropriate data compression phase. Due to the adopted assumption, the results presented in this paper the author considers as the limited values of the compression gain.

2) SOFTWARE IMPLEMENTATION AND ITS OPERATION

The first software tool has been implemented on top of the VTM 7.2 reference software [31] of the VVC video codec [12]. When decoding the encoded VVC data streams the prepared software extracts first all the syntax elements, and then the individual sub-streams of bins, and writes the sub-streams to separate files. That is, each file contains bins (binary symbols) which were the subject of arithmetic coding in VVC CABAC, within a given sub-stream and a given slice type (slices I, B0, B1, B2, B3).

The above-mentioned files are then the input to the second and the third author's software. The second software reads the bins from files, and calculates the bit cost of entropy compression of bins, when assuming a given author's variant of the VVC CABAC extension and specific values of parameters for the considered extension. The task of the second software is to find the optimal parameter values (from the point of view of CABAC coding efficiency) for a given extension of VVC CABAC. Having determined the optimal parameter values, the third software reads the bins from the files prepared by the first software and after the estimating probabilities of the bins (according to a given variant of the VVC CABAC extension) calculates the minimum bit cost of entropy compression of bins. The minimum bit cost of bin compression is calculated with the following formula:

$$Cost\,(bin) = -log_2\,(p\,(bin)) \tag{3}$$

where $p\,(bin)$ is the probability of bin.

The sum of the bit costs obtained for successive bins that are encoded within individual CABAC sub-streams makes the final number of bits produced by the entropy encoder. In this research, this number of bits is referred to the number of encoded bins (bits to bins ratio – see tables below) to determine the efficiency of arithmetic encoding in a given variant of VVC CABAC extension as well as the efficiency of arithmetic encoding in the original HEVC CABAC and VVC CABAC algorithms. The smaller the ratio, the stronger compression of data. As reported in the literature this method of estimating the efficiency of arithmetic encoding is practiced and used in video compression [e.g., 32]. Comparing the ratios for the two different methods (e.g., VVC CABAC to HEVC CABAC) allows to determine the percentage reduction of bitrate, after applying a more efficient solution.

3) PREPARATION OF TEST DATA

The goal was to carry out the research when compressing syntax elements that are present in the newest VVC video encoding. To fulfil this requirement the VTM 7.2 software was used in order to prepare the VVC-encoded data streams. These VVC data streams were then the input data to the first software of the author, which has been highlighted in the previous sub-section of the paper.

With the help of the VTM 7.2 software, three sets of test video sequences were encoded. In the first set, SD (Standard Definition) test sequences were used. In the second set 2K test sequences were used, whilst sequences with resolutions beyond 2K were used in the third set. These were the sequences: *BasketballDrill*, *BQMall*, *PartyScene*, *RaceHorses* (in 832 × 480 resolution), *BQTerrace*, *Cactus*, *Kimono1*, *ParkScene* (each in 2K resolution), and *PeopleOnStreet*, *Traffic* (in resolutions beyond 2K). All of them have been recommended by the ISO/IEC MPEG and ITU-T VCEG groups of experts as the proper testing material for studies in the field of video compression. The goal of the research was to perform the analysis under the scenario of video encoding with high compression efficiency (such a scenario is usually used in ''video on demand'' and ''video streaming'' applications). Thus, in the research the VTM 7.2 software was configured according to the ''random access'' scenario of video encoding that has been defined by international groups of experts (ISO/IEC MPEG and ITU-T VCEG) in the ''common test conditions'' recommendation [33]. This encoding scenario involves the use of computationally complex but very efficient compression tools in the video encoder. Following this recommendation, each of the test sequences was encoded four times, each time with a different value of the quantization parameter (QP) set in the VTM 7.2 software. In the research 22, 27, 32, 37 values of QP were considered. Those values of QP correspond to video encoding from high (QP=22) to low (QP=37) subjective quality of video frames after compression and decompression. The result of this part of experiments were sequences encoded with the VVC video encoder. They contain syntax elements compressed with the VVC CABAC technique. The encoded sequences made the input to the first author's software mentioned above.

