2005 WSEAS/IASME Int. Conf. on ELECTROSCIENCE and TECHNOLOGY for NAVAL ENGINEERING and ALL-ELECTRIC SHIP
Miami, Florida, USA, November 17-19, 2005 (pp34-43)

Use of Probability Distribution Matching in H.26L For Adaptive Configuration of UVLC –A Novel Approach

AJITH KUMARAYAPA,ZHANG XIAOFENG, ZHANG YE
School of Electronics and Information Engineering, Dept. of Electronics and Communication Engin.
Harbin Institute of Technology(HIT)
P.Box.314,No.92,West Da-Zhi Street, Harbin 150001

***Abstract:* - Popular H.26L video coding standard employ UVLC as one of its entropy coding technique, since it provides an easy way to encode all syntax elements in a regular form. However, the disadvantage of the UVLC is that it failed to conform the symbol probability distribution to get the best coding result, specially with small quantization parameter (i.e. QP) values. We propose a navel approach with Probability Distribution Matching (PDM) which has advantage of obtaining the configuration for encoder and decoder in the same way without having any overhead information. More over, PDM is used to choose the most suitable configuration from a predetermined configuration table according to already encoded symbol probability distribution. Experimental results convince that the use of PDM for UVLC is a convenient method to adjust the configuration. Thus, the proposed method is beneficial for almost most of the configurable Variable Length video coding applications.**

***Key-words:* - UVLC (Universal Variable Length Coding), Probability Distribution Matching (PDM),Entropy coding(EC)**

1 Introduction

H.26L is a more popular video compression standard being jointly developed by ITU-T video Expert Group and ISO/IEC Motion Picture Expert Group.The main theme of this standardization is to enhance compression performance and to prove a packet based network friendly video representation. It has number of excellent features[1] such as 1.Up to 50% in bit rate savings compared to H.263v2(H.263+) or MPEG-4 Simple Profile, 2.High quality video consistently at more bit rates,3.Adaptation to delay constraints as it can operate in a low-delay mode to adapt to real-time communications applications (e.g.video conferencing),4.Error resilience, 5.Network friendliness:e.g.,the conceptual separation between a Video Coding Layer (VCL) which provides the core high-compression representation of the video picture content, Network Adaptation Layer (NAL).The primarily applications are Video Conferences (communications) over the internet, mobile wireless channels etc.Similer to traditional video coding

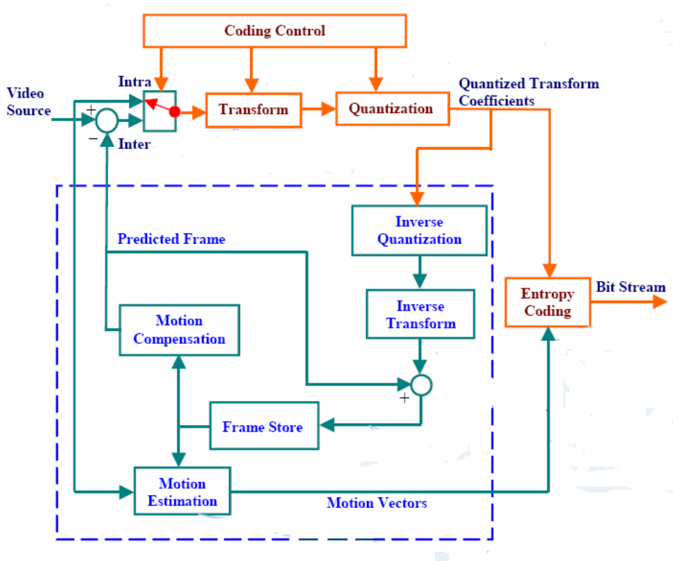

Fig 1 Block diagram of the H.26L encoder.

standards like H.263 or MPEG-4, the underlying process is made up of prediction,transformation, quantization and entropy coding, as shown in Fig 1.[1]. H.26L has adopted two entropy coding techniques, one is based on Universal Variable Length Codes(UVLCs) and the other is based on Context-Based Adaptive Binary Arithmetic Coding (CABAC).UVLC is the most widly used method for the compression of quantized transform coefficients, motion vectors, and other encoder information. VLCs are based on assigning shorter codewords to symbols with higher probabilities of occurrence, and longer codewords to symbols with less frequent occurrences. The symbols and the associated codewords are organized in look-up tables, referred to as VLC tables, which are stored at both the encoder and decoder[1].Contrary to some other general video coding standards (H.263, MPEG-1/2/4) which use number of VLC tables ,H.26L use only a single Universal VLC for entropy coding of any type of data symbols in the encoder.

