UNIVERSAL VARIABLE LENGTH CODE FOR DCT CODING

Yuji Itoh and Ngai-Man Cheung
Tsukuba Research & Development Center, Texas Instruments Japan Ltd
17 Miyukigaoka, Tsukuba, Ibaraki, 305-0841 Japan

ABSTRACT

A new technique for entropy coding named Universal VLC (i.e., UVLC) was developed and applied to motion vector. It was shown that UVLC for motion vector coding provides good performance in terms of coding efficiency as well as error resiliency. In this paper UVLC for DCT coefficient coding is presented. The features of the scheme are: 1) adaptable by parametric representation, 2) support of very low bitrate mode and 3) extensible to error resilient mode with bi-directionally decodable code. This favors a flexible image and video coding that should handle a variety of picture types and applications. The simulation results show that the proposed scheme slightly outperforms several VLC tables of existing standards in terms of coding efficiency. It is also expected that the proposed scheme can simplify the implementation and processing of VLC decoder. It shall also be noted that UVLC is being considered as a candidate component technique for ITU-T H.26L, i.e., next generation video coding algorithm standard.

1. INTRODUCTION

Variable-length coding (VLC) is a statistical coding technique that assigns codewords to values to be encoded. Values of high frequency of occurrence are assigned short codewords, and those of infrequent occurrence are assigned long codewords. On average, the more frequent shorter codewords dominate such that the code string is shorter than the original data. The VLC tables in MPEG-2 [1] and H.263 [2], for example, are not Huffman tables in the true sense of Huffman coding, but are more like the tables used in Group 3 fax. They are entropy constrained, that is, non-adaptable and optimized for a limited range of bit rates. A better way would be to say that the tables are optimized for a range of ratios of bit rate to sample rate (e.g. 0.25 bit/pixel to 1.0 bit/pixel). It is obvious that DCT VLC table of H.263, which is tuned for low bitrate coding, can't readily handle near lossless still picture coding. Therefore a type of configurable VLC is desirable for generic coding that should be able to deal with various coding conditions. We already invented such VLC technique named Universal VLC (UVLC) [3]. The UVLC was applied to motion vector coding and gave better error resiliency compared to MPEG-4 Video, whereas the coding efficiency is fairly comparable.

In this paper UVLC for DCT coefficient coding is described. This was mainly designed to handle a wide range of bitrates, but has some other merits. The features of the scheme are: 1) adaptable by parametric representation, 2) support of very low bitrate mode and 3) extensible to reversible VLC for error resilient coding. Also the proposed scheme gives an advantage that the VLC decoding process can be significantly simplified.

2. CONVENTIONAL VLC FOR DCT

We have investigated the conventional VLC techniques employed for existing video coding standards. In DCT coefficient coding, a coefficient to be coded is usually represented in terms of three components; last non-zero coefficient indication (LAST - '0': there are more non-zero coefficients in this block, '1': this is the last non-zero coefficient in this block), the number of successive zero coefficients preceding the coded coefficient (RUN), and magnitude of coded coefficient (LEVEL). It was, then, found that there are mainly two types regarding VLC tables; 1) (LAST, RUN, LEVEL) type in H.263 and MPEG-4 [4], 2) {(RUN, LEVEL)+EOB} type in H.261, MPEG-1 and MPEG-2. Note EOB stands for "end of block" which indicates that a coefficient being processed is the last coded one in a block. In H.263, for instance, since LAST, LEVEL and RUN range [0,1], [-127, 127] and [0,63], respectively, up to 32K codewords are present in the VLC table. Among them, 103 codewords are represented as VLC. The remaining combinations of (LAST, RUN, LEVEL) are represented as escape codes. The escape codes are 22 bit word consisting of 7 bits ESCAPE, 1 bit LAST, 6 bits RUN and 8 bits LEVEL.

3. UNIVERSAL VLC

Fig. 1 illustrates the structure of UVLC for DCT coefficient coding. This looks similar to {(RUN, LEVEL)+EOB} type employed in H.261, MPEG-1 and MPEG-2. But the proposed method is unique in the sense that RUN and LEVEL are coded separately, so it is expressed like (EOB+LEVEL+RUN).

