May 2006, Vol.21, No.3, pp.315–322 J. Comput. Sci. & Technol.

Context-Based 2D-VLC Entropy Coder in AVS Video Coding Standard

Qiang Wang[1] (王　强), De-Bin Zhao[1,2] (赵德斌), and Wen Gao[2] (高　文)

[1] *Department of Computer Science and Technology, Harbin Institute of Technology, Harbin 150001, P.R. China*

[2] *Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100080, P.R. China*

E-mail: {qwang, dbzhao, wgao}@jdl.ac.cn

Received October 30, 2005; revised March 14, 2006.

Abstract In this paper, a Context-based 2D Variable Length Coding (C2DVLC) method for coding the transformed residuals in AVS video coding standard is presented. One feature in C2DVLC is the usage of multiple 2D-VLC tables and another feature is the usage of simple Exponential-Golomb codes. C2DVLC employs context-based adaptive multiple table coding to exploit the statistical correlation between DCT coefficients of one block for higher coding efficiency. Exp-Golomb codes are applied to code the pairs of the run-length of zero coefficients and the nonzero coefficient for lower storage requirement. C2DVLC is a low complexity coder in terms of both computational time and memory requirement. The experimental results show that C2DVLC can gain 0.34dB in average for the tested videos when compared with the traditional 2D-VLC coding method like that used in MPEG-2. And compared with CAVLC in H.264/AVC, C2DVLC shows similar coding efficiency.

Keywords entropy coder, context-based, video coding, VLC

1 Introduction

Entropy coding plays an important role in image and video compression. Its basic function is to eliminate data redundancy existing in source symbols and therefore supply an efficient and equivalent expression for the source, where "equivalent" means the expression is reversible and uniquely lossless decodable. Ideally, we expect to use the minimal bits to express the source, its bound equal to the source entropy. To reach this object, a common scheme is, first to estimate the probability $P(\cdot)$ for the current source symbol to be coded, and then to use $-\log P(\cdot)$ bits to signal it (logarithm are taken to the base 2). Generally these two steps are called *modeling* and *coding*[1,2], respectively. In the practical applications, the sources are always not independent and identically distributed so that entropy coding relies on context modeling to get source symbols' conditional probability for further coding efficiency.

In block-based hybrid video codec, the transformed residual data is a kind of important source. It accounts for the major percentage of total coding bits in most applications. So the coding efficiency of the whole video codec is tightly relative to how efficiently the residual data are entropy coded. The statistical characteristics of the transformed residual data exhibit nonstationariness. Under different coding conditions, such as different quantization step sizes and different coding frame rates, the residual data will have diverse statistical behaviors. Even in the same coding condition, the residual data coming from different sequences or the same sequence but in different local parts will also comply with different probability distributions. So how to identify the local statistical variations based on context information and estimate the conditional probability is getting more and more important for higher coding efficiency.

A natural realization of the upper context *modeling* (conditional probability estimation) and *coding* is adaptive arithmetic coding. Adaptive arithmetic coding always contains two operation modules, one is context modeling which performs the collection and analysis of the previous coded symbols and outputs the conditional probability for the next symbol, the other is coding engine which performs arithmetic coding based on the assigned probability. Especially, the probability can be any precise if the implementation cost is not considered and can be any value with no limitation of probability smaller than 0.5. These inherit features enable arithmetic coding with high coding efficiency. Previous video coding standards have adopted arithmetic coding as an efficient tool. Earlier in H.263[3], an optional arithmetic coder is specified in the normative part, which contains multiple probability distribution models which are selected automatically in coding process. It expresses a simple context adaptive mechanism but has limited ability for tracking probability variation. After that, a sophisticatedly designed arithmetic coder, named Context-based Adaptive Binary Arithmetic Coding (CABAC)[4] was proposed and adopted by H.264/AVC[5]. CABAC combines an adaptive binary arithmetic coder with many well-designed context models to fulfill adaptation to the symbol statistical behavior and results in significant coding efficiency improvement. Although CABAC brings high coding efficiency and perfectly matches the paradigm of modeling and coding for source compression, in some cases its hardware implementation cost is still unaffordable, which limits its application.

Regular Paper

Supported by the National Natural Science Foundation of China under Grant No. 60333020 and the Natural Science Foundation of Beijing under Grant No. 4041003.

Another important class of entropy coding methods is Variable Length Coding (VLC). VLC-based entropy coding has been popular in video coding standards and market products since it is regarded as a better compromise between complexity and efficiency. But in the view of context modeling, VLC-based entropy coding is not as powerful and efficient as arithmetic coding. Because in VLC coding only can integer bits be assigned to a symbol, which cannot match the probability of not 2^{-i} distribution (arithmetic coding can do). An attempt of context modeling based on VLC codes is LOCO-I[1] for lossless and near-lossless image compression. LOCO-I utilizes the context modeling technique and on-the-fly estimates the parameters of G-R[6] codes for VLC coding. Higher coding efficiency is obtained, but not remarkable. If applied in real-time video encoding and decoding applications, LOCO-I's parameter estimation and context model update are still high complexity sensitive. So the design of VLC-based entropy coder in video coding should still pursue a better compromise between complexity and efficiency, and take some simple, straightforward and efficient context adaptation technique for higher coding efficiency.

