Joint coding Scheme for Maximum Macroblock Prediction Type and Coding Block Pattern

Honggang Qi, Wen Gao, Siwei Ma and Debin Zhao

Institute of Computing Technology
Chinese Academy of Sciences, Bejing, 100080, China

Abstract. In this paper, a joint coding scheme for improving the coding efficiency of macroblock prediction type (MBTYPE) and coded block pattern (CBP) is proposed. The proposed coding scheme exploits the characteristics of zero-order Exp-Golomb code for coding MBTYPE and CBP, and makes use of the fractional bits which are unused by MBTYPE to code both maximum MBTYPE and CBP so that each bit can be more efficiently used. With joint coding scheme, the average length of coded bits for the maximum MBTYPE and the CBP is shortened. The total bitrate is therefore reduced. The analysis and experimental results show that the proposed scheme is efficient for improving the coding performances of sequences, especially to those sequences with more blocks coded with maximum MBTYPE.

***Index Terms*—** CBP, MBTYPE, joint coding, Exp-Golomb code

1. INTRODUCTION

With the development of video compression technology, the digital video represented by very large data can be efficiently compressed into smaller bitstream for practice applications. In recent decades, many video coding standards are coming forth, such as MPEG2 [1], H.264/AVC [2] and so on. They all serve the video coding very well. Presently, High-definition (HD) TVs are becoming the mainstream of TV market, as its high resolution and frame frequency provide higher visual quality of video to audiences. However, the higher visual quality is achieved at the cost of more data. The larger data of HD video require power storage and high transmission capabilities. Besides, longer encoding and decoding time of a HD picture prevents them from many real-time applications, such as HD broadcasting. For solving these problems, many video coding standards specially orienting to HD video coding are developed in the world. In China, a new coding standard for HD video, named China Audio and Video coding Standard (AVS), has been developed and made a great progress. Similarly to H.264/AVC, AVS is based on the framework of hybrid coding, too. Many new techniques are adopted in the standard, such as new bi-direction prediction, new direct mode, low-complexity 8x8 transform and quantization and so on. The new standard reduces the coding complexity greatly while maintaining the high compression performance to HD video.

In the coding standards based on block prediction, a macroblock is partitioned into different block sizes hierarchically for motion estimation/compensation. One of the block size modes with least coding cost can be selected as the optimal coding mode for the macroblock. The syntax macroblock prediction type (MBTYPE) is used to identify the selected coding mode for coding a macroblock. The syntax coded block pattern (CBP) is a bit sequence where each bit is used to indicate whether a corresponding quantized transform coefficient block contains non-zero coefficients or not. MBTYPE and CBP are widely employed in many coding standards. In the bitstreams complying with video coding standards, the proportion of coded bits for MBTYPE and CBP to the total coded bits may be quite few. Thus, the improvement of their coding efficiency is not cared. However, the coding performance of them still has an effect on the whole coding performance, especially at low bitrate. Thus, it is worth improving the coding performance of MBTYPE and CBP with a very small increasing complexity. In this paper, considering the characteristic of the zero-order Columbus code, the MBTYPE and CBP are jointly coded for reducing the coded bits. The joint encoding scheme aims at exploiting each bit fully so that carries as much information as possible. As a result, the same information can be represented with fewer bits.

The remainder of this paper is organized as follows, In the next section, variable length coding applied to video coding standards is first introduced. Then, the joint coding scheme of maximum MBTYPE and CBP is described in the section 3. The experimental results on HD test sequences are shown in the section 4. Finally, the conclusions are made.

2. VARIABLE LENGTH CODING

Entropy coding is a lossless compression tool. The original data compressed with entropy coding can be completely reconstructed without any loss. Entropy coding contains two parts, modeling and coding. Modeling assigns probabilities to the symbols, and coding produces a bit sequence from these probabilities. Variable length coding (VLC), one of commonly used entropy coding scheme, is often used in symbol coding. The Huffman coding is a typical VLC, which mapping each symbol into a bit sequence, and named code word. The frequent symbols are coded with short code words. Contrarily, the non-frequent symbols are coded with long code words. The appearance frequencies of symbols are measured with their statistic probabilities. Symbols are mapped into code word with different lengths according to their statistic probabilities. Average code word length of Huffman coding is approximate to entropy. However, the real Huffman coding is not usually used in standards due to its complexity that the code table often needs to be updated with the altered statistic probabilities. Thus, universal variable length coding (UVLC), a simpler entropy coding tool derived from Huffman coding, is usually used in standards. In UVLC, symbols are mapped into the code words through looking up a UVLC table which are not updated when coding.

The all code words in UVLC are created following the rule of Exp-Golomb coding. Golomb coding is a form of entropy coding invented by Solomon W. Golomb [3]. It is optimal for alphabets with geometric distribution. Rice coding is a special case of Golomb coding where the tunable parameter is a power of two [4]. It can be efficiently computed on computers. The k-order Exp-Golomb code words are shown in the Table 1, where the bit sequence of xn… x1, x0 in code words is defined as INFO where the value of xn is 0 or 1. The code numbers are the decimal values of INFOs.

