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

Document AVC-30
910507

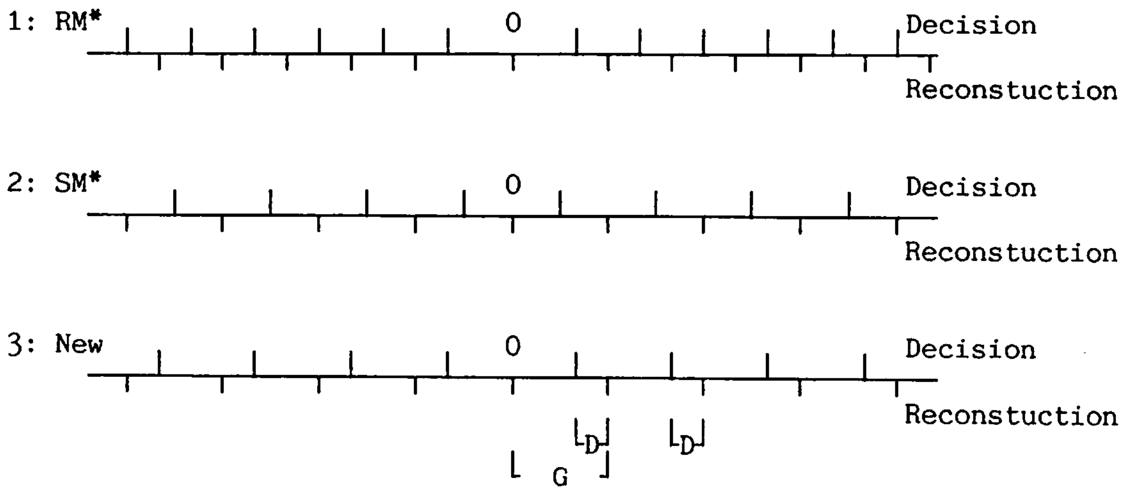
SOURCE : Norway
TITLE : Improvements of hybrid DCT coding.
PURPOSE: Information.

1. Introduction.

In this document some ideas for improving the coding performance of hybrid DCT coding schemes are described. The gain in efficiency is demonstrated on the sequencies TABLETENNIS, FLOWERGARDEN and MOBCAL. The document is accompanied by tape demonstrations.

2. Optimized quantizer for transform coefficientes.

A quantizer is characterized by its decision- and reconstruction levels. To optimize coding performance, it is important that the levels are correctly spaced. Below are shown the positions of the different levels used in H.261 RM*, MPEG1 SM* and the proposed quantizer.



Decision levels will not have to be standardized. In that respect, the main difference is between 1: and 2:/3:.

It has been noticed that for Laplacian distribution of transform coefficients, it is optimal to have linear reconstruction levels. It is also optimal to have equally spaced decision levels except around zero. The "nonlinearity" around zero may be described by D in the figure. D should be chosen so that:

D → 0 when the distribution is peaked around zero.
D → G/2 for very wide distributions (many large coefficients).

Some further comments on how to optimize D is given in Section 4. For the simulations presented here, it was chosen to use:

D = G/3 || D = G/4 utilisé par UCL pour simplicité

Simulations were made for the sequences TABLETENNIS, FLOWERGARDEN and MOBCAL at around 4 Mb/s. The stepsizes were adjusted so that the SNR values were kept constant. The bitsaving was (5-12) % using quantizer 3: instead of 2:.

3. Calculating prediction.

For high quality image coding the amount of filtering of the prediction is of great importance. The problem is often too strong filtering. This is particularly the case in panning situations. In the sketch below, capital letters indicate integer pixel positions. Lower case letters indicate fractional pixel positions. In MPEG 1 the predictions are:

```
a = (A+B+E+F)/4
b = (B+F)/2
c = (C+B+G+F)/4
d = (E+F)/2
e = (G+F)/2
F = F
f = (I+J+E+F)/4
g = (J+F)/2
h = (J+K+G+F)/4
```

```
          A     B     C

             a  b  c

    D     E  d  F  e  G     H

             f  g  h

          I     J     K
```