VLC coding based on fixed VLC tables were used in previous video coding standards like MPEG-2/4[7,8] and H.263. It is a very simple method. To obtain further coding efficiency, some simple adaptations are used. MPEG-2/4 uses different VLC tables for intra- and inter-predicted residual blocks. H.263 adds optional advanced INTRA coding mode and alternative INTER VLC mode[3] to gain some adaptation. But all these VLC-based coders only use one single VLC table when a transformed residual block is coded. So the context correlation existing in one block cannot be fully exploited, which results in low efficiency. In the transformed residual data block, such as the DCT block in hybrid video coding, it is a well-known statistical observation that as DCT subband frequency increases, the magnitude of nonzero coefficient gets smaller and the run-length of successive zero coefficients becomes longer, which also has been pointed out in image coding[9]. This indicates a kind of statistical context correlation for further compression. Context-based Adaptive Variable Length Coding (CAVLC)[10,11] designed for 4 × 4 DCT in H.264/AVC makes use of this context to reduce inter-coefficients redundancy and obtains higher coding efficiency.

For superior coding performance the design of C2DVLC also exploits the above mentioned context correlation in DCT block, but with different ways compared to CAVLC. Two requirements are fully considered in the design to make C2DVLC a good compromise between efficiency and complexity: 1) the coder should be suitable for 8 × 8 DCT which is used in AVS1-P2 video coding standard[12]; 2) the exploration of the context correlation should be both simple and effective which can be viewed as a low complex mode of the context modeling technique. The first requirement comes from the fact that CAVLC will arouse high table storage requirement if directly implemented in 8 × 8 transform size. The second is that the context exploration can be highly complex if no constraint is set. The goal of C2DVLC is to obtain the easier and relatively bigger efficiency improvement at the cost of smaller complexity increase.

The rest of the paper is organized as follows. Section 2 overviews the design of C2DVLC and gives its algorithm flowchart to illuminate how a DCT block is entropy coded. Section 3 explains in detail the C2DVLC design and the underlying ideas. Section 4 provides the experimental results and performance comparison. The conclusions are presented in the last section.

2 Overview of C2DVLC

In AVS video, zig-zag scan[13] is used to translate two-dimensional DCT coefficients into a sequence of (Run, Level) pairs, where Level indicates a nonzero coefficient and Run indicates the number of successive zero coefficients before a nonzero one. This translation makes the above stated context correlation in DCT having a new form. That is, in the statistical view, Levels show a decreasing tendency with respect to the magnitudes while Runs show an increasing tendency. Further, it can be seen that a Level with small magnitude is more likely to be preceded by a large Run and the probability of being preceded by a large Run diminishes while Level's magnitude increases. From this new form, we can get two pieces of knowledge: 1) Run and Level are correlated and 2) the probability distribution of Run and Level combinations varies along the (Run, Level) sequence. So C2DVLC uses a single VLC code to jointly encode Run and Level to exploit their correlation. We define this kind of coding as Two-Dimensional VLC coding (2D-VLC). To exploit the second knowledge, C2DVLC utilizes multiple 2D-VLC tables for the probability distribution variation. To realize context-based adaptive switch of these tables in coding process, a function of the previously coded Levels' magnitudes is used as an indicator to identify the big distribution variations and decide the corresponding tables used. This function is simply defined as producing the maximal value. Specially, the coding is in the reverse zig-zag scan order, which makes it easier to follow the variation based on Level information. A default 2D-VLC table is used to code the first (Run, Level) pair along the reverse scan order. Exp-Golomb[14] codes (E-G codes) are used in C2DVLC for simplicity and efficiency.

Fig.1 illustrates the algorithm flowchart of C2DVLC to highlight how a DCT block is entropy coded. Suppose a DCT block contains N nonzero coefficients and the (Run, Level) pairs after the zig-zag scanning are sequentially stored in the buffer RLBuf$[i]$, $i = 0 \sim (N-1)$. C2DVLC begins the coding at the lattermost (Run, Level) pair, RLBuf$[N-1]$, with the default 2D-VLC table indexed by the variable *TableIndex* $= 0$. If the current (Run, Level) pair is in the range of current 2D-

VLC table TableArr[*TableIndex*], then a table look-up operation is done to find the *CodeNumber* and the corresponding E-G codeword is outputted (*CodeNumber* is a mapping relationship between a (Run, Level) pair and its unique E-G codeword). Otherwise Run and Level are coded separately by escape coding method which will be discussed in the following section. After the coding, *TableIndex* update is needed to decide which table will be used to code the next (Run, Level). The maximal magnitude, *Lmax*, of the previously coded Levels is used to decide how to switch to the next table. Then the coding and table switch are iteratively going on until all the (Run, Level) pairs are coded. At last the flag, EOB, is coded to signal the end of block.

