ISSN(Online): 2320-9801
ISSN (Print): 2320-9798

International Journal of Innovative Research in Computer and Communication Engineering

(An ISO 3297: 2007 Certified Organization)

Vol. 2, Issue 8, August 2014

An Efficient Adaptive Binary Arithmetic Coder and Its Application in Video Coding

R N M S Sindhu, G Rama Krishna

Postgraduate Student, Department of ECE, SVCET (Autonomous), Chittoor, A.P, India.

Professor, Department of ECE, SVCET (Autonomous), Chittoor, A.P, India.

ABSTRACT: Adaptive Binary Arithmetic coding is an effective mode of coding. In this paper we propose an efficient adaptive binary arithmetic coder, which is multiplication-free and look-up table free. To achieve this, we combine the probability estimation based on the virtual sliding window technique with the approximation of multiplication and the use of simple operations to calculate the next approximation after the encoding of each binary symbol. We show that the proposed algorithm is faster and provides the better compression efficiency compared to the M-coder in the CABAC entropy coding scheme of the H.264/AVC video coding standard.

KEYWORDS: Binary arithmetic coding, M-coder, H.264/AVC.

I.INTRODUCTION

Arithmetic coding has attracted a growing attention in the past few years. It is an effective mode of coding and it is one of the very important algorithms in all image and video compression standards such as JPEG [1], JPEG-2000 [2] and H.264/AVC [3], HEVC [4], Dirac [5]. The algorithm is not only a lossless data compression method but also a type of entropy encoding. The arithmetic coder used in these standards which is multiplication free adaptive binary arithmetic coder with bit-renormalization and look-up tables used for multiplication approximation and probability estimation.

A basic idea about arithmetic coding uses a single floating point instead of string of input symbols so that it can be obtain a more efficient compression. The method is representing the encoded information as an interval between real axis of 0~.The interval is smaller than longer information, and indicating a longer information needs more number binary.

As a common requirement for all these applications, there is need for a fast and efficient binary arithmetic coding scheme. Efficient implementations in arithmetic coding both in hardware and software, however are not only required in the application domains of image and video coding but also in the data compression scheme needs high throughputs of arithmetic codes.

The most efficient ABAC is the M-coder [7], which is the core of the context adaptive binary arithmetic coding (CABAC) [9] used in H.264/AVC. Since, M-coder is a key component of the entropy encoding in both compression schemes, the implementation of the ABAC with smaller memory foot print and lower computational complexity and higher compression efficiency remains an important challenge.

II.RELATED WORK

Arithmetic coding is a form of Entropy used in Lossless Data Compression. Normally, a string of characters such as the words "hello there" is represented using a fixed number of bits per character, as in the code. When a string is converted to arithmetic encoding, frequently used characters will be stored with fewer bits and not-so-frequently occurring characters will be stored with more bits, resulting in fewer bits used in total. Arithmetic coding differs from other forms of entropy encoding such as in that rather than separating the input into component symbols and replacing each with a code, arithmetic coding encodes the entire message into a single number, a fraction *n* where ($0.0 \leq n < 1.0$).

What most compression systems have in common is the fact that the final process is entropy encoding, which is the process of representing information in the most compact form. It may be responsible for doing most of the compression work or it may just complement what has been accomplished by previous stages. When we consider all the different entropy coding methods, and their possible applications in compression applications, arithmetic coding stands out in terms of elegance, effectiveness and versatility, since it is able to work most efficiently in the largest number of circumstances and purposes.

