Stockholm meeting

N 2

CCITT SGXV
Working Party XV/1
Experts Group for
ATM Video Coding

International Organization for Standardization
Organisation Internationale de Normalisation
ISO/IEC JTC1/SC2/WG11
Coding of Moving Pictures and
Associated Audio

Document AVC-292
July 1-10 1992

ISO-IEC/JTC1/SC2/WG11 N
MPEG 92/ 289
July , 1992

Title: Frequency Scanning & Entropy Coding using MUVLC

Source: G. SCHAMEL & H. LI / Heinrich-Hertz-Institut (HHI) Germany
O. PONCIN / RTT BELGACOM, Belgium
B. HAMMER / Siemens, Germany
on behalf of VADIS

Purpose: Proposal, results, syntax modification, C-programs

1 Introduction

One of the most important applications for scalability will be future digital broadcasting services. Especially in terrestrial and satellite channels the requirement of graceful degradation arises because maximum service quality cannot be guaranteed under all circumstances (e.g. at borders of service area or in buildings). Furthermore, compatibility between TV and HDTV receivers will play an important role, because portable (plug free) receivers must be able to decode parts of the HDTV bit stream. These two requirements should be kept in mind during the development of the MPEG 2 standard and therefore scalability is of utmost importance.

Due to this reason several experiments are currently carried out within the scalability ad-hoc group. The two parts of the core experiment are called BROADCAST WITH HETEROGENEOUS RECEIVERS and MULTI-CHANNEL BROADCASTING. The syntax extension described in TM1 allows the introduction of a scaled bitstream within the test model. The main disadvantage, however, is the block scanning technique of the DCT coefficients. As we know, combining both scalability in terms of different DCT sizes , i.e. 2x2, 4x4 and 8x8 and block scanning reduces both coding efficiency due to the modified scan paths (see documents distributed in the ad-hoc group) and flexibility.

In this contribution, the well-known UVLC technique is mentioned, an improved version MUVLC concerning coding efficiency is described, a syntax proposal to be added into TM1 is given , C-programs for bitstream encoding and decoding a slice of macro-blocks and results on coding gain are also included.

2 Frequency Scanning

The technique of frequency scanning (mostly called UVLC="universal variable length coding") has been briefly described in doc. MPEG 92/28. Frequency scanning performs some kind of re-ordering of all DCT coefficients of a slice of 44 macro-blocks. Thus, we get an array of all DC coefficients within the slice, a second array of all AC1 coefficients, a third array of all AC2 coeffients and so on. Of course, we have to decide on some scanning to group the different arrays. However, this is of minor importance concerning coding efficiency. All arrays of the lowest 2x2 DCT coefficients belong to the base layer. If we would assign variable and run length codes within the arrays, we would not get any increase of coding efficiency. However, if we look to the binary represenation of each coefficient, we can observe long runs of zeros within each array and bit-plane under consideration, i.e. bits of the same level of all coefficients within an array are considered. A 3D space of zeros and ones results from this technique (see Figure 1). Now, the technique of coding the address of the most significant non-zero bit within a row of bits is applied. The algorithm described in doc. CMTT 2/131 (March 1990) could be used. However, we will describe a modified UVLC algorithm, in the following refered to as MUVLC which gives an improvement in coding efficiency.

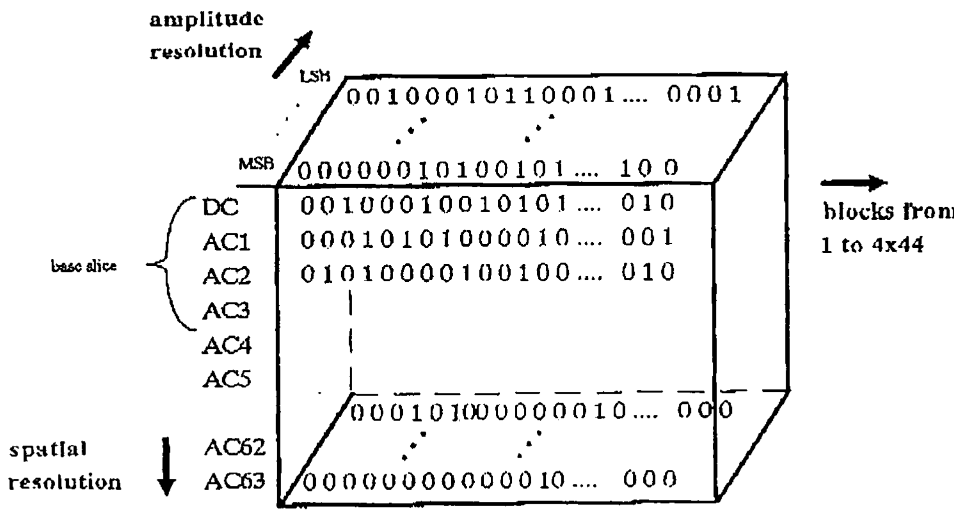

Figure 1: Frequency scanning technique applied to all DCT blocks within a slice of 44 macro-blocks; layer 1 to 4 (DC to AC3) of the slice can be grouped to base layer, layer 4 to 15 to slave layer 1 and layer 16 to 63 to slave layer 2.