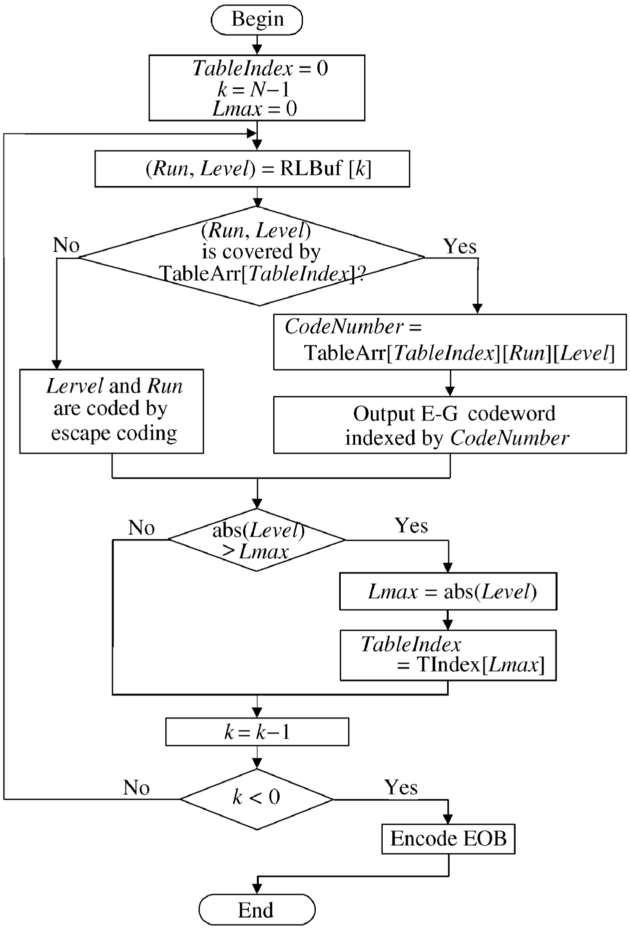


Fig.1. Flowchart of C2DVLC encoding.

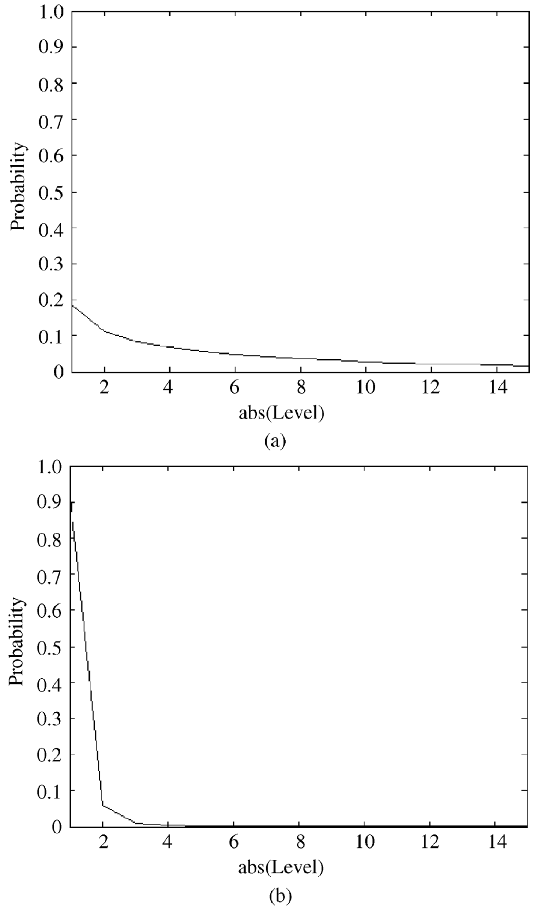
In summary, the main techniques in C2DVLC include: 1) 2D-VLC for Run and Level combination; 2) coding in the reverse zig-zag scan order; 3) coding with multiple 2D-VLC tables; 4) context-based adaptive table switch based on the previously coded Level information; 5) E-G codes.

Except for the transformed residual data, AVS video adopts a finite-extended 0th order E-G codes or sometimes fix length codes to code all other syntax elements. AVS video includes AVS1-P2 and AVS1-P7[15]. In AVS1-P7, the VLC coding scheme for the transformed residual data is a modification of C2DVLC. For detailed information, please refer to [15, 16].

3 Detailed Description of C2DVLC

3.1 Coding in Reverse Zig-Zag Scan Order

C2DVLC utilizes Level's changing tendency to identify the big probability variations, and further designs multiple tables and takes Level-based table switch. Coding in the reverse scan order will make it easier to follow the tendency of Level variation. Fig.2 shows the probability distributions of the first and lattermost nonzero coefficients of DCT blocks in the zig-zag scan order. From the probability curves, we can say, in the global statistical view, Level sequence shows an increasing or decreasing tendency when viewed from different directions. But for a particular Level sequence, the tendency is not monotonically increasing or decreasing. In fact it always shows vibration. So when the Level sequence is viewed along the reverse scan order, it is easy to judge to which degree the increasing tendency has been based on the maximal magnitude of the previously coded Levels. But for the forward scan order it is not so easy and straightforward to determine the degree of the decreasing tendency due to the local tendency vibration.