To reduce the filtering in the prediction, I have tried two different versions of filtering:

Version I:

Only the differences from above are given.

```
d = (9*(E+F)-D-G)/16
e = (9*(F+G)-E-H)/16
```

"Différence entre (1,1) et (-1 9 9 -1) est très importante. Plus long n'apporte pas grand chose" ([illegible])

```
          A     B     C

             a  b  c

    D     E  d  F  e  G     H

             f  g  h

          I     J     K
```

Version II:

This version represents both lower filtering and different positions of the vertical fractional positions as indicated in the sketch. The total number of positions - or average density of positions - are the same as above. The coding of motion vectors may therefore be very similar just by reidentifying position identifications.

```
a = (2*B+F)/3
b = (2*F+B)/3
d = (9*(E+F)-D-G)/16
F = F
e = (9*(F+G)-E-H)/16
f = (2*F+J)/3
g = (2*J+F)/3
```

```
        A   B   C
            a

            b
    D   E d F e G   H
            f

            g
        I   J   K
```

Bitsavings (in %) with the two versions:

	VERSION I	VERSION II
TABLE TENNIS	0	0 + 2
FLOWER GARDEN	0	0 + 4
MOBCAL	14	14 + 1

4. Use of multiple two-dimensional VLCs for coefficient coding.

Two-dimensional VLC for coding of transform coefficients was introduced in the work towards H.261. It proved to be a very efficient way of coding. However, the efficiency of the VLC depends on the statistics of the data to be coded. This is well known from coding theory.

The VLC defined in H.261 is optimized for luminance coding. An example of statistical mismatch is found when using this VLC for coding of chrominance coefficients. Since much less coefficients are coded for chrominance, a VLC with fewer bits for EOB would be more optimal.

The same consideration applies when coding with coarse and fine quantizer (low and high bitrate).

The most critical part of a VLC is that the codeword for the most probable outcome is well adjusted to the statistics. The most probable outcome is usually EOB. It is therefore important always to assign the correct number of bits to EOB.

4.1 Method of multiple VLCs. (Self- adaptive)

Four differnt two-dimensional VLC tables are defined. They use 1-4 bits for EOB (see below).

EOB = 1 bit	EOB = 2 bit	EOB = 3 bit	EOB = 4 bit
4 6 6 7 7 8 ..	3 6 7 8 9 10 ..	3 5 6 7 8 9..	3 4 5 5 6
5 9 10 11 12 13 ..	4 8 9 10 11 12 ..	4 7 8 9 10 11..	5 7 8 9..
6 10 11 12 13 ..	5 9 10 11 12 ..	5 8 9 10 11..	6 8 9..
6 11 12 13 ..	6 10 11 12 ..	6 9 10 11..	6 9..
7 12 11 .	6 11 12 .	6 10 11 .	7
7 13 .	7 12 .	6 11 .	7
8 .	7 .	7 .	8
.	7	7	8
	7	7	.
	8	7	
	.	8	
		.	

For every coefficient to be coded, one of the four VLCs is chosen. The decision is based on the following parameters:

NUM_COF Average no. of coefficients transmitted pr. block over the last 4 macroblocks.

NUM = $\log_2$(NUM_COF) (truncated to integer)

LEVEL Level of the previously coded coefficient.

RUN Number of zeros before the previously coded coefficient.

RUN3 = MIN(1,RUN/3) (integer division).

The VLC NUMBER is then chosen as follows:

For the first coefficient in a block:

VLC_NO = MIN(4,MAX(1,NUM))

For the following coefficients:

VLC_NO = MIN(4,MAX(1,(4*LEVEL + NUM - 2*RUN3)/2)) (integer division)

In this way the coding is very adaptive to:

- Source data.
- Quantizer/bitrate.
- Luminance/Chrominance coding.

The above scheme has been tested on the sequences: TABLETENNIS - FLOWERGARDEN - MOBCAL at 4 Mb/s. The saving in bits was (8-10)% compared to using the H.261/MPEG1 VLC.

sans utiliser CBP car 8SB/MB(l 2:2

It may be noted that VLC_NO (or a similar parameter) may be used to decide the parameter D in the quantizer in Section 2. This will further optimize the quantizer.

