Improving the Efficiency of Entropy Coding Method in Video Encoder H.265/HEVC

Andrey Tropchenko, Doan Tien Ban and Nguyen Van Truong

ITMO University, Saint Petersburg, 197101, Russian Federation
zayka_98rus@mail.ru

Abstract. The High Efficiency Video Coding (H.265/HEVC) has better coding efficiency, but the encoding performance must be improved to meet the growing multimedia applications. This paper deals with the entropy encoding methods and algorithms in video encoder H.265/HEVC. Based on an analysis of their advantages and disadvantages, a method called the entropy coding algorithm using the enumerative coding of the hierarchical approach is proposed. The proposed algorithm consists of the Context-Adaptive Binary Arithmetic Coding (CABAC) algorithm and the enumerative coding algorithm with a hierarchical approach. The proposed algorithm is tested in the Visual C++ development environment on various test video sequences. The results of the experiments shows a greater efficiency of coding of multimedia data. Proposed method reduces up to 38.13% of the storage volume compared to the traditional CABAC method in HEVC. However, this method requires a longer coding time. The proposed method can be recommended for use in telecommunication systems for storage, transmission and processing of multimedia data, where a high compression ratio firstly required.

Keywords: H.265/HEVC, entropy coding, method CABAC, enumerative coding algorithm, video compression.

1 Introduction

Entropy coding is performed at the last stage of video encoding (and first stage of video decoding), after the video signal has been reduced to a series of syntax elements. The entropy encoder converts binary sequences representing elements of a video sequence into a compressed bit stream that can be stored in a file or transmitted over communication networks [1]. This encoder plays an important role in the whole coding process. The main methods of entropy coding are Huffman codes [2] and arithmetic coding [3]. On their basis, modern methods and algorithms for coding binary nonstationary sequences such as the CAVLC (Context-Adaptive Variable-Length Coding) [4,5] and CABAC [6,7] methods, which are widely used in the H. 264 [8] and H.265 [9].

An adaptive binary arithmetic coder with low memory consumption is proposed in [10], which provides comparable computational complexity by canceling the search tables requirement, but the decrease in the required memory is small (up to 2.3% according to the H.264 standard and up to 3.6% H.265). In [11], a new hardware-effective adaptive coder of the binary range and its architecture with very large-scale integration are proposed. The experimental results of the proposed encoder show that it significantly wins over existing encoders (up to 8% over the MQ encoder, and up to 24.2% by the M-encoder). And in [12], authors propose a context-adaptive binary arithmetic coding using code words with fixed length, which provides simplified computational complexity in comparison with the MEGC-encoder JPEG2000 and the HEVC M-encoder (in the order of about 2%).

HEVC uses CABAC as its single entropy coding method. HEVC CABAC was redesigned to offer higher throughput than its AVC predecessor, while still maintaining a higher compression ratio. This paper will focus on HEVC CABAC, and his aim is to propose a new method with higher efficiency compared to the traditional one.

2 The CABAC method

The CABAC method is based on arithmetic coding with several changes to adapt it to the needs of video encoding standards [13] and has the following features:

- It encodes binary symbols, which reduces the complexity of the method and allows the use of probabilistic models for the most commonly used bit sequences.
- Probabilistic models are chosen adaptively based on the local context, since the coding modes are locally well correlated.
- It uses undivided division of the range using quantized probability ranges and probabilistic states.

Fig. 1 shows a general block diagram of the coding of one syntax element in CABAC. The coding process consists of three elementary stages:

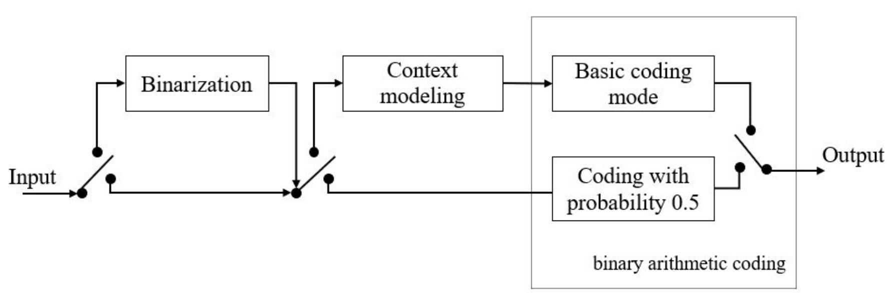

Fig. 1. General block diagram of CABAC coding

– Stage 1: binarization;
– Stage 2: context modeling;
– Stage 3: binary arithmetic coding.