B. EXPERIMENTAL RESULTS

This section presents the results of experiments that were carried out for the four extensions of the VVC CABAC technique considered in this paper. The results of the experiments include data on the compression efficiency of each variant of VVC CABAC extension, as well as the comparison of this efficiency to the results of the original VVC CABAC technique. In order to better evaluate the presented data it is worth referring them to those presenting the efficiency of the older version of CABAC, used in H.265/MPEG-H HEVC (denoted as HEVC CABAC in the tables presented in this paper) as well as the data of the compression gain of VVC CABAC over HEVC CABAC.

1) FIRST VARIANT OF VVC CABAC EXTENSION–VVC CABAC_0.5+

When extending VVC CABAC so that it uses the optimal values of α_1 and α_2 for both the individual sub-streams, the test sequence and the value of QP, only a very small improvement can be observed. The average reduction of bitrate (and so the gain of compression) is only 0.08241% for SD sequences, 0.07623% for 2K sequences and 0.1224% for sequences with higher resolutions. The detailed results for individual sequences were presented in Tables 1, 2 and 3. Interestingly, in most cases the higher the QP, the higher the gain of compression. The differences in percentage results between the highest and the lowest QP used in the experiments amount to 0.01% to 0.06%. Those values seems to be small, however, they may be considered as significant when referring the data to the mentioned values of average reduction of bitrate.

Better results obtained for higher QPs highlights the fact that the original VVC CABAC technique uses values of α_1 and α_2 parameters that are not so well adjusted to the ''nature'' of syntax elements data that occur for large QP values. The additional element that influences the results is the content of the test sequence. Different results were obtained for individual sequences. However, noticeably better results were obtained for sequences with resolutions beyond 2K. According to the author's opinion, this fact can be explained as follows. The VVC CABAC_0.5+ algorithm (as well as the original VVC CABAC) is initialized (reset) to its default settings each time at the start of a new slice. In the experiments carried out in this work, a slice is a single video frame. The higher the spatial resolution of a video frame, the more syntax elements data there is within one video frame that is compressed with the entropy encoder. It translates into a bigger set of bins that are compressed between two consecutive initializations (resets) of VVC CABAC_0.5+ (as well as the original VVC CABAC). As it is known, each reset of the entropy encoder negatively affects the compression performance of data.

2) SECOND VARIANT OF VVC CABAC EXTENSION–VVC CABAC+

When extending the first variant further by introducing dedicated, optimal values of α_1 and α_2 also for a given slice type (I, B0, B1, B2, B3 types of slice), noticeably higher compression efficiency can be observed. The corresponding results are shown in columns of Tables 1, 2 and 3

TABLE 1. Results of experiments for SD test video sequences.