The UVLC has many useful features such as non necessity of look-up tables, simplified encoding and decoding process, easy extensibility and error resilience, etc[2]. Nevertheless, UVLC has major draw backs; the UVLC single table is usually derived using a standard probability distribution model,which ignore the correlation between the encorder symbols.Moreover, UVLC cannot handle a wide range of bit-rates as desired due to the uniformity of its code words, specially with small QP values, the code words produced by UVLC are not in accordance with the symbol probability distribution.



Mobile (CIF) Foreman (QCIF) Container (QCIF)
Fig. 2 The first frames of the video sequences.

Furthermore, it is evident that the inefficiency of UVLC is caused due to two inconsistencies. One is between the probability distribution represented by the code word and the real symbol probability distribution. The other is between the symbol order provided in H.26L and the real symbol probability order. These inconsistencies can be illustrated by Fig.3 graph; the probability distributions of inter Macroblock (MB) type 3 different video sequences (‘Mobile’, ‘Foreman’ and ‘Container’) Vs a Code word lengths ,when QP is equal to 5. The first frames of them are shown in Fig.2.These can be

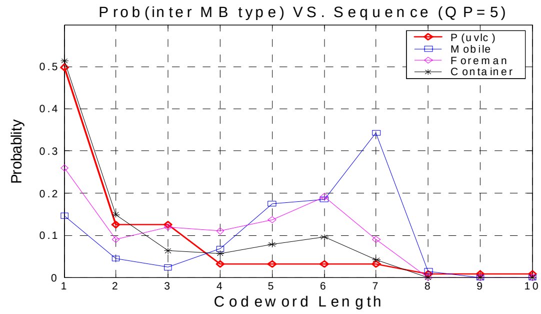

Fig. 3 Probability distribution of symbol “MB type” with respect to different sequences.

compared with the curve of the probability distribution represented by UVLC in H.26L.

Under this consideration, the improvements of UVLC are thoroughly investigated. In [3], the Dynamic Symbol Reordering (DSR) method computes the symbol probability of several encoded frames to get a mapping table, and re-maps the relation between symbols and code words. DSR method is proposed only for a few coding elements, e.g., the MB type that takes only small portion of total bit stream. In [3], a UVLC scheme with adaptive codeword mapping using fixed re-association table (FRAT) is proposed. It takes more symbols into account. Different from the two methods described above, without taking the probability distribution inconsistency into consideration, in [5] the UVLC is configured with additional code configurations (ACC) in order to make VLC configurable. The ACC is used to adapt UVLC to different symbol distributions by adjusting the partitioning of the symbols into different categories, and arraigning the code size assignment to different categories. Since the variability of symbol probability distributions is mainly concerned with the QP value and the image content[2], the configurations used in [6] are assorted with these two key factors. However, it is yet to solve the simply and directly is the undetermined major problem “the way to determine the configurations”, which limits its further applications.

In this paper, we propose navel concept of Probability Distribution Matching (PDM) based on the configurable VLC which adaptively choose configuration for the UVLC. PDM is applicable to determine the configurations of different symbols like MB type, coded block pattern (CBP) and the transform coefficients, and these symbols are encoded with different configuration respectively. The promising validity and efficiency of PDM are proved by our Experimental results.

Section 2 of this paper reviews and analyze the property of configurable VLC. The method of performing the PDM describes in Section 3. Section 4 presents the experimental and results, in which depicts the fact that the UVLC configuration can be easily adjusted using PDM, hence making it more conform to the symbol probability distribution..Section 5 concludes the works.