At the first stage, this non-binary syntactic element is uniquely mapped into a binary sequence. When a binary element is given, this step is skipped. Depending on the encoding mode, one or two consecutive steps may follow for each element.

In the basic coding mode, the binary element enters the context modeling stage, where the probabilistic model is chosen so that the corresponding choice depends on the coded elements. Then, after assigning the model value, the binary element with its model goes to the basic coding mode, where the final stage of arithmetic coding with the subsequent modification of the model (for example, if the value was "1", the unit counter increases).

Alternatively, encoding with a probability of 0.5 is selected for the selected binary elements using a simplified coding mechanism (fig. 1). In H.264, an arithmetic coder is a set of procedures of low complexity in which there are no multiplication operations. Procedures include shifts and referrals to tables. Using CABAC allows you to reduce the bitrate by an average of 10-15%. The greatest gain is usually obtained when processing interlaced TV signals.

3 The proposed method

After analyzing the advantages and disadvantages of existing methods, a new one was proposed. The general scheme of the proposed algorithm is shown in fig. 2.

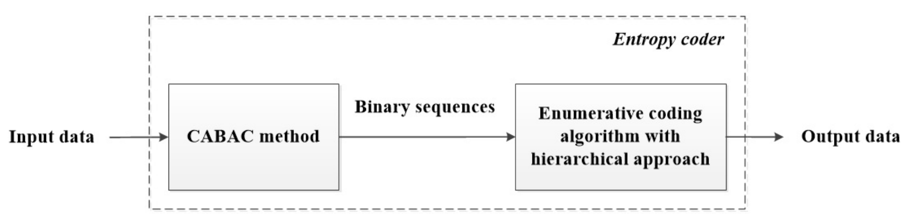

Fig. 2. General block diagram of coding by the proposed method

The input data of the entropy encoder is the parameters of the temporal model (motion vectors) and the spatial model (conversion coefficients), as well as markers (codes denoting synchronization points in the video sequence) and headers (macroblock, image, sequence and other object headers). After the execution of the CABAC algorithm, the input data is converted into binary sequences that are input to the enumerative coding algorithm with a hierarchical approach. After it, the final output of the entropy encoder is obtained, or, in other words, the compressed data of the video codec.

The enumerative coding algorithm with using hierarchies [14,15] includes the Lynch-Davisson method [16,17] and the enumerative coding method of bounded integers.

Let $\{0,1\}^n$ - the vector n of binary numbers and $x = (x_1, x_2, \ldots, x_n)$ - the elements of this vector, S - the set of vectors n of binary numbers. Define n_s - the number of elements in and denote $n_s(x_1, x_2, \ldots, x_n)$ the number of elements in S, the first k elements were defined. We suggest the vector w has elements 1 or $\sum_{j=1}^{n} x_j = w$.

Coding by enumerative coding method using hierarchical approach includes the following steps:

1. Construct a sum tree (fig. 3).

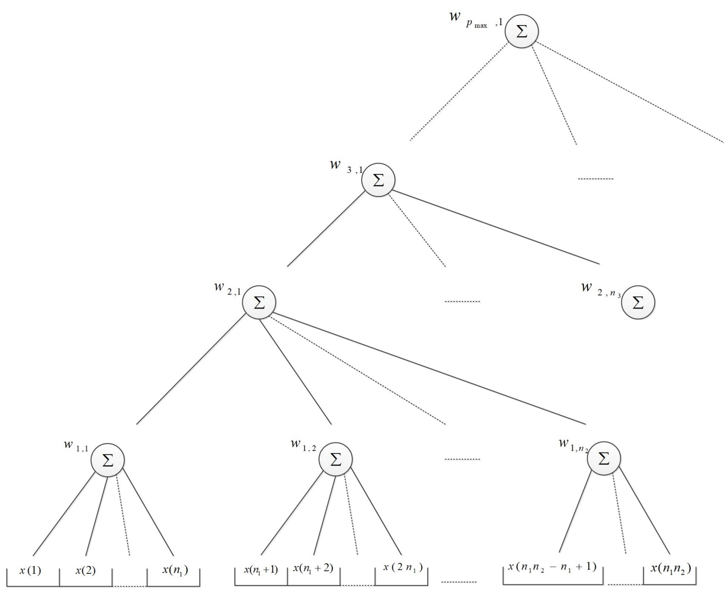

Fig. 3. Sum tree

2. Encode $w_{p_{max},1}$, using $[\log_2 (N+1)]$ bits.
3. Produce cycle for p from $p_{max} - 1$ down to 1 with a step 1:
   - Produce cycle for i from 1 up to $[\frac{N}{\prod_{j=1}^{p} n_j}]$,
   - Encode vector $\{w_{p,(i-1)n_p+1}, \ldots, w_{p,in_p}\}$ by enumerative encoding method of bounded integers.
4. Produce cycle for i from 1 up to $[\frac{N}{n_1}]$,
   - Encode $\{w_{p,(i-1)n_p+1}, \ldots, w_{p,in_p}\}$ by Lynch-Davison method to obtain an output sequence

The decoding process by enumerative coding method using hierarchical approach is performed in the same order as the encoding process.