		original algorithms			considered improvements to VVC CABAC				bitrate reduction [%]			
		bits to bins		bitrate reduction [%]	bits to bins				VVC CABAC_0.5+ to VVC CABAC	VVC CABAC+ to VVC CABAC	VVC CABAC_0.5++ to VVC CABAC	VVC CABAC++ to VVC CABAC
SD video		HEVC CABAC	VVC CABAC	VVC CABAC to HEVC CABAC	VVC CABAC_0.5+	VVC CABAC+	VVC CABAC_0.5++	VVC CABAC++				
BasketballDrill	QP = 22	0.75712	0.75142	0.75220	0.75089	0.75053	0.75063	0.75015	0.07040	0.11858	0.10540	0.16928
	QP = 27	0.75771	0.75211	0.73920	0.75146	0.75091	0.75114	0.75042	0.08616	0.15836	0.12871	0.22404
	QP = 32	0.75174	0.74632	0.72179	0.74549	0.74484	0.74508	0.74421	0.11094	0.19844	0.16535	0.28312
	QP = 37	0.74421	0.73906	0.69201	0.73805	0.73719	0.73751	0.73642	0.13653	0.25208	0.20919	0.35721
BQMall	QP = 22	0.74334	0.73761	0.77165	0.73704	0.73671	0.73678	0.73629	0.07741	0.12215	0.11280	0.17814
	QP = 27	0.75395	0.74791	0.80204	0.74729	0.74687	0.74692	0.74636	0.08236	0.13852	0.13143	0.20698
	QP = 32	0.74796	0.74182	0.82130	0.74112	0.74064	0.74070	0.74004	0.09477	0.15907	0.15125	0.23928
	QP = 37	0.73869	0.73292	0.78112	0.73209	0.73149	0.73163	0.73080	0.11215	0.19484	0.17560	0.28830
PartyScene	QP = 22	0.75191	0.74672	0.69104	0.74624	0.74596	0.74598	0.74559	0.06401	0.10124	0.09870	0.15120
	QP = 27	0.76136	0.75571	0.74261	0.75529	0.75499	0.75508	0.75461	0.05505	0.09594	0.08337	0.14516
	QP = 32	0.75941	0.75354	0.77297	0.75314	0.75275	0.75289	0.75235	0.05322	0.10457	0.08626	0.15686
	QP = 37	0.74718	0.74129	0.78736	0.74084	0.74042	0.74055	0.74000	0.06124	0.11750	0.10050	0.17523
RaceHorses	QP = 22	0.70837	0.70320	0.72998	0.70280	0.70263	0.70260	0.70237	0.05717	0.08063	0.08476	0.11860
	QP = 27	0.73866	0.73311	0.75055	0.73266	0.73235	0.73243	0.73198	0.06125	0.10421	0.09289	0.15482
	QP = 32	0.73051	0.72492	0.76617	0.72423	0.72384	0.72393	0.72338	0.09477	0.14788	0.13629	0.21202
	QP = 37	0.70887	0.70385	0.70747	0.70314	0.70256	0.70284	0.70204	0.10116	0.18370	0.14392	0.25673
								Average:	0.08241	0.14236	0.12540	0.20731

TABLE 2. Results of experiments for 2K test video sequences.

		original algorithms			considered improvements to VVC CABAC				bitrate reduction [%]			
		bits to bins		bitrate reduction [%]	bits to bins				VVC CABAC_0.5+ to VVC CABAC	VVC CABAC+ to VVC CABAC	VVC CABAC_0.5++ to VVC CABAC	VVC CABAC++ to VVC CABAC
2K video		HEVC CABAC	VVC CABAC	VVC CABAC to HEVC CABAC	VVC CABAC_0.5+	VVC CABAC+	VVC CABAC_0.5++	VVC CABAC++				
BQTerrace	QP = 22	0.64553	0.64072	0.74512	0.64020	0.64002	0.64007	0.63982	0.08131	0.10878	0.10098	0.14140
	QP = 27	0.69280	0.68758	0.75318	0.68710	0.68681	0.68687	0.68646	0.06937	0.11155	0.10297	0.16362
	QP = 32	0.72019	0.71438	0.80715	0.71376	0.71345	0.71345	0.71299	0.08553	0.12906	0.12990	0.19458
	QP = 37	0.71241	0.70664	0.80979	0.70588	0.70556	0.70545	0.70499	0.10798	0.15241	0.16897	0.23392
Cactus	QP = 22	0.67022	0.66529	0.73587	0.66487	0.66468	0.66472	0.66446	0.06388	0.09259	0.08523	0.12491
	QP = 27	0.72361	0.71788	0.79214	0.71759	0.71729	0.71739	0.71697	0.04068	0.08288	0.06909	0.12648
	QP = 32	0.72541	0.71962	0.79762	0.71928	0.71888	0.71901	0.71848	0.04739	0.10297	0.08532	0.15939
	QP = 37	0.71945	0.71395	0.76544	0.71351	0.71306	0.71319	0.71255	0.06149	0.12368	0.10673	0.19511
Kimono1	QP = 22	0.71176	0.70630	0.76740	0.70570	0.70538	0.70547	0.70506	0.08396	0.12983	0.11652	0.17471
	QP = 27	0.72130	0.71597	0.73894	0.71523	0.71489	0.71493	0.71444	0.10336	0.15182	0.14512	0.21439
	QP = 32	0.71107	0.70601	0.71118	0.70520	0.70478	0.70482	0.70418	0.11572	0.17535	0.16926	0.25920
	QP = 37	0.69490	0.69024	0.67103	0.68945	0.68889	0.68897	0.68815	0.11503	0.19500	0.18399	0.30337
ParkScene	QP = 22	0.71304	0.70727	0.81019	0.70690	0.70672	0.70672	0.70649	0.05203	0.07720	0.07748	0.11057
	QP = 27	0.71657	0.71069	0.82085	0.71023	0.71002	0.71004	0.70974	0.06402	0.09456	0.09202	0.13339
	QP = 32	0.70908	0.70345	0.79483	0.70300	0.70271	0.70277	0.70240	0.06397	0.10434	0.09667	0.14813
	QP = 37	0.69661	0.69123	0.77317	0.69078	0.69044	0.69049	0.69004	0.06394	0.11415	0.10619	0.17100
								Average:	0.07623	0.12164	0.11478	0.17839