2.1 Principle

The efficiency of UVLC and also MUVLC is based on some elementary observations, which may be considered valid in a first approximation, namely:

- the probability distribution function of DCT coefficients is uncorrelated. Thus it is possible to use adaptive codes, which calculate the coded data after some on-line estimation of the statistics of a sequence of uncorrelated coefficients having the same probability distribution.

- when the amplitude of a coefficient is analysed from the MSB to the LSB, the probability for the next bit being zero remains very high as long as no bit equal to one is met. However, when the first non-zero bit (MSNZB: most significant non zero bit) has been met, the probabilities of following bits to be zero or one are roughly equal. Therefore, the entropy of these bits is close to one and very little is gained by coding them.

Both, UVLC and MUVLC use these properties by coding the positions of the MSNZB bits and by transmitting the remaining bits uncoded. Some further refinements have been added to increase the coding efficiency. The VLC words are generated in two steps:

- storing all the blocks of transform coefficients of one slice (4 x 44 blocks, also non-coded blocks concerning the coded block pattern!) into a tridimensional table (see Figure 1).

- encoding sequences of DCT coefficients of the same order taken in this table by a run-length coding of their MSNZB and by sending the less significant bits uncoded. Sequences of coefficients of the same order are processed at the bit-level. Such sequences correspond to horizontal slices in the tridimensional table illustred on Figure 1. Figure 2 illustrates such a slice considered at the bit level as well as the analysis process. The coefficients of a given order are encoded by a run-length coding of their lines of bits from the MSB to the LSB (the sign bit is placed after the LSB): when a non-zero bit is encountered, the other less significant bits are sent uncoded and the whole non-zero coefficient is removed from the stack. As the encoded coefficients are removed from the table, the lines to encode become progressively shorter when the process runs from the MSB line to the LSB line.

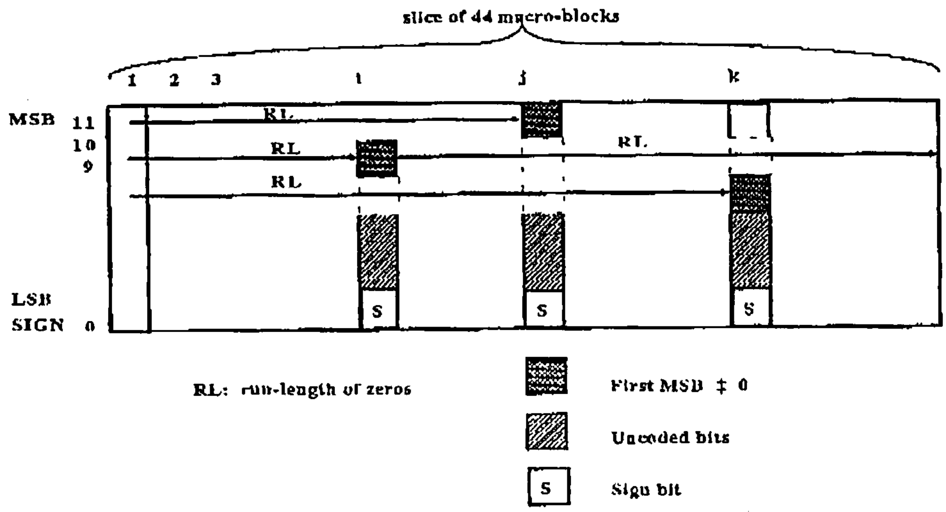

Figure 2: Principle of MUVLC and UVLC coding of a stripe of 44 macro-blocks.

2.2 Coding of the MSNZB

The adaptive truncated run-length coding (ATRL) proposed by Tanaka and Leon-Garcia [see reference 2] has been used for encoding the positions of the MSNZB. This scheme is robust with respect to the statistical fluctuations of the pictures and allows a very simple implementation. For a DCT coefficient of a given order, the ATRL code is applied succesively to the lines of the table shown in Figure 2, from the MSB to the LSB. Each time a MSNZB is met, the corresponding coefficient is removed from the table, which makes the latter progressively narrower.