2 Review and Analysis for Configurable VLC

In this section we simply review the configurable VLC. The codeword of the original UVLC in H.26L follows the format $0x_n0x_{n-1}\ldots0x_001$, $x_nx_{n-1}\ldots x_0$ identifies an individual symbol[6]. The configurable VLC divides the symbols into different categories and assigns different coarse code (prefix) to each category. Within a category, the additional code (suffix) identifies an individual symbol. The coarse code follows the format 00…01. For the k^{th} category, it has (k+1) bits, k 0s followed by one 1. The structure of the additional codes is defined by a configuration.The configuration is a one-dimensional array of L positive integers; [r_k: k=0 to L-1], where r_k is the number of symbols in the k^{th} category. If we assume r_k is a power of 2 (r_k being any other integer will increase the burden of the encoder[7]), r_k can be represented by its power, {p_k: k=0 to L-1}.

$$r_k = 2^{p_k} \qquad (k = 0,1\ldots L\text{-}1) \quad \ldots\ldots\ldots\ldots(1)$$

$$2^{-(k+1+r_k)} \cdot 2^{r_k} = 2^{-(k+1)} \quad \ldots\ldots\ldots\ldots\ldots\ldots (2)$$

The optional entropy coding performance will be made if the symbols within a category have the same code size with similar occurrence probabilities; $2^{-(\text{code size})}$. Table 1 shows the relationship between the original UVLC in H.26L and the configurable VLC.

k	Code NO.	Original UVLC code words in H.26L	Configurable VLC	
			coarse code	additional code
0	0	1	1	
1	1:2	0x1	01	x
2	3:6	0x0x1	001	xx
3	7:14	0x0x0x1	0001	xxx
4	15:30	0x0x0x0x1	00001	xxxx
5	31:62	0x0x0x0x0x1	000001	xxxxx
…	…	…	…	…

Table 1. Relationship between original UVLC and the configurable VLC [p_k: 0, 1,2,3,4,5…]

Fig. 4 illustrates the relation ship between probability and coadwords for several configurations.

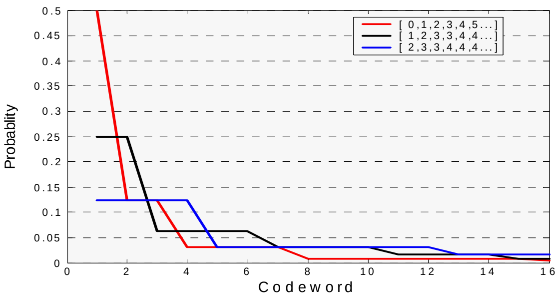

Fig. 4 Probability distributions with different Configurations.

The total probability of the symbols in one category is the same, while the probability of one symbol with different configuration may be different.
For the first 6 categories with different configuration, the total probability $\widetilde{P}$ of the symbols in these can be computed using the equation (3).

$$\widetilde{P} = \sum_{k=0}^{5} 2^{-(k+1)} = 1 - 2^{-6} = 0.984375 \quad \ldots\ldots\ldots.(3)$$

According to (3), most of the symbols are included in the first 6 categories, thus we can relies on them for the major changes of the configuration. Through vast statistic experiments, it is proved that p_k (k=0,1…5) are limited in several values, which are listed in Table 2.

	Value							Default
p_0	0	1	2	3	×			0
p_1	0	1	2	3	×			1
p_2	×	1	2	3	×			2
p_3	×	1	2	3	4	×		3
p_4	×	1	2	3	4	5	×	4
p_5	×	1	2	3	4	5	6	5
p_6	6							6
…	…							…

Table 2. The main values of p_k (k=0,1…5)

With values in table 2, the total number of combinations of p_k (k=0,1…5) is 5760 (the remaining p_k, i.e. k is greater than 5, use the default value k as in the H.26L). However, the greater p_k is, the smaller the corresponding probability of the symbol in the k^{th} category is. And the symbol's alignment conforms to the rule that the greater the occurrence probability of the symbol is, the smaller the code number. So that here two reasonable restrictions are added to p_k (k=0, 1…5):1.The p_k are monotonously incremental.2.The incremental step is no more than 1.Finally, the number of combinations of p_k (k=0,1…5) is reduced to 88 with the subsequent restrictions. Therefore these 88 combinations will be treated as a candidate set for PDM.