TABLE 3. Results of experiments for test video sequences with spatial resolution beyond 2K.

		original algorithms			considered improvements to VVC CABAC				bitrate reduction [%]			
		bits to bins		bitrate reduction [%]	bits to bins				VVC CABAC_0.5+ to VVC CABAC	VVC CABAC+ to VVC CABAC	VVC CABAC_0.5++ to VVC CABAC	VVC CABAC++ to VVC CABAC
> 2K video		HEVC CABAC	VVC CABAC	VVC CABAC to HEVC CABAC	VVC CABAC_0.5+	VVC CABAC+	VVC CABAC_0.5++	VVC CABAC++				
PeopleOnStreet (2560x1600)	QP = 22	0.74102	0.73487	0.83007	0.73396	0.73381	0.73367	0.73347	0.12438	0.14451	0.16425	0.19051
	QP = 27	0.75325	0.74718	0.80597	0.74606	0.74586	0.74571	0.74546	0.14976	0.17720	0.19674	0.23113
	QP = 32	0.75274	0.74698	0.76560	0.74574	0.74545	0.74539	0.74503	0.16627	0.20456	0.21326	0.26119
	QP = 37	0.75172	0.74631	0.71981	0.74502	0.74464	0.74467	0.74421	0.17352	0.22417	0.21961	0.28192
PeopleOnStreet (3840x2160)	QP = 22	0.72490	0.71901	0.81280	0.71828	0.71815	0.71806	0.71789	0.10111	0.11989	0.13240	0.15619
	QP = 27	0.74526	0.73920	0.81341	0.73819	0.73805	0.73792	0.73770	0.13609	0.15598	0.17343	0.20279
	QP = 32	0.74803	0.74216	0.78393	0.74104	0.74084	0.74073	0.74046	0.15131	0.17867	0.19308	0.22906
	QP = 37	0.74625	0.74069	0.74626	0.73950	0.73922	0.73920	0.73884	0.15999	0.19752	0.20036	0.24855
Traffic (3840x2048)	QP = 22	0.71853	0.71222	0.87888	0.71169	0.71151	0.71146	0.71123	0.07428	0.09843	0.10615	0.13830
	QP = 27	0.72378	0.71766	0.84680	0.71715	0.71695	0.71697	0.71670	0.07065	0.09852	0.09601	0.13349
	QP = 32	0.72327	0.71732	0.82265	0.71676	0.71654	0.71657	0.71627	0.07876	0.10874	0.10539	0.14693
	QP = 37	0.71496	0.70916	0.81096	0.70857	0.70832	0.70834	0.70802	0.08306	0.11817	0.11577	0.16104
								Average:	0.1224	0.1522	0.1597	0.1984