The zero run length encoding is limited to a length of $M = 2^m$ (truncation). The truncated run-length code encodes two types of patterns:

- runs of length 0 to M - 1 consecutive zeros terminated by a one;
- M consecutive zeros

The M consecutive zeros are coded by one bit set to 0. The run length terminated by a one are coded by one bit set to 1 followed by the position of the one coded on $m = \log_2(M)$ bits (see Figure 3). The TRL is made adaptive by changing the value of m in function of the probability of a bit to be one and of the length L of the line to encode. Here, the main difference between UVLC and MUVLC appears. The operation of MUVLC proposed by HHI (VADIS92/TD7) is described in the following:

Assuming a string of zero and one bits we can count the length of all zero runs denoted with d_i and come out with the following formula which describes the necessary bits for transmission still depending on a window size of $M = 2^m$:

SOURCE:	CODE-WORD for Transmission
00000000	0
1	1000
01	1001
001	1010
0001	1011
00001	1100
000001	1101
0000001	1110
00000001	1111

Figure 3: Truncated run-length for a maximum window-size M=8 , m=3. Window sizes of 2, 4 8, 16, 32, 64 and 128 are possible. Thus, m=3 bit are sufficient for the decoder to get this information (m=prefix-line PL)

$$[\frac{d_1}{2^m}] + 1 + m + [\frac{d_2}{2^m}] + 1 + m + [\frac{d_3}{2^m}] + 1 + m + \ldots + [\frac{d_{k-1}}{2^m}] + 1 + m$$

with [ ] denotes the integer part of the number. This formula can be rewritten into

$$f(m) = \sum_{i=1}^{k} [\frac{d_i}{2^m}] + (k-1) + (k-1)\,m \qquad \text{if} \quad \frac{d_k}{2^m} \quad \text{is integer}$$

$$f(m) = \sum_{i=1}^{k} [\frac{d_i}{2^m}] + (k-1) + (k-1)\,m + 1 \qquad \text{if} \quad \frac{d_k}{2^m} \quad \text{is not integer}$$

$$f(m) = \sum_{i=1}^{k} [\frac{d_i}{2^m}] + k + k\,m \qquad \text{if last bit in string is a one bit}$$

with f (m) the number of bits necessary to encode the full line. Each integer operation $[\frac{d_i}{2^m}]$ gives the number of zero bits for transmission until the one bit is reached. This is assigned by the next bit set to one followed by m bits to code the address of the one bit within the window with its 2^m possible bit positions. The last run-length within the string is different because it may contain any number of zeros less than 2^m. In order to obtain the smallest f(m), the optimal window size must be searched. The search algorithm is very simple because there are only 8 possibilities of window size for a string of 176 bits which have to be tested. Be aware, that the "older" UVLC only counts the number of one bits and calculates from that the optimum window. It does not take into account the distribution of the one bits, which is, however, considered in the MUVLC scheme and which results in an improved performance. Two examples of strings of bits which are coded with MUVLC are given in Figure 4.

As the standard deviation of DCT coefficients decreases with increasing order, it is likely that several lines on the top of the table of zero and one bits shown in Figure 1 will never contain MSNZB. The efficiency of the coding scheme can further be increased by using a prefix saying how many lines do not contain any MSNZB. Such an information can be provided for classes of coefficients rather than for each coefficient individually.

Example:

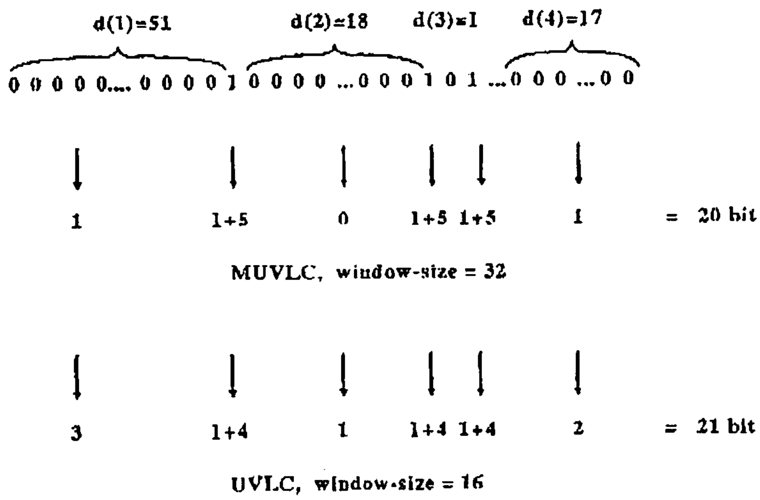

Figure 4: Example of MUVLC and UVLC coding with different window sizes. MUVLC automatically selects the optimum window size(which is 32 in this example; UVLC would select 16 !)