(a)

(b)

Fig.2. Probability distributions of the first and lattermost nonzero coefficients in the zig-zag scan order. (a) First. (b) Lattermost.

The major reason that C2DVLC adopts a reverse coding order lies in this.

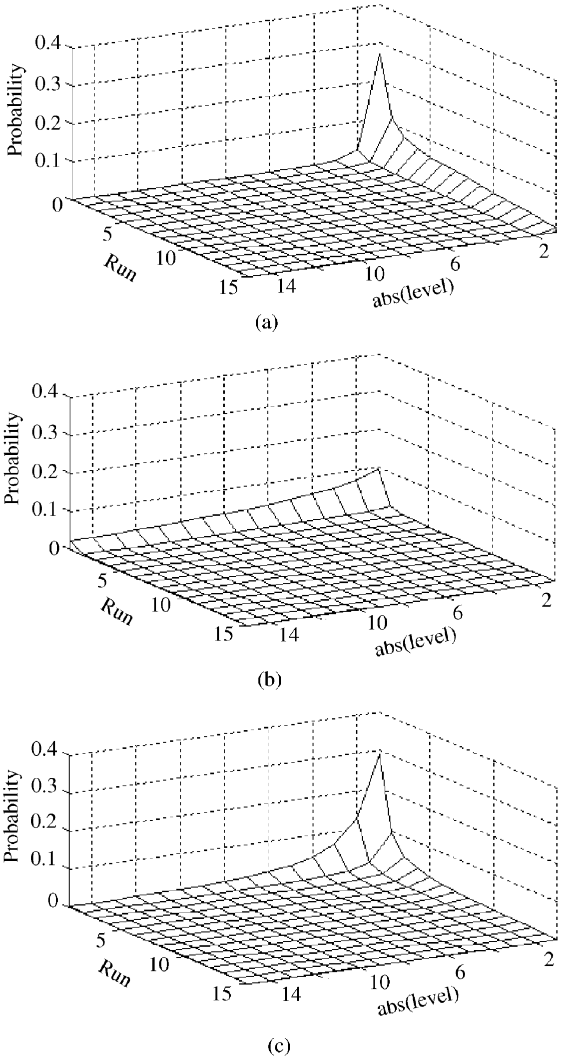


Fig.3. Probability distributions of (Run, Level) pairs. (a) Lattermost (Run, Level). (b) First (Run, Level). (c) All the cases in average.

3.2 2D-VLC Tables

The probability distribution of Run and Level combinations varies progressively along the reverse scan order. Figs. 3(a) and 3(b) show the probability distributions of the lattermost and the first (Run, Level) pairs in the zigzag scan order, respectively. These two distributions can be viewed as the start point and end point of the process of the distribution progressive variation. The intermediate states of this process can be imaginarily interpolated by the two distributions according to the rule that the probability of (Run, Level) with a small Run and a large Level will progressively increase. In fact, from the actual statistical data we also can get the similar intermediate states. Fig.3(c) shows the probability distribution of all the cases of (Run, Level) pairs in a global view. We can see it exhibits a big different shape with Figs. 3(a) and 3(b), because it reflects a global average not a local characteristic. This is also the reason that the traditional VLC coding with a single VLC table, designed to match the global distribution, cannot adapt to local statistical variations and leads to low efficiency. For higher efficient coding of the statistical varying (Run, Level) pairs, a straightforward idea is to utilize multiple tables to match the different local distributions, which is also good at not arousing the high computational complexity as context modeling technique does.

3.2.1 *Table Design*

Now the problem can be depicted as how many tables are needed to achieve the highest efficiency for coding the statistical varying (Run, Level) pairs on the constraint of acceptable table storage size, and how to build these tables. First it can be clarified that not the more the tables are used the higher the coding efficiency can be obtained due to the limitation of E-G codes used in C2DVLC. E-G codes have a regular codeword structure composed of a prefix code and a suffix code as listed in Table 1. When (Run, Level) pairs with different probabilities are coded by E-G codes which have the same length of suffix codes, the probabilities of these (Run, Level) pairs in fact are equal in the view of the same length of the final coding bits. In this case it is not necessary to use multiple tables to adapt to this kind of subtle probability variation due to their utility equivalent to just one table. Therefore, we need to identify those typical distribution variations that are big enough and only assign tables for them. Accompanying these big probability variations the magnitude increases of Level are observed. So we can utilize the Level increase as an indicator to find the superset of the typical probability distributions in the process of distribution progressive variation and design 2D-VLC tables for them. Here a Level increase means a Level with its magnitude bigger than all the previous appears.