3.1 The Lynch-Davisson method

The coding process consists in calculating the lexicographic index (the usual dictionary ordering in interpreting $0 < 1$) of the vector $x \in S \subseteq \{0, 1, 2, \ldots, M\}^n$ is determined by the following formula:

$$i_s(x) = \sum_{j=1}^{n} \sum_{m=0}^{x_j - 1} x_j n_s(x_1, x_2, \ldots, x_n) \tag{1}$$

To build the output code, $[\log_2 C_w^n]$ bits are required.

The decoding process is performed according to the following algorithm: if $i_s(x) > n_s(x_1, x_2, \ldots, x_n)$, $k = 1, \ldots, n$, then $x_k = 1$; otherwise $-$ $x_k = 0$. Run until the end of the sequence.

3.2 Enumerative coding method for bounded integers

Consider vector x of size n, each element x_i of vector satisfies the condition: $0 \le x_i \le M$, where M is a constant positive integer.

Here:

$$i_s(x) = \sum_{j=1}^{n-1} \sum_{m=0}^{x_j - 1} f_M((w - m - \sum_{j=1}^{k} x_i), n - j), \tag{2}$$

where:

$$f_M(p, q) = \sum_{i=p-M}^{p} f_M(i, q - 1),$$

$$f_M(p, 1) = \begin{cases} 1, \text{i}f : 0 \le p \le M \\ 0, otherwise. \end{cases}$$

To build the output code, $[\log_2 f_M(w, n)]$ bits are required.

When $M = 1$ this process is converted into Lynch-Davisson coding.

The decoding process is performed according to the algorithm presented in fig. 4.

4 Experimental results

The approach proposed above was implemented in the C++ development environment and tested with video sequences of different information character and different dimensions.

In the experiment we will compare the proposed algorithm with the algorithm CABAC, which are used in HM 16.0 (H.265/HEVC Test Model) with configuration "intra main profile" by the following criteria: the quality of the encoding (by the volume of the output stream) and the encoding time (here, QP is the quantization parameter). The experiment is carried on Window 10 OS platform with Intel(R) Core(TM) i5-4210U CPU @1.70GHz 2.40 GHz and 6GB RAM.

The quality of the encoding is estimated by comparing the volume of the output stream (shown in Table 1). The experimental results showed that the proposed algorithm reduces the output file up to 38.13%.

Table 1. The volume of the output stream of different video sequences

Type	Video sequences	The percentage reduction in the volume of the proposed method over CABAC H.265 / HEVC, %				
		QP = 22	QP = 27	QP = 32	QP = 37	QP = 42
QCIF	Bus_QCIF	13.82	36.69	36.99	35.26	39.79
	Football_QCIF	36.16	36.36	35.98	36.69	38.23
	Foreman_QCIF	26.39	37.12	36.56	39.26	39.12
	Ice_QCIF	16.19	36.05	37.38	35.96	40.22
	Mobile_QCIF	36.24	36.33	37.01	36.71	37.42
CIF	City_CIF	36.01	36.17	36.23	15.98	37.84
	Football_CIF	36.01	36.11	36.18	36.29	36.81
	Foreman_CIF	36.26	36.19	37.02	36.23	37.24
	Ice_CIF	35.11	36.21	36.75	36.19	38.16
	Soccer_CIF	35.84	36.14	35.43	36.81	39.20
D1	Crew_D1	36.13	35.87	36.53	36.54	37.26
	Ice_D1	25.56	36.26	36.23	36.76	37.06
	Harbour_D1	36.00	35.64	35.92	36.10	36.41
Full HD	Beauty_HD	15.23	36.17	36.15	36.44	37.54
	Bosphorus_HD	35.96	36.03	36.23	36.25	38.26
	ReadySetGo_HD	34.10	35.94	36.54	35.98	37.69
4K	Jockey_4K	38.39	29.24	36.36	36.05	38.59
	YachtRide_4K	32.89	35.98	35.94	35.98	39.45
Average		31.24	35.81	36.41	35.30	38.13

The coding time is estimated by comparing the average time required to encode each video stream shown in Table 2. It can be seen that the proposed algorithm increases the coding time by an average of 15.4% compared to the CABAC H.265 / HEVC method.