3 PDM for Adaptive UVLC

In order to choose the most suitable configurations for UVLC ,motivated by Vector Quantization (VQ), PDM is proposed. However, contrary to the training set concept in VQ , in this work, in order to obtain a book of probability distribution ,the 88 combinations are treated as a candidate set, Moreover, contrary to idea with VQ, here we just directly find the most suitable candidate. The novel PDM algorithm is given in the Fig.5 block diagram.

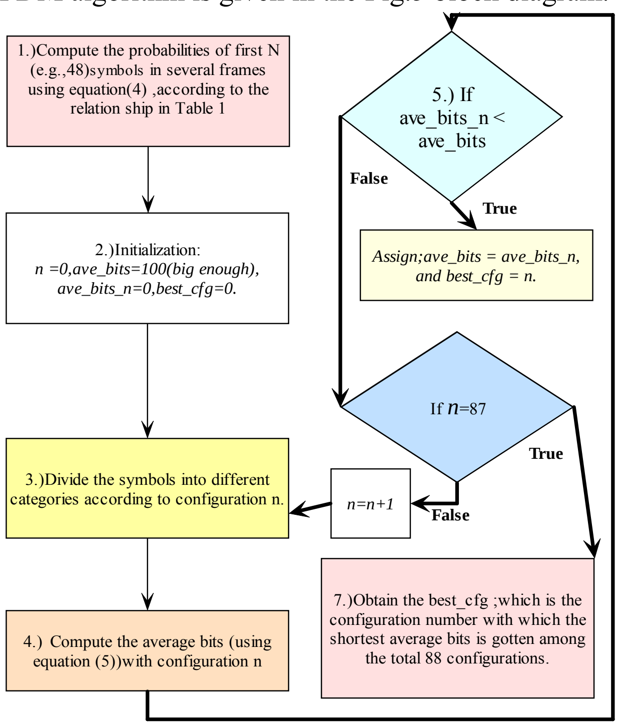

Fig.5 The novel PDM algorithm to choose the most suitable configurations for UVLC.

Note:The symbol meanings and the equations related to Fig. 5.

1) c_i is the number of times that the symbol a_i is occurred.

$$prob_i = c_i / \sum_{j=0}^{N-1} c_j \quad \ldots\ldots\ldots\ldots\ldots\ldots\ldots(4)$$

2) n is the configuration number (0,1…87), *ave_bits* is shortest (among the already computed ones) *ave_bits_n* is average bits using configuration n, *best_cfg* is the current configuration number with which the shortest average bits is gotten.

$$4)\ ave\_bits\_n = \sum_{k=0}^{5} \sum_{i=0}^{2^{p_k}} prob_i \cdot 2^{-(k+1+p_k)} \quad \ldots\ldots(5)$$

PDM is conducted on different symbols such as MB type, intra/inter CBP and intra/inter luma level_run, with different QP values and the most probable configurations are chosen from the candidate set listed in table 3.

No	Configuration					
	P_0	P_1	P_2	P_3	P_4	P_5
0	0	0	1	1	1	1
1	0	0	1	2	3	3
2	0	1	1	1	1	1
3	0	1	2	2	2	2
4	0	1	2	2	3	3
5	0	1	2	3	3	4
6	0	1	2	3	4	5
7	1	1	1	1	1	1
8	1	2	2	2	2	2
9	1	2	3	3	4	4
10	1	2	3	4	4	4
11	2	2	2	2	3	3
12	2	2	3	3	4	4
13	2	3	3	3	3	3
14	2	3	3	4	4	4
15	3	3	3	3	3	3

Table 3. The most probable configurations.

The Table 3 is used as a configuration book, just like the codebook in VQ, and the configuration is adjusted with different symbol probability distribution using PDM(here total configuration number is 16, not 88). Moreover the configuration 6 in Table 3 corresponds to the default UVLC in H.26L and its corresponding probability distribution is shown in Fig. 3(with that of configuration 9 and 14).

It has been pointed out that the symbol probability distribution has great concern with the QP value and the image content, especially inter luma level_run, which takes up great portion of total bit stream. Fig.6 shows the most suitable configurations for inter luma level_run in terms of different QP values and video sequences.