Refer to a (Run, Level) sequence as $(r_n, l_n), \ldots, (r_0, l_0)$, which is indexed in the reverse scan order, and define sets S_T as follows,

$$S_0 = \{(r_i, l_i) | i = 0\}, \tag{1}$$

$$\begin{aligned} S_T = \{(r_i, l_i) | k \leqslant i \leqslant k+m,\ m \geqslant 0, &\text{ and for part} \\ \text{of the sequence}(r_{k+m}, l_{k+m}), \ldots, (r_k, l_k), & \\ \mathrm{abs}(l_{k-1}) = T, \mathrm{abs}(l_{k+m}) > T, \mathrm{abs}(l_p) \leqslant T, & \\ k \leqslant p \leqslant k+m-1\}, T > 0. & \end{aligned} \tag{2}$$

At the same time, we suppose sets S_T have the functionality to count the occurrences of a particular (Run, Level) in it. According to the definition of upper sets, when a (Run, Level) sequence is tracked along the reverse scan order, we can put each (Run, Level) pair into one and only one of them. Fig.4 demonstrates how a (Run, Level) sequence is tracked to build the sets S_T.

It can be seen a set S_T collects the (Run, Level) pairs between two successive Level increases including the (Run, Level) incurring the second increase. So with the assistance of the sets S_T, the statistical data for calculating the potential typical probability distributions identified by Level increase can be obtained through building these sets.

To make sure the generality of the obtained typical probability distributions and the final 2D-VLC tables, it is on a set of test sequences representing typical material of HD application and at a range of acceptable visual quality of about 30dB to 40dB that a two-step statistical procedure is used.

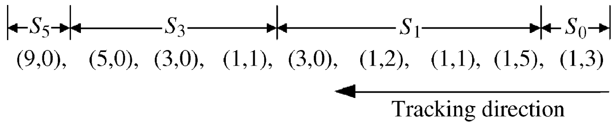


Fig.4. A sample of tracking a (Run, Level) sequence to build sets S_T.

Step 1. Do statistics to get valuable distributions.

Tracking all (Run, Level) sequences produced in coding process in the reverse scan order to build sets S_T, we obtain sets $S_0, S_1, \ldots, S_{19}$, and S_{upper}. S_{upper} is equal to $S_{20} \cup S_{21} \cup \cdots$. In common coding conditions the sets S_T with T bigger than 19 are very small so that it is necessary to combine these infrequent cases to be managed as a whole one. Then these twenty-one sets can produce twenty-one probability distributions of Run and Level combinations, referred to as pmf_i, $i = 0 \sim 20$, because these sets have the ability of memorizing the counter of (Run, Level) instances. We think these *pmf*s are the valuable distributions based on the observation that a Level increase potentially indicates a big distribution variation. For each upper *pmf* the optimal k-th order E-G codes can be decided for realizing VLC coding.

Step 2. Merge to get typical distributions.

It is not economical to design twenty-one 2D-VLC tables corresponding to the upper *pmf*s, because many of them are similar. So a merging procedure is needed to combine the similar sets or *pmf*s into a typical one. A coarse merging can be firstly carried out based on distance calibration between two *pmf*s. A *pmf* can be viewed as a curving surface in the three dimensional space with three axes of Run, Level and probability of (Run, Level). So the distance between two curving surfaces reflects the similarity degree of two *pmf*s. The bigger the distance the lower the similarity, which indicates they should not be merged. We can use common criteria to calculate the distance such as

$$D_{A,B} = \sum_{(r,l) \in S_A \cup S_B} (P_A(r,l) - P_B(r,l))^2. \qquad (3)$$

In (3), $D_{A,B}$ denotes the distance between pmf_A and pmf_B, (r,l) is a basic element of sets S_A and S_B, and $P(r,l)$ represents the (r,l)'s occurring probability. So for upper twenty-one *pmf*s we can get twenty distances D_i, $i = 0 \sim 19$, such as D_0 for pmf_0 and pmf_1, D_1 for pmf_1 and pmf_2 and so on. According to the final expected number K of 2D-VLC tables, we can use the most $K-1$ biggest D_i as separation points and merge the *pmf*s between the two successive points into a typical one. After the coarse merging, a fine merging should be done for some refinements. The fine merging aims to further consider the factors, such as the weightings of *pmf*s and the probability sorting of (Run, Level) pairs, to produce more efficient and reasonable typical distributions in the view of the final 2D-VLC tables.

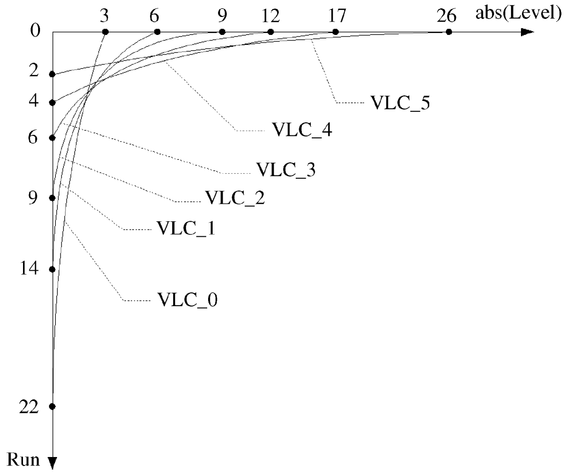


Fig.5. (Run, Level) ranges covered by different 2D-VLC tables.