If the bit length of INFO is L which is equal to 2n-1, the code numbers can be represented by the following formula with INFO and L,

$$codenumber = 2^{L/2} + INFO - 1 \qquad (1)$$

If the value of INFO and its length L are known, the regular structure of the table makes encoder easy to create a code word. Similarly, the decoder can easily parse a code word by reading n-bit prefix followed by n-1-bit INFO. And then, the code number is obtained according to formula (1). Analyzing the characteristics of Exp-Golomb code, the proposed scheme codes the maximum MBTYPE and CBP jointly in order to save the coding bits for them.

Table 1. The K-order Exp-Golomb code table.

Order	Code word structure	Code number
k = 0	1	0
	0 1 x0	1-2
	0 0 1 x1 x0	3-6
	0 0 0 1 x2 x1 x0	7-14
	......	......
k = 1	1 x0	0-1
	0 1 x1 x0	2-5
	0 0 1 x2 x1 x0	6-13
	0 0 0 1 x3 x2 x1 x0	14-29
	......	......
k = 2	1 x1 x0	0-3
	0 1 x2 x1 x0	4-11
	0 0 1 x3 x2 x1 x0	12-27
	0 0 0 1 x4 x3 x2 x1 x0	28-59
	......	......

3. JOINT ENCODING FOR MAXIMUM MBTYPE AND CBP

In AVS and some other coding standards, syntaxes MBTYPE and CBP are independently coded into bitstream with zero-order Exp-Golomb code. The number of MBTYPE *m* may not just be a value of the power of two, i.e. $2n–1 < m < 2n$. In this case, the *n* bits have to be cost for the binary code of each value of MBTYPE, which leads to the $n–log2m$ bits be wasted. Taking AVS as an example, the number of MBTYPE is 5, thus 3 bits are used to code each value of MBTYPE. As a result, the $(3–\log_2 5)=0.678$ bits are wasted for each MBTYPE coding. To the zero-order Exp-Golomb code used for coding MBTYPE, the situation of calculating bit cost is a little different from that above. However, their principles are completely the same. Taking AVS as an example again, the number of MBTYPE 5 is represented in zero-order Exp-Golomb code of 00110, thus 5 bits are used to code the code number 3, 4 and 5. However, the code number 6 will never be used in the MBTYPE coding. It will result in $(\log_2 4–\log_2 3)=0.415$ bits to be wasted when the code number of MBTYPE of a macroblock is 3, 4 or 5. Moreover, the waste of bits is relevant with the occurrence frequency of the code number 3, 4 and 5. If the waste of fractional bits occurs frequently, the coding performance will be degenerated. That is why

Table 2. Some comparisons between joint coding scheme and independent coding scheme of maximum MBTYPE and CBP.

CBP			maximum MBTYPE			Joint coding of maximum MBTYPE and CBP			Saving of bits
Code numbers	Code words	Length	Code numbers	Code words	Length	Code numbers	Code words	Length	
0	1	1	5	00110	5	5	00110	5	1
1	010	3	5	00110	5	6	00111	5	3
2	011	3	5	00110	5	7	0001000	7	1
3	00100	5	5	00110	5	8	0001001	7	3
4	00101	5	5	00110	5	9	0001010	7	3
5	00110	5	5	00110	5	10	0001011	7	3
6	00111	5	5	00110	5	11	0001100	7	3
7	0001000	7	5	00110	5	12	0001101	7	5
…	…		…	…		…	…		…

Arithmetic Coding substitutes for the Variable Length Coding for achieving higher coding performance (around 0.5 dB reported in H.264/AVC) [5]. Usually, the syntax element MBTYPE is followed by the CBP and they are independently coded in bitstream. In joint coding scheme, the two syntax elements are sometimes jointly coded together to make full use of the fractional coded bits. The joint coding scheme can be implemented by coding the sum of values of maximum MBTYPE and CBP, i.e.

$$J=M+C \quad (2)$$

where J is the sum of values of maximum MBTYPE M and CBP C, which will be coded into bitstream in stead of coding M and C independently. The comparisons between independent coding and joint coding of maximum MBTYPE and CBP are shown in Table 2, where the code numbers means the values or mapping values of the syntax elements, code words means the zero-order Exp-Golomb codes of the code numbers, and length means the number of coded bits of the code word. First several values of MBTYPE and CBP in AVS are listed for illustrations in Table 2. In independent coding scheme, the code words corresponding to the values of MBTYPE and CBP are input into bitstream in turn. However, in joint coding scheme, the code word corresponding to the sum of the values of maximum MBTYPE and CBP is input into bitstream. For example, a block is coded as Intra 8x8 mode, whose code number is 5 in AVS, and the value of CBP of the block is 1. In independent coding case, the Intra 8x8 mode and CBP are respectively coded as 00110 and as 0001000. As a result, the 12-bit sequence 001100001000 for syntax elements MBTYPE and CBP appears in bitsteam. However, in joint coding case, the sum of code numbers of MBTYPE and CBP is 12 (5 adds 7), which is coded as 0001101. As a result, the only 7-bit sequence 0001101 in stead of 12-bit sequence 001100001000 for MBTYPE and CBP appears in bitsteam. Apparently, 5 bits (12 subtract 7 bits) are saved. The percentage of bits saving in joint coding is about 42%. Theoretically, the average length of code words is shortest in joint coding case. In general cases, joint coding of maximum MBTYPE and CBP will save several bits. In the worst case, nothing is saved. Thus, in any case the joint coding mode added in codec never brings any performance loss. The saving of bits of joint coding case compared with independent coding case is also appended to Table 2 to specify it.