referring to VVC CABAC+. For this scenario, compared to the VVC CABAC technique, the average increase in coding efficiency is 0.14236%, 0.122% and 0.152%, for SD, 2K and beyond 2K spatial resolutions of video frames, respectively. When comparing these results with those for the VVC CABAC_0.5+ variant, the additional average gain of compression is 0.05994%, 0.045% for the SD and 2K video, respectively and 0.0298% for sequences with even higher resolutions. As it can be seen, higher additional gain can be observed for the SD and 2K sequences. Similarly as for the VVC CABAC_0.5+ variant, the higher the QP, the higher the increase in coding efficiency of data, when comparing the results with those for the original VVC CABAC. Here, the difference of compression gain (between low and high QP) can be up to 1.16 – 2.28-fold (when the obtained percentage values are directly compared). In the author's opinion, the essence of this effect is as presented in the previous sub-section. Additionally, the exact bitrate savings directly depend on the content of the test video sequence.

3) THIRD AND FOURTH VARIANTS OF VVC CABAC EXTENSION–VVC CABAC_0.5++ AND VVC CABAC++

The best results were obtained for the variants of the VVC CABAC extension in which optimal weighting factors w_1 and w_2 for the two probabilities (calculated with α_1 and α_2 values) were used. Detailed experimental data has been presented in the columns of Tables 1, 2 and 3 referring to VVC CABAC_0.5++ and VVC CABAC++. As seen from the Tables, the additional application of the optimal weighting factors w_1 and w_2 for the two probabilities in the VVC CABAC_0.5+ variant (achieving as a result the VVC CABAC_0.5++ variant) further increased the average gain of compression from 0.08241% to 0.12540% for SD video and 0.076% to 0.11478% for 2K video. In the case of resolutions beyond 2K it was further improved from 0.1224% to 0.1597%. When using the optimal weighting factors w_1 and w_2 in the VVC CABAC+ variant (achieving this way the VVC CABAC++ variant), the average compression gain rose from 0.14236% to 0.20731% for SD video sequences, from 0.12164% to 0.17839% for the 2K sequences and from 0.1522% to 0.1984% for resolutions beyond 2K. When looking at the detailed results obtained for individual test sequences, the reported additional increase of compression gain varied between 0.02759% and 0.07266% for SD video, 0.02% and 0.069% for 2K video, and between 0.025% and 0.047% for resolutions beyond 2K in the case of the VVC CABAC_0.5++ variant (relative to the VVC CABAC_0.5+). For the VVC CABAC++ variant, the corresponding values were 0.03797% to 0.10513% for SD video, 0.032% to 0.108% for 2K sequences and 0.036% to 0.058% for beyond 2K (relative to VVC CABAC+). Comparing the results to the performance of VVC CABAC, there was on average a 0.115% to 0.207% compression gain depending on the variant of the VVC CABAC extension (VVC CABAC_0.5++ or VVC CABAC++) and the class of video test sequences. The results discussed in this section show that it is profitable to use values of weighting factors w_1 and w_2 different than 0.5 for calculated two probabilities in the future improvements of the VVC CABAC technique.

V. CONCLUSION AND FINAL REMARKS

VVC CABAC is currently the most efficient entropy coding technique that is practically used in video compression. This technique has become a part of the new Versatile Video Coding technology – the technology that outperforms all other known technologies in terms of compression efficiency of video.

The superior compression efficiency of VVC CABAC is a result of applying improvements to the previous versions of CABAC that are used in the H.264/MPEG-4 AVC and H.265/MPEG-H HEVC video compression technologies. Those improvements concern mainly the mechanism of bin probability estimation, as highlighted in detail in Section 2 of the paper.