Through Step 1 and Step 2, we get the typical probability distributions, which depict the most important distribution states followed by the statistical varying (Run, Level) pairs. Then the optimal k-th order E-G codes can be decided for each of these typical distributions for efficient coding. Given the typical distributions and their optimal k-th order E-G codes, 2D-VLC tables can be designed. A 2D-VLC table stores the indexes or *CodeNumbers* of the k-th order E-G codewords, which are mapping relationships between a (Run, Level) pair and its particular E-G codeword. A 2D-VLC table has limited size and only the most frequently occurred (Run, Level) pairs are covered in it. Fig.5 shows the different ranges of (Run, Level) pairs covered by different 2D-VLC tables. The shapes of different tables in Fig.5 also indicate the suitability of the obtained typical distributions, which will bring higher coding efficiency. Specifically, in AVS1-P2 video there exits three groups of tables for intra-predicted Luma, inter-predicted Luma, and Chroma components. As an example, Chroma component has five tables which come from five typical distributions or equally five merged sets as $S_0, S_1, S_2, S_3 \cup S_4$ and $S_5 \cup \cdots \cup S_{upper}$.

3.2.2 Table Switch

The principle of table switch is that the current (Run, Level) pair to be coded must comply with one of upper typical distributions so that it should be coded by the table coming from this typical distribution. Therefore the Level increase is applied to guide the table switch. Another equal description of Level increase is the update of variable $Lmax$, the maximal magnitude of previously coded Levels, because $Lmax$ varies only if a Level increase occurs. Therefore, the rule of table switch can be simply derived from an array look-up operation. For example, for the five 2D-VLC tables, coming from sets $S_0, S_1, S_2, S_3 \cup S_4$ and $S_5 \cup \cdots \cup S_{upper}$ of Chroma component in AVS1-P2 video, the array TIndex[] with $Lmax$ as its input for table switch is as follows,

$$TIndex[] = \{0, 1, 2, 3, 3, 4, 4, 4, 4, \ldots\}. \tag{4}$$

By look-up of upper TIndex[] we can get the next table indexed by TIndex[$Lmax$] to code the next (Run, Level) pair. Therefore the rule of table switch is consistent with the principle of building those typical distributions. One more thing necessary to be testified is whether $S_3 \cup S_4$ is equal to

$$\begin{aligned} S_{3,4} = \{\, (r_i, l_i) | k \leqslant i \leqslant k+m,\ m \geqslant 0,\ &\text{and for part} \\ \text{of the sequence } &(r_{k+m}, l_{k+m}), \ldots, (r_k, l_k), \\ \text{abs}(l_{k-1}) &= 3 \text{ or } 4, \text{abs}(l_{k+m}) > 4, \\ \text{abs}(l_p) &\leqslant 4, k \leqslant p \leqslant k+m-1\}, \end{aligned} \tag{5}$$

where $S_{3,4}$ is the set formed by TIndex[$Lmax$] with $Lmax$ = 3 or 4. From the definition of (2) it is easy to conclude that $S_3 \cup S_4$ is equal to $S_{3,4}$. Thus the rule of table switch exactly matches the rule forming these tables.

From the above description it can be seen that C2DVLC uses multiple pre-designed fix tables to adapt to the local statistical variation in transformed residual data block. No on-the-fly parameter estimation is used and the table switch is simply based on the magnitude information of previously coded Levels. So the context modeling part of C2DVLC is very simple in terms of computational complexity.

3.3 Escape Coding

A 2D-VLC table has limited size and contains the E-G codeword indexes for those most frequently occurring (Run, Level) pairs. For coding the (Run, Level) that is not covered by a table, an escape coding method is used. In escape coding, Run and the sign of Level are jointly coded, and for coding the magnitude of Level a prediction is first taken and then the prediction error is coded. The basic idea for the Level prediction is that the Level magnitude of a (Run, Level) pair not covered by a table will be definitely bigger than the Level magnitudes of (Run, Level) pairs which are covered by the table and have the same Run. So the Level prediction can eliminate this kind of redundancy and improve coding efficiency. E-G codes are also adopted in escape coding. For more detailed information, please refer to [17].

3.4 Exp-Golomb Codes

For the final VLC coding, codewords are constructed based on Exp-Golomb codes. In AVS1-P2 video, k-th order Exp-Golomb codes with k equal to 0, 1, 2 and 3 are used. Table 1 lists part of the codewords (E-G codes) when k's values are 0 and 1. We can see a codeword has regular code structure, which is a concatenation of a prefix code and a suffix code. Given a CodeNumber N and a specific order k, the prefix part consists of l zeros followed by one 1 and the suffix part is the binarization representation of value $N - 2^k\ (2^l - 1)$. l is given by

$$l = \min\{0, \lceil \log_2((N+1)/2^{k+1} + 1/2) \rceil\}. \tag{6}$$

Due to the regular codeword structure, Exp-Golomb codes can be real-time constructed in coding process without involving high computational complexity. So the entries stored in 2D-VLC tables could be mapping relationships (CodeNumbers) from Run and Level combinations to E-G codewords instead of real codes like Huffman codes in MPEG-2. It is a valuable feature that resolves the problem of high memory requirement resulted from multiple 2D-VLC tables.