0-7803-6297-7/00/$10.00 © 2000 IEEE

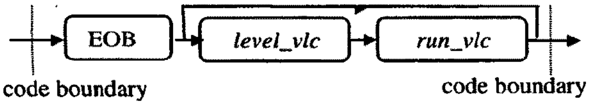


Figure 1. Proposed DCT coefficient coding

3.1. Adaptability

The existing coding standards don't offer configurable VLC that is adaptable to distribution of DCT coefficients in an on-the-fly fashion. So we set the code length of EOB as a parameter, and achieved the adaptability by tuning the parameter. This approach is not optimal in terms of coding efficiency, however, it can provide near optimal efficacy while attaining low implementation cost and easy decoding process. The EOB was designed to save unnecessary run-length codes. MPEG-1 uses EOB of 2 bit length, which implies there is an average of only 3 or 4 non-zero AC coefficients per block. While in MPEG-2 an "intra VLC table" is added, which has been optimized for intra DCT block. The "intra VLC table" has a 4 bit EOB code, that indicates there should exists 9 - 16 coefficients per block. In the proposed scheme, EOB length (named EOB_length) is variable so that the VLC table can support various kinds of DCT coefficient distribution, hence, various application.

Table 1 shows the UVLC table for the run length, i.e., RUN. Each codeword is composed of two parts; coarse code followed by additional code. While the coarse code is true VLC part, the additional code is a sort of fixed length code (FLC) whose length is determined by the preceding coarse code. Also Table 2 and 3 represent the UVLC table for the amplitude, i.e., LEVEL, with 2 bit EOB and 3 bit EOB, respectively. Similarly one can construct LEVEL UVLC tables with arbitrary EOB length. The RUN table (Table 1) is common to both intra and non-intra blocks.

With this technique, as many as 32K VLC entries can be constructed from a dozen of VLC roots, i.e., coarse code. Therefore VLC decoder for the UVLC table only have to search one such code among one dozen of candidate codes (as compared to 103 candidates for H.263). In addition the UVLC doesn't have to deal with escape codes. On the other hand, the proposed scheme should do the search twice, i.e., each for RUN and LEVEL. However, the UVLC can still release computations from the decoding process.

Table 1. RUN UVLC table

run_vlc coarse	additional	code size	value of run
1		1	0
01		2	1
001	x_0	4	"x_0"+2 (2:3)
0001	x_1x_0	6	"x_1x_0"+4 (4:7)
00001	$x_2x_1x_0$	8	"$x_2x_1x_0$"+8 (8:15)
000001	$x_3x_2x_1x_0$	10	"$x_3x_2x_1x_0$"+16 (16:31)
0000001	$x_4x_3x_2x_1x_0$	12	"$x_4x_3x_2x_1x_0$"+32 (32:63)

Table 2. LEVEL UVLC table (EOB_length: 2)

level_vlc coarse	additional	code size	absolute value of level
1	s	2	1
01		2	EOB
001	x_0s	5	"x_0"+2 (2:3)
0001	x_1x_0s	7	"x_1x_0"+4 (4:7)
00001	$x_2x_1x_0$s	9	"$x_2x_1x_0$"+8 (8:15)
000001	$x_3x_2x_1x_0$s	11	"$x_3x_2x_1x_0$"+16 (16:31)
0000001	$x_4x_3x_2x_1x_0$s	13	"$x_4x_3x_2x_1x_0$"+32 (32:63)
00000001	$x_5x_4x_3x_2x_1x_0$s	15	"$x_5x_4x_3x_2x_1x_0$"+64(64:127)
's' denotes the sign of level. '0' for positive and '1' for negative			

Table 3. LEVEL UVLC table (EOB_length: 3)

level_vlc coarse	additional	code size	absolute value of level
1	s	2	1
01	x_0s	4	"x_0"+2 (2:3)
001		3	EOB
0001	x_1x_0s	7	"x_1x_0"+4 (4:7)
00001	$x_2x_1x_0$s	9	"$x_2x_1x_0$"+8 (8:15)
000001	$x_3x_2x_1x_0$s	11	"$x_3x_2x_1x_0$"+16 (16:31)
0000001	$x_4x_3x_2x_1x_0$s	13	"$x_4x_3x_2x_1x_0$"+32 (32:63)
00000001	$x_5x_4x_3x_2x_1x_0$s	15	"$x_5x_4x_3x_2x_1x_0$"+64 (64:127)