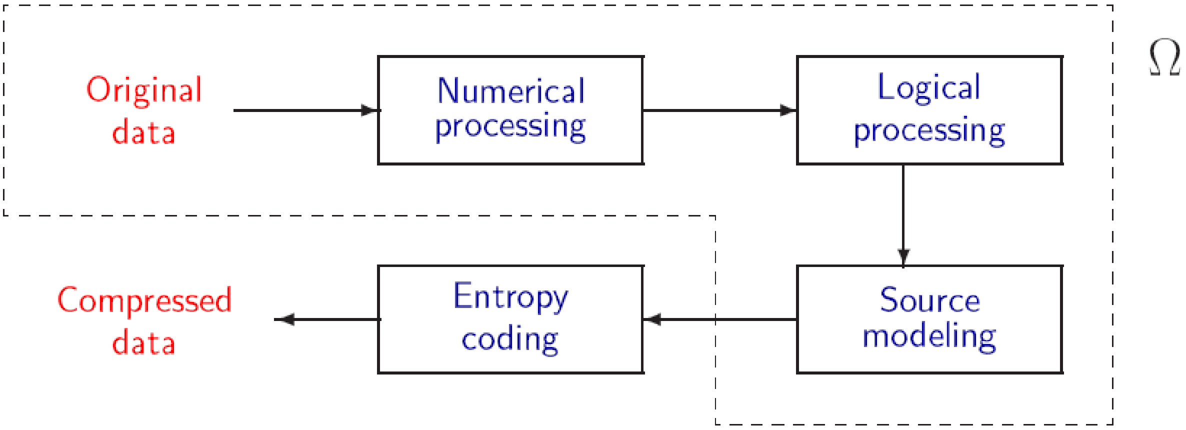


Figure 1: System with typical processes for data compression. Arithmetic coding is normally the final stage, and the other stages can be modeled as a single data source Ω.

Arithmetic coding is different from other coding methods for which we know the exact relationship between the coded symbols and the actual bits that are written to a file. It codes one data symbol at a time, and assigns to each symbol a real valued number of bits.

The code value *v* of a compressed data sequence is the real number with fractional digits equal to the sequence's symbols. We can convert sequences to code values by simply adding “0." to the beginning of a coded sequence, and then interpreting the result as a number in base-*D* notation, where *D* is the number of symbols in the coded sequence alphabet. For example, if a coding method generates the sequence of bits 0011000101100, then we have

Code sequence d = [0011000101100]

Code value $v = 0{:}\ 0011000101100_2 = 0.19287109375$

III.THE MAIN CONTRIBUTION OF THE PAPER

In this paper, we present an efficient adaptive binary arithmetic coder which does not require multiplications and look-up tables. The proposed algorithm is a modification of ABAC based on virtual sliding window (VSW). It also allows assigning a specific window length depending on the statistical properties of the corresponding binary source. Therefore, in comparison to the state-of-the-art M-coder, it is faster and requires no look-up tables. Furthermore, it improves the compression efficiency.

IV.PROPOSED METHOD

A. Multiplication-free implementation:

Probability estimation can be implemented by state machine approach. Some probability value is corresponds to each state. The transition state to state is defined by the value of the input symbol. This approach does not require multiplications or divisions for probability calculations. In addition to this the fixed set of states allows to implement the interval division part of the arithmetic coder without multiplications [6].

For example let us consider a state machine based probability estimation in state-of-the-art M-coder from H.264/AVC standard. In M-coder [7] input symbols are divided in two types: Most probable symbols (MPS) and least probable symbols (LPS). State machine contains 64 states. East state defines probability estimation for least probable symbol. Set of probability values { ̂p0,p̂1,p̂2......p̂63 }.

To remove the multiplication, M-coder uses its approximation. After the renormalization procedure, register R satisfies the following inequality:

$$\frac{1}{2}2^{b-1} \leq R < 2^{b-1} \qquad (1)$$

From (1) it follows that a multiplication can be approximated in the following way:

$$T = R \times \hat{p}_t \approx \alpha 2^{b-1} \times \hat{p}_t \qquad (2)$$

B. Look-up table implementation:

A probability estimation using virtual sliding window is proposed [10]

$$s'_{t+1} = \begin{cases} s'_t + \left\lfloor \dfrac{\alpha 2^{b-1} 2^{\omega} - s'_t + 2^{\omega-1}}{2^{\omega}} \right\rfloor, & if\ x_t = 1 \\ s'_t - \left\lfloor \dfrac{s'_t + 2^{\omega-1}}{2^{\omega}} \right\rfloor, & if\ x_t = 0 \end{cases} \qquad (3)$$

In comparison with the M-coder, provides better compression efficiency due to an assignment of a specific virtual sliding window length selected according to the statistical properties of the corresponding binary source [10] and using multiplication in interval division part of the arithmetic coder instead of its approximation. Nevertheless, using these multiplications is not efficient; especially for a hardware implementation.

The proposed method gives the solution uses look-up table-free approach based on virtual sliding window (VSW). As an alternative to arithmetic coders, range coders use bytes as output bit stream element and do byte renormalization at a time. It was shown that adaptive binary range coder allows decreasing computation complexity. The new CABAC architecture was proposed. This architecture reduces the memory requirement by 50%, with at the expense of up to 0.02% coding loss.