Table 1. 0th and 1st Order Exp-Golomb Codes

CodeNumber	Codeword			
($N = 0 \sim 6$)	$k = 0$		$k = 1$	
	Prefix	Suffix	Prefix	Suffix
0	1	–	1	0
1	01	0	1	1
2	01	1	01	00
3	001	00	01	01
4	001	01	01	10
5	001	10	01	11
6	001	11	001	000

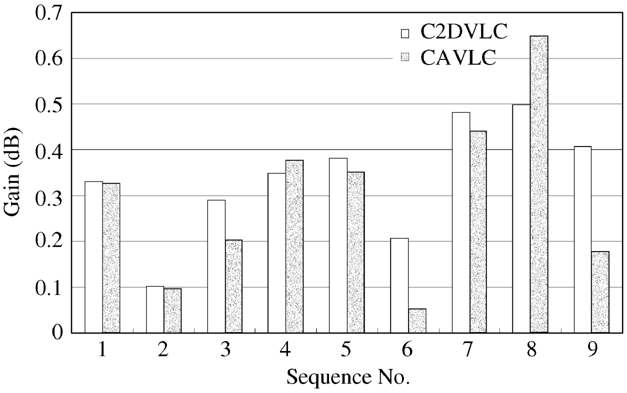


Fig.6. C2DVLC and CAVLC coding capability relative to ABT 8 × 8 VLC in terms of rate-distortion performance.

4 Experimental Results

This section reports the coding performance of C2DVLC. Some typical progressive and interlaced sequences are used as listed in Table 2. RM52c platform, which is developed by AVS working group as AVS1-P2 reference software, is used. The coding conditions are IBBP order with only the first frame coded as I frame, two reference frames, 1/4-pixel motion vector resolution,

+/−32 pixel motion search range, RDO on, and for interlaced sequences PAFF is used.

As the comparing counterparts of C2DVLC, CAVLC in H.264/AVC and the VLC coding method for 8 × 8 block depicted in [18] for Adaptive Block-size Transform (ABT)[19] are used. CAVLC is applied to code the 8 × 8 DCT coefficients at RM52c platform by the same way as used in H.264/AVC High Profile[20]. Fig.6 shows that C2DVLC has the similar coding efficiency as CAVLC. As for the 8 × 8 VLC in ABT, it follows the traditional VLC coding method which uses only one table when a block is coded. It is like MPEG-2 but differs at using Exp-Golomb codes instead of Huffman codes. From Fig.6, it can be seen that C2DVLC outperforms ABT 8 × 8 VLC. The improvement is about 0.34dB in average and for fireworks sequence we can gain up to 0.50dB. This indicates the effectiveness of multiple tables and their adaptive switch.

Table 2. Test Sequences

No.	Name	Interlace/ Progressive	Resolution	Frame rate	Number of frames
1	City	Progressive	1280 × 720	60Hz	60
2	Crew	Progressive	1280 × 720	60Hz	60
3	Night	Progressive	1280 × 720	60Hz	60
4	Spincalendar	Progressive	1280 × 720	60Hz	60
5	Tempete	Interlace	704 × 480	30Hz	120
6	Football	Interlace	704 × 480	30Hz	120
7	Mobile & Calendar	Interlace	720 × 576	25Hz	120
8	Fireworks	Interlace	1920 × 1080	30Hz	60
9	Kayak	Interlace	1920 × 1080	30Hz	60

5 Conclusion

The paper describes C2DVLC entropy coder defined in AVS1-P2 video coding standard and explains the underlying designing ideas in detail. The experimental results show that better coding efficiency can be achieved compared to the traditional 2D-VLC coding method where no multiple tables are used in coding one DCT block. The efficiency improvement comes from two aspects, one is using multiple tables to adapt to the probability variation of Run and Level combinations in a DCT block, and the other is the automatic context-based adaptive table switch to avoid transmitting side-information. Another valuable feature of C2DVLC is low complexity with respect to storage requirement and computational time. This comes from Exp-Golomb codes which keep multiple tables with low storage size and the simple context modeling that multiple pre-designed fix tables are used with a simple switch rule instead of on-the-fly parameter estimation.

References

[1] Rissanen J. Universal coding, information, prediction, and estimation. *IEEE Trans. Inform. Theory*, 1984, 30(4): 629–636.

[2] Weinberger M J, Seroussi G, Sapiro G. The LOCO-I lossless image compression algorithm: Principles and standardization into JPEG-LS. *IEEE Trans. Image Processing*, 2001, 9(8): 1309–1324.

[3] ITU-T. Video coding for low bitrate communications. Version 1, ITU-T Recommendation H.263, 1995.

[4] 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, 13(7): 620–636.

[5] ITU-T. Advanced video coding for generic audiovisual services. Version 1, ITU-T Recommendation H.264, 2003.