We propose to use separate classes for all 64 coefficients. To calculate the class-prefix PC, the absolute values of the corresponding DCT coefficients can be analysed and the maximum value within the class is used to define the 3 bit class prefix.

The 3 bit class-prefix says at which line of bits in Figure 1 the coding process starts for all coefficients of this class.

The decoding algorithm consist in simply writing zeros sequentially as the run-lengths are arriving. The line-prefix indicates the number of zeros which the RL's are truncated on. In the following, m is refered to as prefix-line PL which is transmitted as 3 bit value at the beginning of each line.

The 3 bit line-prefix PL is transmitted to the decoder to adapt to the window-size selected in the encoder.

In summary, the following syntax for a slice of DCT blocks results:

Begin of slice --> firstly, necessary control parameters for all macro-blocks followed by

PC PL RL NCB PL RL NCB PL NCB RL NCB ...

with

- PC: class prefix
- PL: line prefix
- RL: run-length
- NCB: non coded bits

This is described in more detail in Appendix A.

2.3 Luminance and chrominance signals

A macro-block which consist of four luminance L and two chrominance blocks U and V (in the 4:2:0 case !) is re-ordered for entropy coding with MUVLC. DCT coefficients of U and V are transmitted after the corresponding L coefficients, i.e. 176 L coefficients of the same spatial frequency are followed by 2 x 44 U and V coefficients, respectively. The four L coefficients are zig-zag scanned within the macro-block. If all coefficients of a block are zero (marked by the coded block pattern), this block is not removed as in TM1!!

3. Syntax Extension

Some syntax modifications have to be introduced into TM1. Frequency scanning changes the transmission order of the DCT coefficients. Firstly, all control information of macro-blocks (nearly most of the macro-block syntax) has to be transmitted for all macro-blocks of the slice. This information is part of the ***stripe-control layer***. Then, MUVLC starts transmitting all DC, all AC1, all AC2, ... coefficients. This is called the ***stripe-data layer***. Thus, the block-layer is not necessary at all. If necessary, the stripe-data layer could be split into the ***base-data layer*** and some ***slave-data layers*** similar to the syntax extension proposed by IBM in TM1.

MQUANT is calculated taking the activity measure into account. Changing QP on the macro-block level is prohibited in this version. Thus, a new buffer controlled QP can only be calculated at the beginning of the next slice

4. Results from Frequency Scanning

In the following, simulation results on MUVLC (and also UVLC) carried out by different companies are reported.

- **HHI-results:** We simulated TM1 (without FAMC, only Dual Field Prediction), fixed the quantizer feedback from the buffer to QP=4, 8 and 12. Then, we counted the number of bits for TM1 entropy coding (N=10, M=2) and for MUVLC inserted into TM1. We still used CBP, i.e. blocks which have coefficients with amplitude zero are not removed. Due to the very short slice length of the chrominance, MUVLC is less efficient as expected. Therefore we put both chrominance signals U and V together (after 44 U macroblocks 44 V macroblocks follow) applied MUVLC and achieved some improved results.

Bit Statistics		MOB&CAL		FLOWER		TAB-TEN		Gain in
fixed QP = 4		TM1	MUVL	TM1	MUVL	TM1	MUVL	[%]
I	Y	1203028	1037603	894795	774209	1164363	986326	14.23
	U	121528	232500	97216	178481	55188	118885	-0.79
	V	109025		78550		64175		
B	Y	444143	473793	346657	370319	416875	393410	-2.41
	U	36418	84374	34738	84180	14409	44769	-26.28
	V	28608		30617		12462		
P	Y	742432	738482	567859	558107	534601	503784	2.47
	U	60997	136912	54448	115483	19434	56444	-20.5
	V	50289		42059		18052		

Bit Statistics fixed QP = 8		MOB&CAL TM1	MOB&CAL MUVL	FLOWER TM1	FLOWER MUVL	TAB-TEN TM1	TAB-TEN MUVL	Gain in [%]
I	Y	698131	661168	541527	494465	657204	584704	8.25
	U	59297	127095	47237	91500	22560	63095	-8.53
	V	56164		35430		36986		
B	Y	148450	174860	123025	140276	126480	115937	-7.68
	U	6253	25830	6819	25289	1485	19147	-72.66
	V	4707		4634		2542		
P	Y	303889	332708	251242	259943	190053	163585	-1.46
	U	14442	45896	15539	40053	2538	22273	-46.26
	V	13235		8210		4196		