5. Prediction from two frames as alternative to interpolative coding.

One of the major differences between H.261 and MPEG1 coding is the inclusion of interpolative coding in MPEG1. This gives a cosiderable gain - particularly in scenes with panning. The gain is particularly important in MPEG2 due to the frequent use of panning in broadcast sequences.

It has been unclear - at least to me - why interpolative coding gives such a gain. After some investigation it seems that the main reason is this:

- Predictions are made from INTRA and PREDICTED images only.
- PREDICTED and INTRA pictures are coded with lower quantizer and thereby better quality.
- With pure panning - like in MOBCAL - "correlation distance" in time is of little importance since the previous frame and the Nth previous frame contains the same information - only linearly translated.
- The conclusion from the above points is that the prediction from an "old" PREDICTED or INTRA frame is better than prediction from the previous frame for panning situations.
- In addition to the point above, prediction from two possible frames increases the number of motion vectors by a factor of two. This may be seen as finer resolution of motion vectors that gives a better match to true motion.

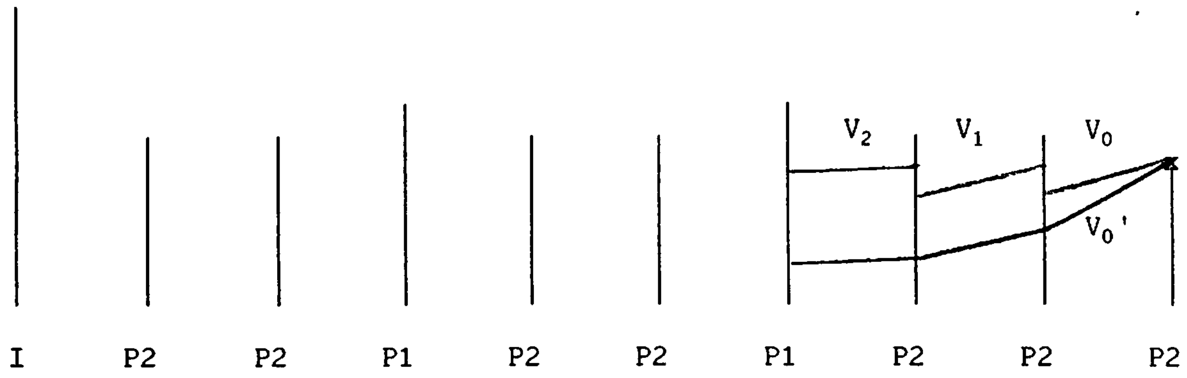
The better prediction gives the better picture quality.

5.1 Dual frame prediction.

The improved predictor is the result of pictures being coded with different quality. This may be obtained also without using interpolative coding.

In the following I describe a method for obtaining the same advantages concerning prediction in panning situations as interpolative coding. Only frame prediction is considered here. The same method could be used for field prediction.

- Three types of frames are used: INTRA(I), PREDICTED(P1,P2). see figure below.
- Only forward prediction is used.
- The predicted frames are coded with different quantizers. P1 is coded with the same quantizer as intra -I. P2 is coded with larger quantizer.
- For prediction either of two frames may be used:
  - * The previous frame. This is called PREV. (last coded frame)
  - * The last frame of type I or P1. This frame is called PREVQ.



The figure indicates how motion vectors are obtained. Assume that we are to find the prediction at point x in the figure. Assume also that the vectors for the same positions at the two previous frames are stored - V_1, V_2. We then find:

- V_0 to give the best prediction using PREV.
- V_0' so that (V_2+V_1+V_0') gives the best prediction using PREVQ. The search for V_0' is done only in the vicinity of V0 (+/- 1 pixel in each direction).
- The sum of absolute differences are used to decide whether to use V_0 or (V_2+V_1+V_0') as predictor.
- V_0 or V_0' - plus one bit to tell which - is transmitted.

Advatages with this method:

CODING EFFICIENCY.