[6] Rice R F. Some practical universal noiseless coding techniques—Part I–III. Tech. Rep. JPL-79-22, JPL-83-17, and JPL-91-3, Jet Propulsion Laboratory, Pasadena, CA, Mar. 1979, Mar. 1983, Nov. 1991.

[7] ISO/IEC JTC1. Generic coding of moving pictures and associated audio information—Part 2: Video. ISO/IEC 13818-2 (MPEG-2), 1994.

[8] ISO/IEC JTC1. Coding of audio-visual objects—Part 2: Visual. ISO/IEC 14496-2 (MPEG-4 visual version 1), 1999.

[9] Lakhani G. Optimal Huffman coding of DCT blocks. *IEEE Trans. Image Processing*, 2004, 14(4): 522–527.

[10] Bjontegaard G, Lillevold K. Context-adaptive VLC (CVLC) coding of coefficients. Joint Video Team of ISO/IEC JTC1/SC29/WG11 & ITU-T SG16/Q.6 Doc., Fairfax, VA, May 2002, JVT-C028.

[11] Au J. Complexity reduction of CAVLC. Joint Video Team of ISO/IEC JTC1/SC29/WG11 & ITU-T SG16/Q.6 Doc., Klagenfurt, Austria, July 2002, JVT-D034.

[12] Final draft of information technology—Advanced coding of audio and video—Part 2: Video. AVS Working Group Doc., Shanghai, China, Sept. 2005, N1214.

[13] Tescher A G, Cox R V. An adaptive transform coding algorithm. In *Proc. IEEE XII ICC,* June 1976, pp.20–25.

[14] Teuhola J. A compression method for clustered bit-vectors. *Inform. Process. Lett.*, 1978, 7(6): 308–311.

[15] Final draft of information technology—Advanced coding of audio and video—Part 7: Mobile video. AVS Working Group Doc., Shanghai, China, Sept 2005, N1208.

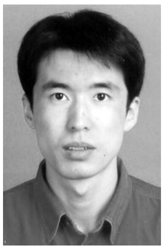
[16] Dong J, Wang J, Yu L. Entropy coding scheme for 4 × 4 residues and improvement for CBP. AVS Working Group Doc., Beijing, China, Nov. 2004, M1455.

[17] Dong J, Lou J, Yu L. Improved entropy coding method. AVS Working Group Doc., Beijing, China, Dec. 2003, M1214.

[18] Wiegand T. Joint final committee draft (JFCD) of joint video specification. Joint Video Team of ISO/IEC JTC1/SC29/WG11 & ITU-T SG16/Q.6 Doc., Klagenfurt, Austria, July 2002, JVT-D157.

[19] Wien M. Variable block-size transforms for H.264/AVC. *IEEE Trans. Circuits Syst. Video Technol.*, 2003, 13(7): 604–613.

[20] Joint Video Team of ITU-T and ISO/IEC JTC 1. Draft ITU-T recommendation and final draft international standard of joint video specification. Joint Video Team of ISO/IEC JTC1/SC29/WG11 & ITU-T SG16/Q.6 Doc., Redmond, WA, July 2004, JVT-L050.



Qiang Wang received the B.S. and M.S. degrees in computer science and technology from Harbin Institute of Technology (HIT), China, in 2001 and 2003, respectively, where he is currently working toward the Ph.D. degree. Since 2002, he has been with the Joint R&D Lab (JDL), Chinese Academy of Sciences. His research interests mainly focus on image and video compression.

De-Bin Zhao received the B.S., M.S. and Ph.D. degrees in computer science all from Harbin Institute of Technology (HIT), China, in 1985, 1988 and 1998, respectively. After graduated he joined Dept. Computer Science, HIT. He was promoted to associate professor in 1993. Now he is a professor of HIT and Institute of Computing Technology, Chinese Academy of Sciences. He has been a research fellow at Department of Computer Science, City University of Hong Kong. His research interests include multimedia compression and its related applications. He has authored or co-authored over 100 publications.

Wen Gao received the M.S. and Ph.D. degrees in computer science from Harbin Institute of Technology, China, in 1985 and in 1988, respectively, and the Ph.D. degree in electronics engineering from the University of Tokyo, Tokyo, Japan, in 1991. He was a research fellow at Institute of Medical Electronics Engineering, University of Tokyo, Tokyo, Japan, in 1992, and a visiting professor at Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, in 1993. From 1994 to 1995 he was a visiting professor at the MIT AI Lab, Massachusetts Institute of Technology (MIT), Cambridge, MA. Currently, he is the vice president of the University of Science and Technology of China, the deputy president of Graduate University of Chinese Academy of Sciences, professor of Computer Science at Harbin Institute of Technology, and the honor professor in Computer Science at City University of Hong Kong. He is the head of Chinese National Delegation to the MPEG Working Group (ISO/SC29/WG11). He has published seven books and over 400 scientific papers. His research interests are in the areas of signal processing, image and video communication, computer vision, and artificial intelligence. Dr. Gao chairs the Audio Video coding Standard (AVS) working group of China. He is also the Editor-in-Chief of the Chinese Journal of Computers and the general Co-Chair of the IEEE International Conference on Multimodel Interface in 2002.