Thus, it can be argued that the proposed method provides a greater compression ratio (up to 38.13%), while it requires more time to work compared to other algorithms.

Table 2. Average time for encoding different video sequences

Type	Video sequences	The percentage increase in the volume of the proposed method over CABAC H.265 / HEVC, %				
		QP = 22	QP = 27	QP = 32	QP = 37	QP = 42
QCIF	Bus_QCIF	23.33	21.91	27.51	14.56	17.21
	Football_QCIF	8.78	28.43	16.53	10.63	16.57
	Foreman_QCIF	6.96	32.15	21.03	13.94	15.24
	Ice_QCIF	24.61	12.30	34.54	17.23	14.84
	Mobile_QCIF	26.88	29.41	12.32	23.99	22.07
CIF	City_CIF	12.32	18.76	15.25	21.48	6.48
	Football_CIF	23.46	13.38	38.11	23.37	12.48
	Foreman_CIF	39.03	21.22	24.13	14.36	7.89
	Ice_CIF	39.42	13.02	14.34	13.20	5.99
	Soccer_CIF	5.87	37.99	12.36	15.09	5.03
D1	Crew_D1	30.40	14.65	11.36	6.70	4.68
	Ice_D1	27.25	14.88	8.48	6.32	3.44
	Harbour_D1	34.33	14.52	8.63	16.42	4.56
Full HD	Beauty_HD	42.04	7.96	4.09	2.71	1.02
	Bosphorus_HD	27.85	10.86	9.06	4.46	1.98
	ReadySetGo_HD	17.70	20.85	9.12	3.12	2.63
4K	Jockey_4K	10.63	5.83	6.98	2.408	4.56
	YachtRide_4K	13.84	8.80	7.76	2.929	2.53
Average		23.04	18.16	15.64	11.83	8.29

5 Conclusion

In this paper, we propose to apply the entropy encoding algorithm to improve the efficiency of multimedia data compression. The experimental results show that proposed method gives a good compression ratio (the output file size is reduced up to 38.13%), but the encoding procedure time is increased (approximately 15.4%) compared to the traditional CABAC method in the HEVC standard.

References

1. *Zhang, H. et al.* Evaluation of beyond-HEVC entropy coding methods for DCT transform coefficients // 2016 Visual Communications and Image Processing (VCIP). IEEE, 2016. P. 1–4.
2. *Huffman, D.Á.* A method for the construction of minimum-redundancy codes // Proc. IRE. 1952. Vol. 40, No. 9. P. 1098–1101.
3. *Rissanen, J., Langdon, G.Ġ.* Arithmetic Coding // IBM Journal of Research and Development. 1979. Vol. 23, No. 2. P. 149–162.
4. *Acharyya, A., Kothari, S., Reeve, J.* A new CAVLC algorithm for higher bit compression by introducing the concept of Position Coding of the coefficients in H.264/AVC // 2012 IEEE International Conference on Industrial Technology. IEEE, 2012. P. 123–128.