Research on further improving the VVC CABAC technique will certainly be continued. From the point of view of such works it is worth knowing, even indicatively, the possible limits of further improving the efficiency of the VVC CABAC. The goal of this paper was to try and find the answer to this question.

For this purpose the author has pointed out four possible directions of the VVC CABAC technique improvement.

When encoding a single bin, they come down to a more precise selection of values of α_1 and α_2 parameters (variants 1 and 2) as well as the use of dedicated weighting factors w_1 and w_2 for the two probabilities (variants 3 and 4). It is proposed that a more accurate estimation of the parameters be performed by taking into consideration the type of the encoded slice, as well as the content of the encoded video sequence. These elements are not taken into account by the current version of the VVC CABAC technique.

The specific goal of the paper was to show the possible limit values of the compression gain when using the considered approaches in the VVC CABAC technique. In connection with this, the author has implemented software which calculates the parameters in the best possible way, since it aims to minimize the value of entropy of the data that is then subject to entropy compression. The paper presents detailed results for the individual methods of VVC CABAC extension.

Based on the partial results, the following main conclusions can be drawn. The adjustment of some of the VVC CABAC parameters (i.e., values of α_1 and α_2) to the characteristics of the encoded sequence can lead to less than 0.122% gain in compression (average result for sequences with resolution beyond 2K). However, the additional dependence of the parameters on the type of the encoded slice increases the limit value of the profit to 0.1522%. The best results were obtained when, in addition to the optimal values of α_1 and α_2, also optimal values of weighting factors w_1 and w_2 of the two probabilities were used. In the latter case, the average increase in coding efficiency reached 0.1984%. For some encoding scenarios (specific test sequence and QP value) it was even 0.2% to 0.357%.

The results presented in this paper clearly show how strongly limited are the possibilities for further improving the efficiency of VVC CABAC (when using the methods presented in this paper). In order to better evaluate the presented results, it is worth referring the data to those presenting the compression gain of VVC CABAC relative to the older version of CABAC, used in H.265/MPEG-H HEVC (denoted as HEVC CABAC in the paper). As a matter of fact, in this case, the compression gains are higher, however, they refer to the older, less efficient version of CABAC (i.e., HEVC CABAC).

As the results of this paper show, in research for which the newest VVC CABAC is the anchor, obtaining a similar improvement is extremely difficult. The paper also shows that further way to improve the VVC CABAC technique, which is analogous to the improvement of HEVC CABAC to VVC CABAC, has already very strong limitations.

REFERENCES

[1] *Generic Coding of Moving Pictures and Associated Audio Information—Part 2: Video. (MPEG-2)*, document ISO/IEC 13818-2 and ITU-T Rec. H.262, Nov. 1994.

[2] 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.

[3] *Generic Coding of Audio-Visual Objects, Part10: Advanced Video Coding*, document ISO/IEC 14496-10, Mar. 2006.

[4] (2020). *AV1 Bitstream & Decoding Process Specification*. [Online]. Available: http://aomedia.org/av1/specification/

[5] M. Srinivasan, "VP9 encoder and decoders for next generation online video platforms and services," in *Proc. SMPTE Annu. Tech. Conf. Exhib.*, Oct. 2016, pp. 1–14, doi: 10.5594/M001734.

[6] 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 Technol.*, vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[7] *High Efficiency Video Coding (HEVC)*, document Rec. ITU-T H.265 and ISO/IEC 23008-2, Jan. 2013.

[8] D. Karwowski, "Significance of entropy coding in contemporary hybrid video encoders," in *Image Processing and Communications* (Advances in Intelligent Systems and Computing), vol. 184. Berlin, Germany: Springer-Verlag, 2013, pp. 111–117.

[9] 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.