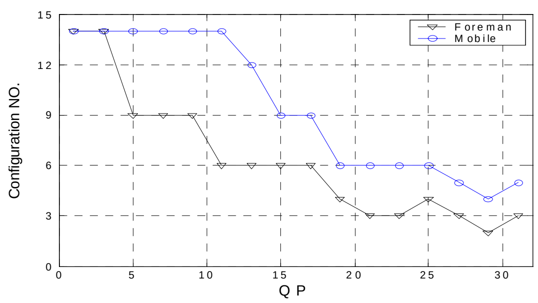

Fig. 6 Most suitable configuration for inter luma level_run

The video sequence 'Foreman' represents the common video content with camera movements while 'Mobile' has more complicate content. According to the graph Fig. 6, it can be concluded that the probability distribution of inter luma level_run is more flat and the configuration number is larger,when video is more complicate and with smaller QP values, and vice versa.

QP	0-4	5-10	11-18	19-31
Cfg.NO.	14	9	6	3

Table 4. Configurations of inter luma level_run with different QP values.

For the experiments in section 4, different configuration is chosen as the start one to encode inter luma level_run with different QP values as shown in Table 4, and later, according to the video content,it will be adjusted.

The main attention is paid to the inter frames. Considering the inconsistency of the symbol order mentioned above, the DSR method[3] is incorporated into the experiments. The experiments are conducted as follows:

1) Encode I frame with default configuration (NO. 6).
2) Encode first several (e.g. 5) P frames using default configuration except inter luma level_run, which uses configuration listed in table 4.
3) Count the occurrences of inter MB type, inter CBP, and inter luma level_run.
4) Use DSR method[3] to adjust the order of inter MB type and inter CBP.
5) Use PDM to get the most suitable configuration for the three kinds of symbols.
6) Encode the following P frames, using configurations determined by step 5 for the three kinds of symbols and using default configuration for the remaining ones.
7) Come across another I frame, go back to step 1.

At the decoder, the same statistic process is used as the encoder to adjust UVLC configurations, so it is unnecessary to send any overhead information about the configurations.

In the experiments, the video sequences 'Container', 'Foreman', and 'Mobile' are encoded with various QP values. For the sake of comparison, we define bit-reduction ratio as,

$$R = \frac{N_1 - N_2}{N_1} \times 100\% \quad \text{..................} \ (6)$$

where N_1 is the number of bits generated by the original TML-97 coder, and N_2 refers to the number of bits generated by our scheme. The bit-reduction ratio of various conditions are shown in the figure7(a),(b),(c).

Fig. 7 The bit-reduction ratio of different symbol in PDM experiments using the H.26L TML-97