Bit Statistics fixed QP = 12		MOB&CAL TM1	MOB&CAL MUVL	FLOWER TM1	FLOWER MUVL	TAB-TEN TM1	TAB-TEN MUVL	Gain in [%]
I	Y	495081	487756	395590	369026	453335	394323	6.91
	U	41180	95127	32179	66312	15611	48757	-12.9
	V	40380		24278		29327		
B	Y	80252	100481	64506	83054	54144	58989	-17.98
	U	2201	18217	2643	18144	448	15756	-81.71
	V	1848		1351		1041		
P	Y	168097	193306	146253	157284	91152	82942	-6.46
	U	6410	28823	7072	25755	1127	18690	-64.37
	V	6157		2921		2415		

The results have been bitstream verified. It can be concluded that CBP should be exploited (see SIEMENS results for B and P) and that the overhead information due to PC and PL is too high for the chrominance (e.g. PCs need 7 kbit/fram, which is in some cases more than both U and V in TM1) in the case of high QP, i.e. low bit rate. Furthermore, there is always a large gain in intra and some loss in B and P pictures. The inter-mode seems to be very critical because runs of zeros are interrupted by intra blocks. However, there are some ideas to overcome these difficulties.

- **SIEMENS-results:**

  - TM1 was simulated without the prediction modes enclosed in the annex of TM1
  - macroblock layer: all references to block information are deleted
  - after transmitting the macroblock information of one macroblock slice, the coefficient information of all macroblocks, which are indicated as coded by CBP are arranged and coded by UVLC [1]
  - no field prediction is used
  - no field DCT coding is used
  - Y, U and V components are encoded separately
  - adaptive quantization: $mquant_j = Q_j$ if buffer control is used, $mquant_j$ is updated for each slice.
  - the results have been bitstream verified

Bit Statistics fixed QP = 4		MOB&CAL TM1	MOB&CAL UVL	Gain in [%]
I	Y	1246726	1121529	12.79
	U			-1.84
	V			-5.04
B	Y	568126	537166	6.43
	U			3.00
	V			-1.93
P	Y	754568	710594	7.20
	U			0.34
	V			-3.65

Real bitrate: 18.167 Mbit/s

Bit Statistics		MOB&CAL		Gain in [%]
4.430 Mbit/s		TM1	UVL	
I	Y	529227	514934	6.23
	U			-13.85
	V			-17.56
B	Y	82308	87047	18.02
	U			-3284
	V			-2063
P	Y	199040	192351	11.11
	U			-62.87
	V			-72.30

The results have been bitstream verified. The rather high loss in the chrominance is only related to those blocks of coefficients which have to be coded, i.e. which are not removed due to CBP.

- FI/DBP results
  - these results, which confirm the results described above, will be presented in a separate paper during the Rio-MPEG meeting
- RTT BELGACOM results
  - these results have already been presented in a separate paper during a VADIS meeting [COST 211Ter-VADIS Algorithm Group 2, Madrid 3-5 April 1991, UCL (Belgium)]

5 Conclusions

The MUVLC (and also the UVLC) technique gives some improvement in coding efficiency. However, the simulations showed some strong relation both between the length of the stripe and coding gain and between target bit rate and coding gain. The following observations have been made:

- The longer the stripe, the higher the coding gain. As a consequence, the coding gain for chrominance signals is lower as the chrominance stripes are shorter than the luminance stripes.
- The higher the target bit rate, the larger the coding gain.

These observations which have to be confirmed by further tests make this entropy coding scheme well suited for broadcasting applications of TV and HDTVwith its increased stripe length.

Therefore, the advantages of MUVLC in terms of flexibility, adaptivity and coding gain are strong enough to demand for inclusion of this technique into the test model.

6 References

[1] B. Macq: "An Universal Entropy Coder for Transform or Hybrid Coding", Picture Coding Symposium, session 12, March 26-28 1990, Cambridge, Mass., USA.
[2] H. Tanaka and A. Leon-Garcia: "Efficient Run-Length Encoding", IEEE Transactions on Information Theory, vol. IT-28, no.6, pp. 880-890, November 1982.
[3] CCIR Study Group - Document CMTT-2/31, March 1990 - Belgium RTT
[4] G. Schamel, "Contribution to the Discussion of new Entropy Coding Techniques", VADIS92/TD7, Helsinki 1992.
[5] Th. Selinger/HHI, "Implementation Study of a MUVLD", contribution to MPEG, July 1992
[6] A. Knoll/FI-DBP Telekom, "Experiments with UVLC-coding within MPEG2-TM1.