3.2. Very low bitrate mode

The proposed scheme is also applicable to very low bitrate coding. Through the statistical analysis, it has been found that the distribution of DCT coefficients is not yet fully taken into consideration to further optimize the VLC table. The following property will be employed in the proposed scheme; P{"LAST:1"} > P{"LAST:0"}, where P{"EVENT"} indicates the occurrence probability of an event "EVENT". To prove this, one may attempt to code DCT coefficients using the modified VLC table in which the meaning of the events "LAST:1" and "LAST:0" are exchanged. In (LAST, RUN, LEVEL) type VLC, for instance, this is simply done by adding the process LAST = LAST⊕1 just prior to coding a coefficient (LAST, RUN, LEVEL). By applying this modification at very low bitrates, one can reduce the bits for DCT coefficients up to several percent whenever P{"LAST:1"} > P{"LAST:0"} is guaranteed. This is because conventional VLC tables were mostly optimized for such a situation that P{"LAST:1"} < P{"LAST:0"} like 4xP{"LAST:1"} ≈ P{"LAST:0"}. Now we propose one-bit flag "nEOB", which stand for "negative EOB", can be tailored in the syntax. This flag shall be set by the rule:

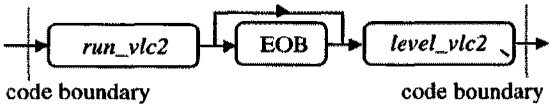
$$nEOB = \begin{cases} 1 & if \quad P\{LAST:1\} > P\{LAST:0\} \\ 0 & otherwise \end{cases}$$

Then, just prior to coding a coefficient (LAST, RUN, LEVEL) we apply the operation LAST = LAST⊕*nEOB*. It shall be noted that this "negative EOB" can be easily

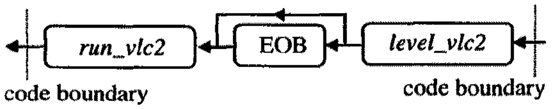
applied to VLC tables with EOB code such as the proposed method as "LAST:1" is identical to "end of block".
We also apply another technique to save the number of bits. In MPEG the first code of an inter-coded block cannot be an EOB. So we can assign the code of the EOB (=01, when EOB_length=2) and a sign bit to represent the pairs (RUN=0, LEVEL=±1) when they are the only pair (i.e. coefficient) in a block. The code also signals the end of the block. This way we could save about 0.8% number of bits in low bit-rate applications, when the pairs (RUN=0, LEVEL=±1) are often the only run-level pair in a block.

3.3. Extension to reversible VLC for error resilient coding

Fig. 2 depicts the structure of reversible (i.e., decodable both in forward and backward directions) UVLC in error resilient mode. Table 4 and 5 show the UVLC tables for RUN and LEVEL, respectively. With this scheme, a part of bitstream which can't be decoded in the forward direction due to errors can be recovered by backward decoding. The backward decoding begins with next immediate resynchronization word (e.g., start code) sought after encountering errors, and then decode bitstream from the resynchronization word up to the error portion. This way reversible VLC helps improve error resiliency for image/video compression algorithms employed in error prone environments like wireless communication.



(a) Forward coding process



(b) Backward coding process

Figure 2. Error resilient mode with two-way decoding

As observed in the two tables, the same codeword "000" is assigned as EOB. The coarse code appears in even-indexed bits such as "1", "00", "010", while the additional code is represented as a sequence of odd-indexed bits. In other words, the UVLC code words are constructed by interleaving additional code into coarse code. This enables the decoding in both directions as illustrated in Fig. 2.

Table 4. RUN UVLC table in error resilient mode

run_vlc2	size	value of run
1	1	0
$0x_0 0$	3	if "x_0"='0', then EOB else, run = 1
$0x_1 1x_0 0$	5	"$x_1 x_0$"+2 (2:5)
$0x_2 1x_1 1x_0 0$	7	"$x_2 x_1 x_0$"+6 (6:13)
$0x_3 1x_2 1x_1 1x_0 0$	9	"$x_3 x_2 x_1 x_0$"+14 (14:29)
$0x_4 1x_3 1x_2 1x_1 1x_0 0$	11	"$x_4 x_3 x_2 x_1 x_0$"+30 (30:61)
$0x_5 1x_4 1x_3 1x_2 1x_1 1x_0 0$	13	"$x_5 x_4 x_3 x_2 x_1 x_0$"+62 (62:125)