I have not yet implemented the interpolative method and can therefore not compare directly. However, comparing with reported SNR values and comparing "visual impressions" indicates that the present method compares well.

I have done comparisons with the present method and one based on prediction from the previous frame only(using V0 above). The simulations were made with TABLETENNIS, FLOWERGARDEN and MOBCAL. The savings in bits were:

TENNIS:	4%
FLOWERGARDEN:	6%
MOBCAL:	18%

CODING DELAY.

The coding delay is reduced compared with interpolative coding. With 25 frames/s the difference in delay is (M-1)*40 ms. With M=3 this means 80 ms. This reduced delay is of great importance for conversational services.

FLEXIBILITY.

The present method gives room for large flexibility:

- Since there is always a choice between PREV and PREVQ for prediction the encoder may always adapt to the extremes: "panning" and "violent motion".

- The difference in quantizer may be used or not according to the picture material. No variation in quantizer would be coupled to only PREV prediction.

- The frequency of pictures coded with improved quality (P1) may easily be varied. It also need not to be constant (as in the figure above).

6. Simulation results.

The coding effects described in the present paper are included in the coding model that was used to produce pictures with 4 and 9 Mb/s. The simulation conditions are summarized below.

- Picture format is according to CCIR 601, 50 Hz version. Macroblocks of 16*16 luminance pixels are used. Coding of difference signal is done with 8*8 blocks for Y,U and V.

- Only frame prediction and coding is used.

- Motion search is made in two steps:
  1. search over integer vector positions.
  2. for the fractional part the method of Chapter 3 is used.

- Rate control. No rate control is included. Fixed quantizer is used for INTRA and P1. The quantizer for P2 is half as big. The quantizer is adjusted to give the wanted average bitrate.

- Scanning: zig-zag scanning used.

- Quantization matrix similar to MPEG1 (SM3) was used.

- Quantization of transform coefficients: As in Chapter 2.

- VLCs for coefficient coding: As in Chapter 4.

- INTRA coding was done every 10 frames.

- Sequences for simulations: TENNIS, MOBCAL, FLOWERGARDEN. Fifty frames of each sequence are coded.

Summary of simulation results:

		TENNIS		MOBCAL		FLOWERGARDEN	
	Total bitrate(Mb/s)	4.0	9.0	4.0	9.0	4.0	9.0
1	Bitrate for I (Mb/s)	1.30	2.4	1.6	2.8	1.4	2.5
1	Bitrate for P1 (Mb/s)	1.35	3.1	1.2	2.7	1.2	2.7
2	Bitrate for P2 (Mb/s)	1.35	3.5	1.2	3.5	1.4	3.8
N > ω	SNR-Y(db)	32.48	35.70	29.00	33.50	29.13	34.10

7. Comparisons between coding interlaced and progressive images.

Interlacing causes problems for efficient image compression for several reasons.

For prediction:

- When prediction is made from fields with same parity, the correlation distance in time is two times larger than the distance between adjacent fields.