[10] I. H. Witten, J. G. Cleary, and R. Neal, "Arithmetic coding for data compression," *Commun. ACM*, no. 6, pp. 520–540, Jun. 1987.

[11] A. Said, "Introduction to arithmetic coding-Theory and practice," in *Imaging Systems Laboratory*. Palo Alto, CA, USA: HP Laboratories, Apr. 2004.

[12] B. Bross, J. Chen, S. Liu, and Y.-K. Wang, *Versatile Video Coding (Draft 10)*, document JVET-S2001, Jul. 2020.

[13] A. Segall, V. Baroncini, J. Boyce, J. Chen, and T. Suzuki, *Joint Call for Proposal on Video Compression With Capability Beyond HEVC*, document ISO/IEC JTC 1/SC 29/WG, vol. 11, JVET-H1002, Macao, China, Oct. 2017.

[14] P. Haase, S. Matlage, H. Kirchhoffer, C. Bartnik, H. Schwarz, D. Marpe, and T. Wiegand, "State-based multi-parameter probability estimation for context-based adaptive binary arithmetic coding," in *Proc. Data Compress. Conf. (DCC)*, Mar. 2020, pp. 163–172, doi: 10.1109/DCC47342.2020.00024.

[15] V. Sze and M. Budagavi, "High throughput CABAC entropy coding in HEVC," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 22, no. 12, pp. 1778–1791, Dec. 2012.

[16] P. G. Howard and J. S. Vitter, "Practical implementations of arithmetic coding," in *Image and Text Compression Storer*. New York, NY, USA: Academic, 1992, pp. 85–112,

[17] A. Alshin, E. Alshina, and J. Park, "High precision probability estimation for CABAC," in *Proc. Vis. Commun. Image Process. (VCIP)*, Nov. 2013, pp. 1–6.

[18] A. Alshin, E. Alshina, 1. Park, *CEI (subset B): Multi-Parameter Probability Up-Date for CABAC*, document JCTVCG0764, Joint Collaborative Team Video Coding, Nov. 2011.

[19] J. Stegemann, H. Kirchhoffer, D. Marpe, and T. Wiegand, *Counter-Based Probability Model Update With Adapted Arithmetic Coding Engine*, document JCTVC-547, Joint Collaborative Team Video Coding (JCT-VC), Nov. 2011.

[20] D. Karwowski and M. Domanski, "Context-adaptive binary arithmetic coding with precise probability estimation and complexity scalability for high-efficiency video coding," *J. Electron. Imag.*, vol. 25, no. 1, Jan. 2016, Art. no. 013010, doi: 10.1117/1.JEI.25.1.013010.

[21] D. Karwowski and M. Doma?ski, "Increased compression efficiency of AVC and HEVC CABAC by precise statistics estimation," *Int. J. Electron. Telecommun.*, vol. 64, no. 3, pp. 277–284, 2018.

[22] D. Karwowski and M. Domański, "Improved context-adaptive arithmetic coding in H.264/AVC," in *Proc. 17th Eur. Signal Process. Conf. (EUSIPCO)*, Glasgow, Scotland, Aug. 2009, pp. 2216–2220.

[23] D. Karwowski and M. Domagski, "Improved arithmetic coding in H.264/AVC using context-tree weighting method," in *Proc. Picture Coding Symp. (PCS)*, Lisboa, Portugal, Nov. 2007, pp. 1–9.

[24] F. M. J. Willems, Y. M. Shtarkov, and T. J. Tjalkens, "The context-tree weighting method: Basic properties," *IEEE Trans. Inf. Theory*, vol. 41, no. 3, pp. 653–664, May 1995.

[25] F. M. J. Willems, "The context-tree weighting method: Extensions," *IEEE Trans. Inf. Theory*, vol. 44, no. 2, pp. 792–798, Mar. 1998.

[26] D. Karwowski, "Precise estimation of probabilities in CABAC using the cauchy optimization method," *IEEE Access*, vol. 8, pp. 32088–32099, 2020, doi: 10.1109/ACCESS.2020.2973549.