THE PROPOSED ADAPTIVE BINARY ARITHMETIC CODER

To eliminate the multiplication operation, we use a similar reasoning as the one described in the above. Let us multiply both sides of above equation by $\alpha 2^{b-1}$:

$$s'_{t+1} = \left(1 - \frac{1}{W}\right).s'_t + \alpha\, 2^{b-1} x_t \qquad (4)$$

Where $s'_t = \alpha 2^{b-1} s_t$ after integer rounding of (4) we obtain

$$s'_{t+1} = \begin{cases} s'_t + \left\lfloor \frac{\alpha 2^{b-1} 2^{\omega} - s'_t + 2^{\omega-1}}{2^{\omega}} \right\rfloor, & if\ x_t = 1 \\ s'_t - \left\lfloor \frac{s'_t + 2^{\omega-1}}{2^{\omega}} \right\rfloor, & if\ x_t = 0 \end{cases} \qquad (5)$$

Taking into account of equations (1), (2)

$$T = R \times \hat{p}_t \approx \frac{s_t' + \Delta \times \frac{1}{4} s_t'}{2^{\omega}}, \qquad where\ \Delta = \frac{R - 2^{b-2}}{2^{b-4}} \qquad (6)$$

The proposed coder can be implemented by using conditional and addition operations

$$s_t' > a2^{b-2}2^{\omega} \qquad (7)$$

By seeing the above equations (6), (7) an adaptive binary arithmetic coding is proposed by using conditional and addition operations.

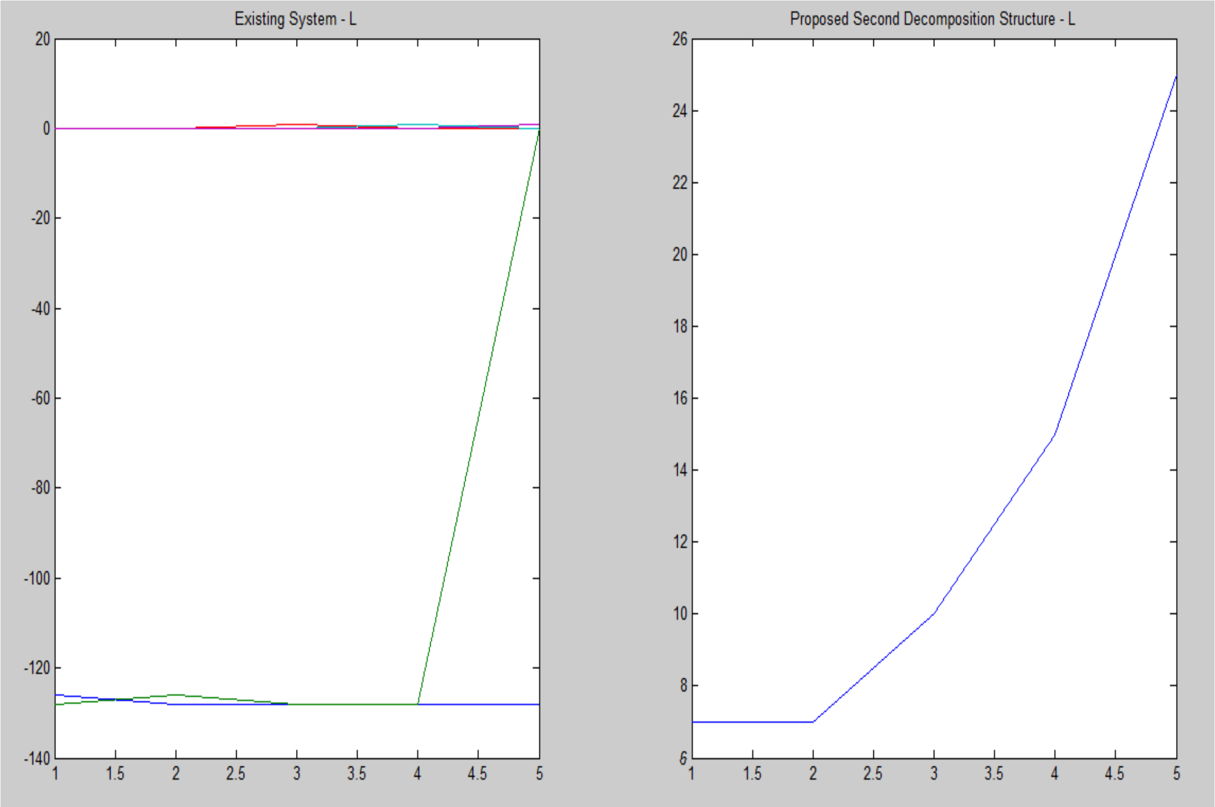
V. SIMULATION RESULTS

For practical experiments the proposed adaptive binary arithmetic coder was embedded into the JM codec which is the reference software of H.264/AVC video coding standard. Compression results were obtained for the test video sequence “Pedestrian Area “for the Main profile of the standard. Bit-rate savings in percentage related to the M-coder for different quantization parameters are presented in Table 1. For both coders L and R are 10-bits registers (b = 10).