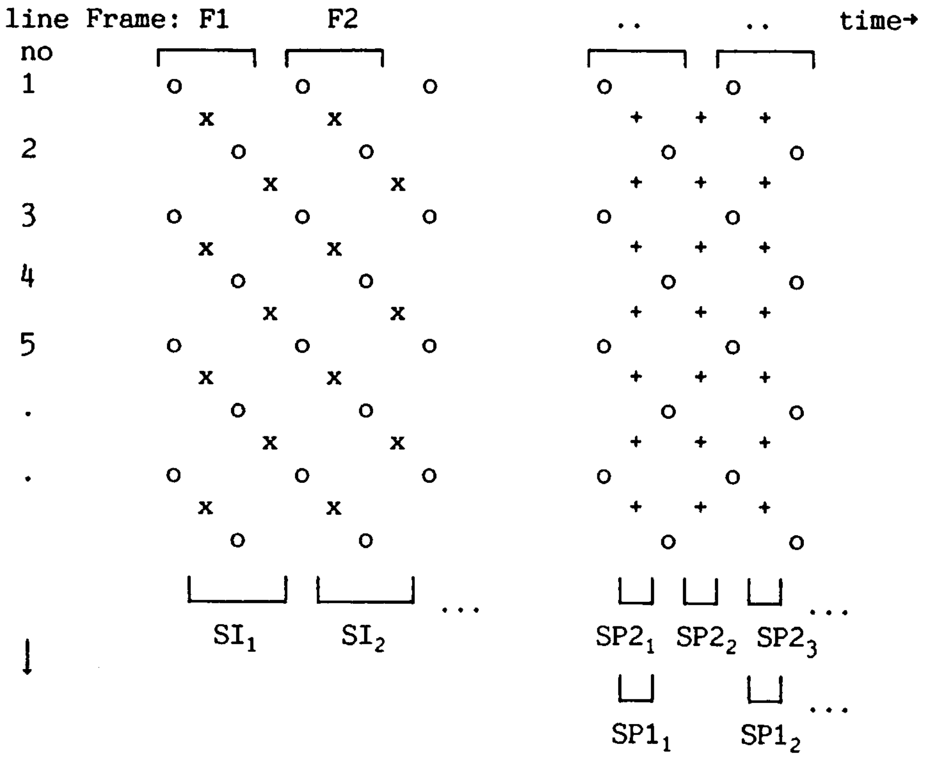
- The pixel accuracy of the vertical motion vectors is reduced by a factor two.

For coding of difference signal:

- The signal to be coded is a three dimensional signal, whereas we try to use two dimensional methods (e.g. DCT). This gives inefficient coding for scenes with motion.

I have made simulations to compare coding of interlaced and progressive sequences. For this purpose I have used the same testsequences as elsewhere in this paper. The figure below shows how the original interlaced signal is used to produce three new sequences:

SI	- New interlaced sequence.
SP1	- Progressive sequence with 25 frames/sec.
SP2	- Progressive sequence with 50 frames/sec.



Production of new sequences. o: original interlaced picture.
x: interlaced picture SI.
+: progressive picture used for SP1 and SP2.

To produce the new points, the interpolation filter: (-1,9,9,-1)/16 was used in vertical direction. Notice that SP2 has twice as many points as SI and SP1. In the simulations this is described by SP2 having twice as many pictures/sec as SI and SP1.

In connection with the two types of sequences, I have also made two types of coding models:

PI:
This is the "interlaced" model and is is used for simulations of the interlaced test sequences in this document. Coding is done framebased. Prediction of a pixel belonging to one field is based on previous pixels belonging to field with same parity.

PP:
This is the "progressive" version. It is derived from PI. The main differences:

- Frame to frame prediction makes the prediction distance in time half as long as for PI.

- The vertical spacing between motion vectors is halved.

Simulations were performed on the sequences SI and SP. Quantizers were adjusted so that SNR was constant for the different coding versions of a sequence. The results are summarized in the table below.

Sequence	Coding	SNR(db)	Bit(Mb/s)	Bitrate rel.to SI,PI
TENNIS SI	PI	33.6	3.30	0
" SP1	PP	"	4.42	31
FLOWER SI	PI	30.0	3.30	0
" SP1	PI	"	5.67	72
" SP1	PP	"	4.28	30
" SP2	PP	"	2.52	-24
MOBCAL SI	PI	29.7	3.78	0
" SP1	PI	"	5.92	57
" SP1	PP	"	4.96	31
" SP2	PP	"	2.88	-24

The following conclusions may be drawn:

- The progressive picture SP2 contains twice as many pixels as the interlaced picture, but the bitrate is only increased by about 30% if the coding method is adjusted to the picture format.

- Comparisons of coding the sequences SI and SP1 - both with 25 frames/sec - shows that the progressive sequence needs about 25 % less bits than the interlaced sequence.

- It takes about (20-30)% more bits to code the progressive format using PI compared to using PP. This is of importance for TV application where a large part of the picture material stems from film cameras and therefore is progressive.

All the results from the simulations above should give clear evidence that progressive picture formats are much to prefer for image coding.