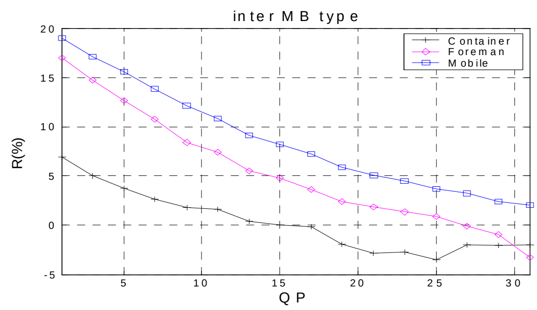

Fig.7 (a) Inter MB type

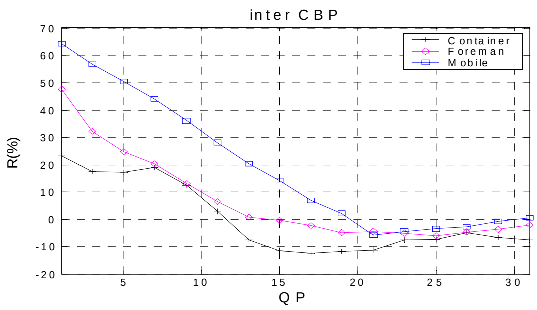

Fig.7 (b) Inter CBP

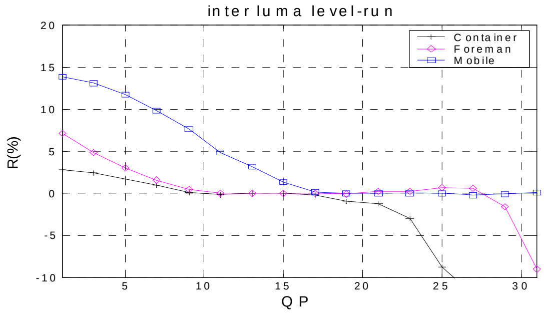

Fig.7 (c) Inter luma level-run

The coding efficiency of each symbol of interest is shown in Fig 7. The question 'whether the coding efficiency is improved or not ' is determined by the condition 'whether PDM works well or not'. For the video sequence such as 'Container', which has smooth content and with slow motion, the prediction is well performed before the entropy coding process, the symbols needed to encode is too little to get good statistics. So the PDM could hardly work well in this situation especially with high QP values (at the same time the total number of bits is too small as well, the negative improvement is negligible). On the contrary, for 'Foreman' and 'Mobile', PDM works well, the improvement for three symbols is up to 19.0%, 64.3% and 13.8% respectively at high bit-rate.

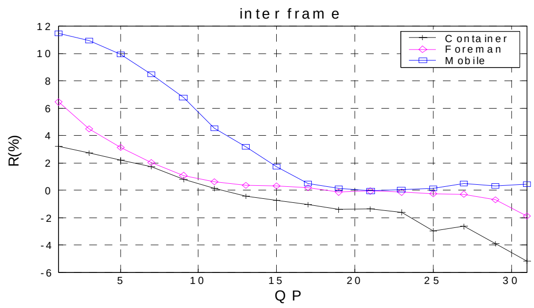

Fig. 8 The bit-reduction ratio of total inter frames in our experiments over the H.26L TML-97

Fig.8 shows the total coding gain of the experiments using the proposed PDM. In consideration of the compression gain of TML-97 coder to TML-5 coder used in [7], the improvements are comparable to that of [7], which testify the applicability and efficiency of the PDM.

6 Conclusions

The novel concept with PDM is proposed as an applicable method to configure the UVLC to conform to symbol probability distribution in H.26L.Employing PDM on coding of several major symbols, experiments are conducted with various QP values and different video sequences. The experimental results show its prominent improvement especially at small QP values (1-20), which convinced the applicability and efficiency of the PDM. Furthermore, with the help of the PDM, configurable VLC can be used efficiently in a wide range of video coding applications, instead of the existing VLC with some limitations .

References

[1] UB Video Inc., www.ubvideo.com White Paper,*Emerging H.26L Standard :Overview and TMS320C64x Digital Media Platform Implementation*,pp1-12, 2002

[2] Yuji Itoh, Bi-directional motion vector coding using universal VLC, *Signal Processing: Image communication 14*, pp.541-557, 1999.

[3] K.-Y. Yoo, B.-S. Choi and Y.-Y Lee, Improvements to the Telenor proposal for H.26L: Preliminary results on Dynamic Symbol Reordering (DSR) method for Universal VLC encoding/decoding, *ITU-T Q.15/16*, Doc. #Q15-H19, Aug.1999.

[4] W.I Choi and B.-W. Jeon, Dynamic UVLC codeword re-mapping using fixed re-association tale for H.26L, *Picture Coding Symposium(PCS)*, Seoul, Korea, pp.167-170, Apr. 25-27,2001.

[5] Yuji Itoh and N.-M. Cheung, Universal variable length code for DCT coding, in *Proc. IEEE Int. Conf. Image Processing(ICIP)*, Vancouver, Canada, Sept. 10-13, 2000.

[6] Thomas Wiegand, H.26L Test Model Long Term Number 9 (TML-9) Draft0, *ITU-T Q.6/16*, Doc. #VCEG-N83, Dec. 2001.

[7] N.-M. Cheung, Louis Kerofsky and Yuji Itoh, Configurable entropy coding scheme for H.26L, *ITU-T Q.6/16*, Doc. # VCEG-L19, Jan. 2001.

[8]ITU-T/SG16/VCEG(Q.6), *H.26L TML-97 software*,ftp://standard.pictel.com/video-site/h26l/tml97.zip.

[9]Tu-Chih Wang,Hung-Chi Fang,Wei-Min Chao, An UVLC Encoder Architecture for H.26L,*IEEE Inernational. Symposium on Circuits and Systems*,pp.308-311,2002