The work flows of joint coding of maximum MBTYPE and CBP in both encoder and decoder are described as follows:

In encoder:

When the code number of the MBTYPE of current coded block is that of the maximum MBTYPE, the sum of code numbers of the maximum MBTYPE and the following CBP is coded as MBTYPE in bitstream and the CBP does not appear in bitstream any more, otherwise the code numbers of the MBTYPE and the CBP are still independently coded into bitstream like original code methd.

In decoder:

When the code number of MBTYPE of currently decoded block is greater than or equal to the code number of the maximum MBTYPE, the MBTYPE of the block is decoded according to the maximum MBTYPE, and the code number of its CBP is not obtained from bitstream but is calculated by the code number of the MBTYPE subtracting that of the maximum MBTYPE.

4. EXPERIMENTAL RESULTS

The proposed scheme is implemented into the reference software of AVS1.0 RM30. Five HD sequences with different characteristic (*city*, *crew*, *night*, *spincalendar* and *harbor*) are used for testing. And each sequence is coded with four fixed QPs, 28, 32, 36 and 40. Since the method is not applied in I frame, only the first picture is coded as I-frame and others are coded as P-frames or B-frames. Two B-frames are inserted between two P-frames, i.e. the picture structure is IBBPBBP…. 600 frames for each sequence are coded. RDO and Loopfilter are enabled and ME/MC range is 48x48 pixel window.

The proposed scheme is compared with independent coding scheme. The coding performance of proposed scheme is measured by the gains of luma PSNR of coded sequences. The savings of bitsrates of coded sequences are also converted into the gains of luma PSNR with the calculation tool [6] for unitary comparisons. The compared results are shown in Table 3. The results indicate that the proposed scheme achieve the coding gains for all sequences. Especially to some sequences having more blocks coded with maximum MBTYPE, such as sequences *crew* and *spincalendar*, the gain of luma PSNR is more than 0.1 dB, which implies that to the class of sequences with such coding characteristics as *crew*, the proposed scheme is able to improve coding performance efficiently. Moreover, the results also indicate that, to the sequences having few blocks coded with maximum MBTYPE, the PSNR does not drop at least.

Table 3. The comparisons of luma PSNR of test sequences coded with and without proposal method.

Test Sequences (1280x720@60Hz)	Gains of Luma PSNR (dB)
city	*0.006585*
crew	*0.132227*
night	*0.037313*
spincalendar	*0.106594*
Harbor	*0.012279*

5. CONCLUTIONS

In this paper, a simple coding scheme is proposed for joint coding the syntaxes of MBTYPE and CBP instead of independent coding them in AVS. The scheme makes use of the bits which are unused by MBTYPE to code CBP partly so that fractional bits are more efficiently used. As a result, the average length of coding bits is shortened and the total bitrate is reduced. The performance improvement of the proposed method depends on the number of macroblock coded by maximum MBTYPE. To the sequences with more blocks coded by maximum MBTYPE, the performance improvement is significant. Moreover, the proposed scheme can substitute for original scheme very easily. It is implemented with a very slight modification and negligible complexity increase on original scheme.

6. ACKNOWLEDGMENTS

This work is partially supported by the key project of National Natural Science Foundation of China under grant No. 60333020 and Natural Science Foundation of Beijing under grant No. 4041003.

7. REFERENCES

1. ITU-T Recommendation H.262. ISO/IEC 13818-2 (MPEG-2): Generic coding of moving pictures and associated audio information. Part 2: Video ITU-T and ISO/IEC JTC1, November (1994)
2. “Draft ITU-T recommendation and final draft international standard of joint video specification (ITU-T Rec. H.264/ISO/IEC 14 496-10 AVC),” in Joint Video Team (JVT) of ISO/IEC MPEG & ITU-T VCEG，JVT-G050, (2003)
3. Golomb, S.W. (1966). Run-length encodings. IEEE Transactions on Information Theory, IT-12(3):399-401
4. R. F. Rice, "Some Practical Universal Noiseless Coding Techniques," Jet Propulsion Laboratory, Pasadena, California, JPL Publication 79--22, Mar. (1979)
5. 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. 620–644, Jul. (2003)
6. G. Bjontegaard, “Calculation of average PSNR differences between RD-Curves”, Doc. VCEG-M33, Mar. (2001)