5. *Nargundmath, S., Nandibewoor, A.* Entropy coding of H.264/AVC using Exp-Golomb coding and CAVLC coding // International Conference on Advanced Nanomaterials & Emerging Engineering Technologies. IEEE, 2013. P. 607–612.
6. *Rapaka, K., Yang En-Hui.* A High Throughput Multi Symbol CABAC Framework for Hybrid Video Codecs // 2013 Data Compression Conference. IEEE, 2013. P. 515–515.
7. *Marpe, D., Schwarz, H., Wiegand, T.* Context-based adaptive binary arithmetic coding in the H.264/AVC video compression standard // IEEE Trans. Circuits Syst. Video Technol. 2003. Vol. 13, No. 7. P. 620–636.
8. *Wiegand, T.* Overview of the H. 264/AVC video coding standard // IEEE Transactions on Circuits and Systems for Video Technology. 2003. Vol. 13, No. 7. P. 560–576.
9. *Sullivan, G.J̇. et al.* Overview of the high efficiency video coding (HEVC) standard // IEEE Transactions on Circuits and Systems for Video Technology. 2012. Vol. 22, No. 12. P. 1649–1668.
10. *Belyaev, E., Turlikov, A., Egiazarian, K.* An Efficient Adaptive Binary Arithmetic Coder With Low Memory Requirement // IEEE J. Sel. Top. Signal Process. Spec. Issue Video Coding HEVC beyond, 2013. IEEE, 2013. Vol. 7, No. 6. P. 1053–1061.
11. *Belyaev, E. et al.* An Efficient Adaptive Binary Range Coder and Its VLSI Architecture // IEEE Trans. Circuits Syst. Video Technol. 2015. Vol. 25, No. 8. P. 1435–1446.
12. *Aulí-Llinàs, F.* Context-Adaptive Binary Arithmetic Coding With Fixed-Length Codewords // IEEE Trans. Multimed. 2015. Vol. 17, No. 8. P. 1385–1390.
13. *Neji, N. et al.* FPGA implementation of improved binarizer design for context-based adaptive binary arithmetic coder // 2016 International Image Processing, Applications and Systems (IPAS). IEEE, 2016. P. 1–4.
14. *Nguyen, V.Ṫ., Tropchenko, A.Ȧ.* Improving the efficiency of data compression using a hierarchically enumerative coding // Izv. vuzov. Priborostroenie. 2016. Vol. 59, No. 12. P. 991—996 (in Russian).
15. *Nguyen, V.Ṫ.* Analysis and development of entropy coding methods and algorithms. Saarbrucken - Berlin - Leipzig, Deutchland: LAP LAMBERT Academic Publishing, 2017. 52 p. (in Russian).
16. *Davisson, L.Ḋ.* Comments on “Sequence Time Coding for Data Compression” // Proceedings of the IEEE. 1966. Vol. 54, No. 12. P. 2010–2011.
17. *Lynch, T.J̇.* Sequence time coding for data compression // Proc. IEEE. 1966. Vol. 54, No. 10. P. 1490–1491.

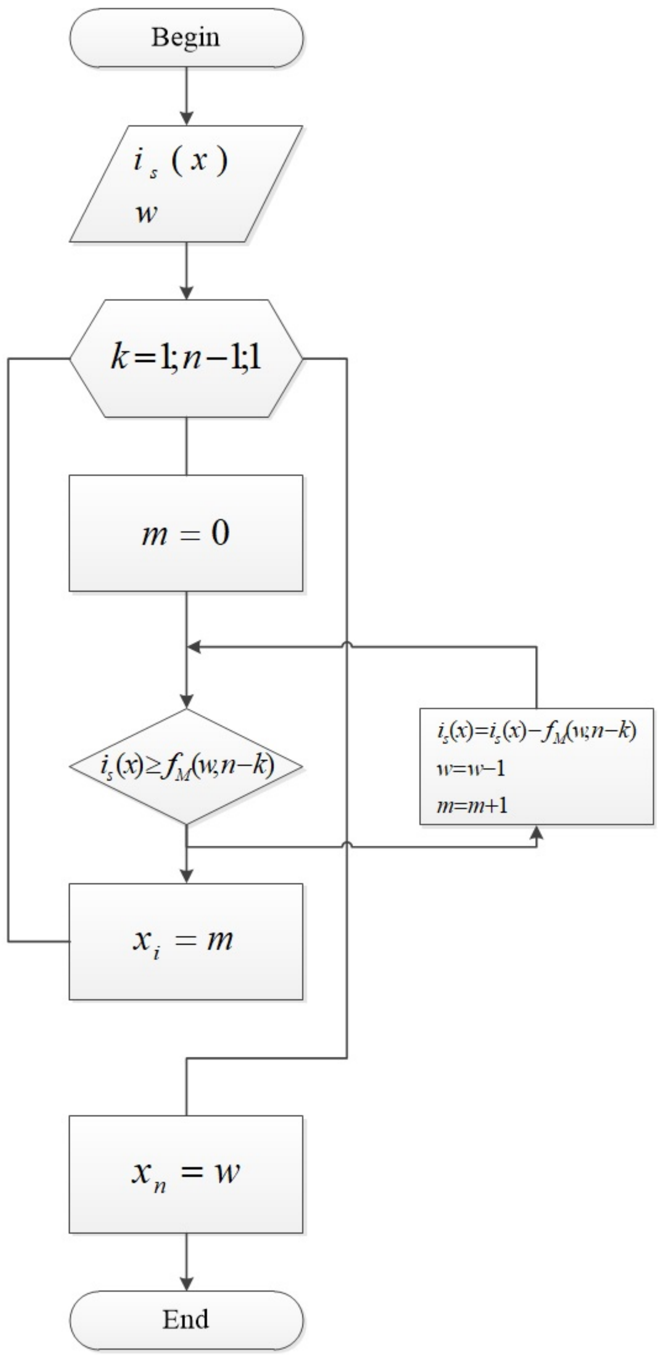

Fig. 4. Decoding process by numbering encoding of bounded integers