Table1. Bit-rate savings (in %) related to the M-coder for “Pedestrian Area“(1920×1080, 60 frames)

QP	0	10	20	30	40	50
W=2^5	0.80	0.46	0.40	0.40	0.39	0.28
W=2^6	1.12	0.67	0.58	0.47	0.21	-1.04

Figure2. Shows the encoding time for 10^8 input binary symbols of binary memory less sources with fixed ones probability.



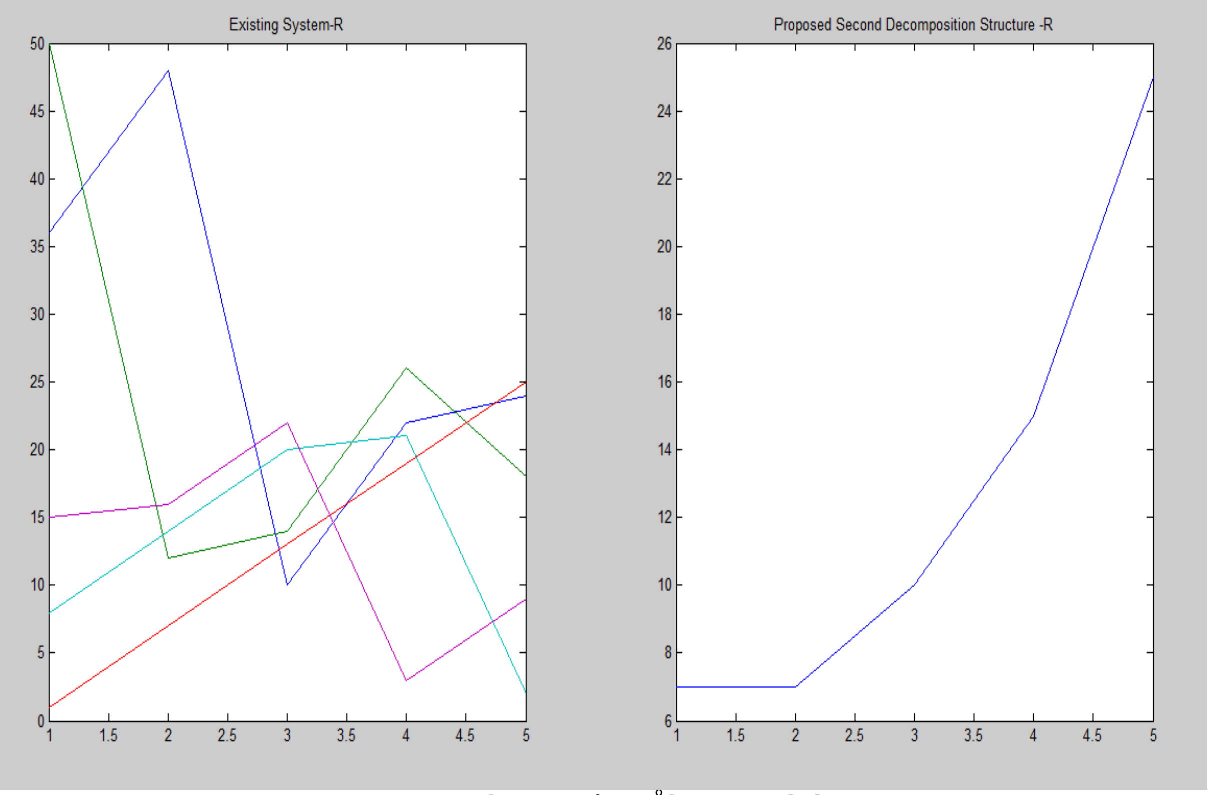


Fig.3 encoding time for 10^8 binary symbols

Our results show that the proposed adaptive binary arithmetic coder allows improving compression ratio of the H.264/AVC video coding standard for fixed visual quality. At the same time it has less encoding time than the M-coder.