Table 5. LEVEL UVLC table in error resilient mode

level_vlc2	size	absolute value of level
1s	2	1
000	3	EOB
010s	4	2
$0x_1 1x_0 0s$	6	"$x_1 x_0$"+3 (3:6)
$0x_2 1x_1 1x_0 0s$	8	"$x_2 x_1 x_0$"+7 (7:14)
$0x_3 1x_2 1x_1 1x_0 0s$	10	"$x_3 x_2 x_1 x_0$"+15 (15:30)
$0x_4 1x_3 1x_2 1x_1 1x_0 0s$	12	"$x_4 x_3 x_2 x_1 x_0$"+31 (31:62)
$0x_5 1x_4 1x_3 1x_2 1x_1 1x_0 0s$	14	"$x_5 x_4 x_3 x_2 x_1 x_0$"+63 (63:126)

3.4. Bitstream Syntax

The bitstream syntax at picture and DCT block layer are given in Table 6 and 7, respectively.

Table 6. Syntax at picture layer

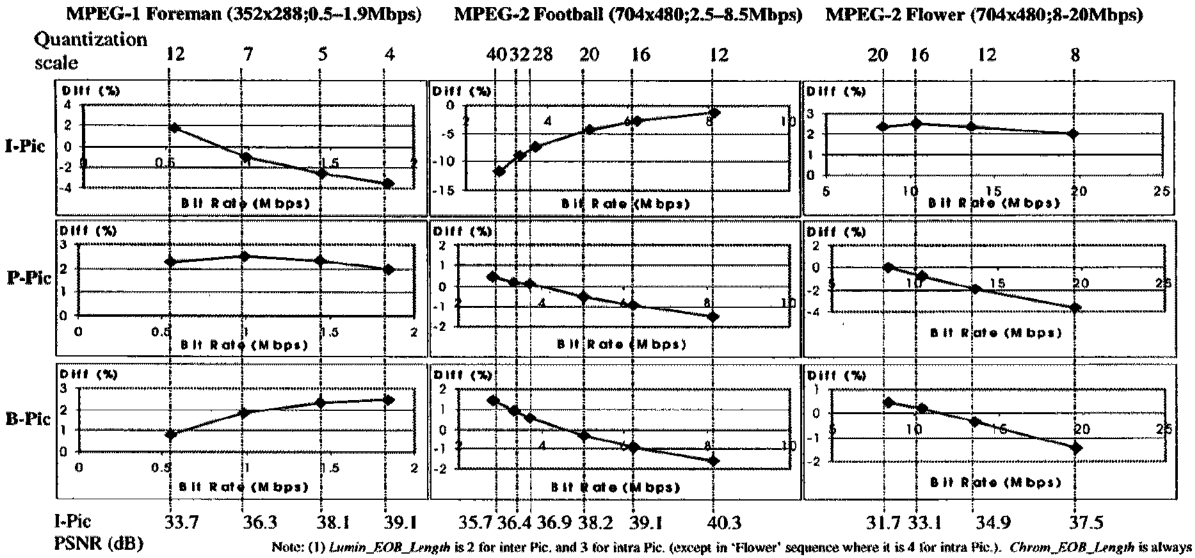
```
                    SYNTAX                      # BITS    MNEMONIC
   ...
   Lumin_EOB_length                              2
   Chrom_EOB_length                              2
    nEOB                                         1        uimsbf
   ...
Note: uimsbf – unsigned integer, most significant bit first
```

Table 7. Syntax at DCT block layer

```
                    SYNTAX                      # BITS    MNEMONIC
dct_code(){
   LAST=0
   if (!nEOB) {  /*nEOB==0*/
      while (LAST==0) {
         level_vlc                               2-15      vlclbf
         if (EOB)
            LAST=1
         else
            run_vlc                              1-12      vlclbf
      } /* while */
   } else {        /*nEOB==1*/
      while (LAST==0) {
         level_vlc                               2-15      vlclbf
         if (EOB)
            level_vlc                            2-15      vlclbf
         else
            LAST=1
         run_vlc                                 1-12      vlclbf
      } /* while */
   }              /* if */
}
Note: vlclbf – variable length code, left bit first
```