[27] H. Zhang, L. Song, X. Yang, and Z. Luo, "Evaluation of beyond HEVC entropy coding methods for DCT transform coefficients," in *Proc. Vis. Commun. Image Process. (VCIP)*, Chengdu, China, vol. 2016, pp. 1–4, doi: 10.1109/VCIP.2016.7805533.

[28] R. Song, D. Liu, H. Li, and F. Wu, "Neural network-based arithmetic coding of intra prediction modes in HEVC," in *Proc. IEEE Vis. Commun. Image Process. (VCIP)*, St Petersburg, Russia, Dec. 2017, pp. 1–4.

[29] C. Ma, D. Liu, X. Peng, and F. Wu, "Convolutional neural network-based arithmetic coding of DC coefficients for HEVC intra coding," in *Proc. 25th IEEE Int. Conf. Image Process. (ICIP)*, Athens, Greece, Oct. 2018, pp. 1772–1776, doi: 10.1109/ICIP.2018.8451166.

[30] C. Ma, D. Liu, X. Peng, Z. Zha, and F. Wu, "Neural network-based arithmetic coding for inter prediction information in HEVC," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, Sapporo, Japan, May 2019, pp. 1–5, doi: 10.1109/ISCAS.2019.8702499.

[31] J. Chen, Y. Ye, and S. H. Kim, *Algorithm Description for Versatile Video Coding and Test Model 7 (VTM 7)*, document JVET-P2002-v1, Geneva, China, Oct. 2019, pp. 1–11.

[32] *HEVC Test Model HM Software*. Accessed: Jan. 2020. [Online]. Available: https://hevc.hhi.fraunhofer.de/

[33] F. Bossen, J. Boyce, X. Li, V. Seregin, and K. Sähring, *JVET Common Test Conditions and Software Reference Configurations for SDR Video*, document JVET-N1010, Geneva, MA, USA, Mar. 2019, pp. 19–27

[34] G. G. Langdon and J. J. Rissanen, "Method and means for arithmetic coding utilizing a reduced number of operations," U.S. Patent 4 286 256, Aug. 18, 1981.

[35] G. G. Langdon and J. J. Rissanen, "Method and means for arithmetic string coding," U.S. Patent 4 122 440, Oct. 15, 1978.

[36] F. Rubin, "Arithmetic stream coding using fixed precision registers," *IEEE Trans. Inf. Theory*, vol. 25, no. 6, pp. 672–675, Nov. 1979.

[37] G. Langdon and J. Rissanen, "Compression of black-white images with arithmetic coding," *IEEE Trans. Commun.*, vol. COM-29, no. 6, pp. 858–867, Jun. 1981.

[38] G. G. Langdon, "An introduction to arithmetic coding," *IBM J. Res. Develop.*, vol. 28, no. 2, pp. 135–149, Mar. 1984.

DAMIAN KARWOWSKI received the M.Sc. and Ph.D. degrees from the Poznan University of Technology, in 2003 and 2008, respectively. He is currently an Assistant Professor with the Poznan University of Technology and a member of the Faculty of Computing and Telecommunications. He has been taking part in many industry-oriented projects that encompasses video and audio compression. He is an author and a coauthor of more than 50 papers on digital video compression in both national and international conferences and journal articles. His research interests include image, video and audio compression algorithms, and realization of video and audio codecs on PC and DSP platforms. He was the Technical Program Chair of the International Workshop on Signals, Systems and Image Processing, IWSSIP 2017, and the Picture Coding Symposium, PCS 2012, and a member of the Organizing Committee of International Workshop on Signals, Systems and Image Processing, IWSSIP 2004, the International Conference on Signals and Electronic Systems, ICSES 2004, and the 73rd Meeting of MPEG and European Signal Processing Conference, EUSIPCO 2007.