VI.CONCLUSION

A new fast and efficient algorithm of binary arithmetic coding was presented. In comparison to the M-coder, it is faster, achieves better compression efficiency and less computational complexity and does not require look-up tables. Taking into account of these considerations the proposed coder is a modification of ABAC based on VSW. This coder can also achieve the trade-off between adaptation speed and precision of the probability estimation due to the utilization of different window sizes w. Experimental results have shown that our proposed coder provides advantages both in terms of compression performance and speed, when compared to the state-of-the-art M-coder. Therefore, the proposed ABAC can be more preferable for a future image and video coding standards and non-standardized codec's.

REFERENCES

[1] E. Belyaev, M. Gilmutdinov, and A. Turlikov, “Binary arithmetic coding system with adaptive probability estimation by Virtual

[2]D.arpe, 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.

[3] V. Sze and M. Budagavi, “Overview of the high efficiency video coding (HEVC) standard,” IEEE Trans. Circuits Syst. Video Technol., vol. 22, no. 12, pp. 1649–1668, Dec. 2012.

[4] T. Nguyen, H. Schwarz, H. Kirchhoffer, D. Marpe, and T. Wiegand, “Improved context modeling for coding quantized transform coeffi- cients in video compression,” in Proc. Picture Coding Symp., 2010.

[5] H. Eeckhaut, B. Schrauwen, M. Christiaens, and J. Van Campenhout, “Tuning the M-coder to improve dirac's entropy coding,” WSEAS Trans. Inf. Sci. Applicat., pp. 1563–1571, 2005.

[6] W.Pennebaker, J.Mitchel,G.Langdon,and R.Arps,“Anoverview of the basic principles of the q-coder adaptive binary arithmetic coder,” IBM J. Res. Develop., vol. 32, pp. 717–726, 1988.

[7] D. Marpe and T. Wiegand, “A highly efficient multiplication-free bi- nary arithmetic coder and its application in video coding,” in Proc. IEEE Int. Conf. Image Process., 2003, pp. 263–266.
[8] High Efficiency Video Coding, [Online]. Available: http://www.h265. net/
[9] V.SzeandM.Budagavi,“HighthroughputCABACentropycodingin HEVC,” IEEE Trans.Circuits Syst.Video Technol.,vol.22,no.12,pp. 1778–1791, Dec. 2012.Sliding Window,” in Proc. 10th IEEE Int. Symp. Consumer Electron., 2006, pp. 194–198.
[10] D. Karwowski and M. Domanski, “Improved context-adaptive arithmetic coding in H.264/AVC,”in Proc.17th Eur.Signal Process.Conf., 2009.
[11]D.arpe, 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.
[12] D. Karwowski and M. Domanski, “Improved context-adaptive arithmetic coding in H.264/AVC,”in Proc.17th Eur.Signal Process.Conf., 2009.
[13] T. Nguyen, H. Schwarz, H. Kirchhoffer, D. Marpe, and T. Wiegand, “Improved context modeling for coding quantized transform coeffi- cients in video compression,” in Proc. Picture Coding Symp., 2010.
[14] K.Vermeirsch,J.Barbarien,P.Lambert,andR.VandeWalle,“Region- adaptive probabilitymodel selection forthe arithmetic coding ofvideo texture,” in Proc. IEEE Int. Conf. Acoust., Speech, Signal Process., 2011, pp. 1537–1540.
[15] D. Hong and A. Eleftheriadis, “Memory-efficient semi-quasi renor- malization for arithmetic coding,” IEEE Trans. Circuits Syst. Video Technol., vol. 17, no. 1, pp. 106–109, Jan. 2007.
[16] M. Schindler, “Byte oriented arithmetic coding,” in Proc. Data Com- pression Conf., 1998.
[17] D.Subbotin, Carryless Rangecoder, 1999, [Online].Available:http://search.cpan.org/src/SALVA/Compress-PPMd-0.10/Coder.hpp.
[18] P. Lindstrom and M. Isenburg, “Fast and efficient compression of floating-point data,” IEEE Trans. Vis. Comput. Graphica, vol. 12, no. 5, pp. 1245–1250, Sep.-Oct. 2006.
[19] A. Said, “Comparative analysis of arithmetic coding computational complexity,” Hewlett-Packard Lab. Rep., HPL-2004-75, 2004.
[20] K. Sarawadekar and S. Banerjee, “An efficient pass-parallel architec- ture for embedded block coder in JPEG 2000,” IEEE Trans. Circuits Syst. Video Technol., vol. 21, no. 6, pp. 825–836, Jun. 2011.
[21] “JPEG 2000 Image Coding System: Core Coding System,” ITU-T and ISO/IEC JTC 1, ITU-T Rec. T.800 and ISO/IEC 15444-1, JPEG 2000 Part 1, 2000.