4. SIMULATION

In the simulation, we compared the bit consumed between the UVLC and the standard VLC employed in the MPEG and H.263. Figure 3 shows the differences in the bit consumed to encode DCT coefficients (except intra DC coefficients) using the UVLC and the MPEG VLC schemes. Negative differences mean UVLC consumed smaller number of bits than the MPEG VLC. As shown in the figure, the UVLC consumed no more than 3% number of bits when compared with MPEG VLC while simplified significantly the implementation of the VLC codec. In another simulation, we used UVLC to encode motion vectors (Table 8) and DCT coefficients (Table 2 for intra chrominance and Table 3 for intra luminance). The parameter *nEOB* was set adaptively according to the criteria described in the section 3.2. The quantization scale was fixed to 15 for intra frame and 20 for inter frame, which resulted in adequate visual qualities, i.e., 31dB -



Note: (1) *Lumin_EOB_Length* is 2 for inter Pic. and 3 for intra Pic. (except in 'Flower' sequence where it is 4 for intra Pic.). *Chrom_EOB_Length* is always 2.
(2) In the MPEG-1 foreman sequence, we assign the code of the EOB (=01) and a sign bit to code the pairs (RUN=0, LEVEL=±1) when they are the only run-level pair in an inter-coded block.

Figure 3. Comparing the bit consumed between the UVLC and the MPEG VLC

33dB. Table 9 summarizes the simulation results in terms of the number of bits consumed per frame. Note the value represents the average bits consumed per frame (each for intra- and inter-frame separately) over 10 seconds. It is observed that the proposed scheme can reduce the bits about 3% for intra frame and 2% for inter frame.

Table 8. Motion Vector (MV) UVLC table

mv_vlc coarse	 additional	code size	absolute value of dmv
1		1	0
01	s	3	1
001	x_0s	5	"x_0"+2 (2:3)
0001	x_1x_0s	7	"x_1x_0"+4 (4:7)
00001	$x_2x_1x_0s$	9	"$x_2x_1x_0$"+8 (8:15)
000001	$x_3x_2x_1x_0s$	11	"$x_3x_2x_1x_0$"+16 (16:31)
0000001	$x_4x_3x_2x_1x_0s$	13	"$x_4x_3x_2x_1x_0$"+32 (32:63)
's' denotes the sign of dmv; '0' for positive and '1' for negative			

Table 9. Simulation result (bits consumed per frame)

Test sequence	intra/ inter	VLC table			
		H.263	TBL-0	TBL-1	UVLC
Mother& Daughter (352x240)	intra frame	14,856	15,019 (+1.10%)	15,623 (+5.16%)	14,377 (-3.22 %)
	inter frame	94.4	105.2 (+11.4%)	N/A	92.8 (-1.69 %)
Foreman (352x288)	intra frame	30,347	29,973 (-1.23%)	29,660 (-2.26%)	29,292 (-3.48 %)
	inter frame	2199.4	2277.2 (+3.54%)	N/A	2148.6 (-2.31%)
Note	TBL-0: MPEG-2 DCT coefficients table zero TBL-1: MPEG-2 DCT coefficients table one				

5. CONCLUSIONS

The UVLC for DCT coefficient coding was presented. The features of the scheme are: 1) adaptable by parametric representation, 2) support of very low bitrate mode and 3) extensible to error resilient mode with bi-directionally decodable code. The simulation results show that the UVLC slightly outperforms several VLC tables of existing standards in terms of coding efficiency. Besides the proposed scheme can simplify the implementation and processing of VLC decoder. These features and advantages can be justified by the fact that the UVLC is being considered for H.26L [5], that is, a next generation video coding algorithm standard.

6. REFERENCES

[1] ITU-T Recommendation H.263, "Video coding for low bit rate communication," March 1996.

[2] ISO/IEC 13818-2, "Generic coding of moving pictures and associated audio information – Part 2: Video," November 1994.

[3] Yuji Itoh, "Bi-directional motion vector coding using universal VLC," *Signal Processing: Image Communication*, Vol. 14, pp. 541-557, May 1999.

[4] ISO/IEC 14496-2, "Information technology - Generic coding of audio-visual objects – Part 2: Visual," October 1998.

[5] G. Bjontegaard, "H.26L Test Model Long Term Number 1 (TML-1)," *ITU-T Q.15/16*, Doc. # Q15-H36, Sept. 1999.