CCITT SGXV **Doc. #420**
Working Party XV/4
Specialists Group on Coding for Visual Telephony

Source : NL,BTRL,F,I,FRG,S,N
Title : Modifications to doc # 396 Description of Reference model 6

1 Introduction

Document 396 gives a comprehensive description of Reference Model 6. Implementation of the different modifications revealed some ambiguous parts. Therfore a modified version is forwarded[1] which would replace doc. 396 which is already distributed. Some major modifications which the meeting should be agreed upon are discussed in the next sections.

2 VLC for Differential motion vector.

The Variable Length Code for the differntial motion vector depicted in table 8 of doc 396 could be shortened and so more efficient. The old VLC is depicted in table 1 and the proposed VLC in table 2.

3 Buffer Control

On page 26 of doc 396 the formula does not hold for the various temporal frequencies.

4 First picture and scene Change

In this section an attempt is made to tackle this item once and foreall. The described section can be interpreted in various way. For simulations it is

[1]Document 396'and is annexed to this document

possible to have a buffer underflow, due to the — for the simulation — valid rule to subtract after each Macro block 30 bits.

What do we want

For simulation purposes we want to study the steady state of the codec, and **not** the response of the codec.
The proposal is intended for the simulations only and to have a way that everyone start at the same way. The requirements are:

- the method must be simple to implement
- the method should yield in a simulation environment for future improvements.
- the method should reflect reasonable hardware performance

4.1 The conditions for the first frame

- Coding starts with empty buffer
- Stepsize for the first picture is tabulated for the various values of p:

p = 1	g = 64
p = 6	g = 32
p = 30	g = 8

- Disregarding buffer overflow
- The second picture is coded with half full buffer content

The temporal reference becomes:

original sequence	1	2	3	4	5	6	7	8	(10 Hz)
coded sequence	1	2	3	4	5	6	7	8	(10 Hz)

Temporal reference

As mentioned before the procedure is ment as a resonable start for the very short used sequences.

Annex 1

Amplitude	Code word	Code word length
-14	11111110000	11
-13	11111101111	11
-12	11111101110	11
-11	1111110110	10
-10	1111110101	10
-9	1111110100	10
-8	111111001	9
-7	111111000	9
-6	11111011	8
-5	11111001	8
-4	111110	7
-3	11110	5
-2	1110	4
-1	110	3
0	0	1
1	100	3
2	1010	4
3	10110	5
4	1011100	7
5	10111010	8
6	10111011	8
7	101111000	9
8	101111001	9
9	1011110100	10
10	1011110101	10
11	1011110110	10
12	10111101110	11
13	10111101111	11
14	10111110000	11

Table 1 Adopted VLC for Differential motion vector

Annex 2

Amplitude	Code word	Code word length
-14	1111111111	10
-13	1111111110	10
-12	1111111101	10
-11	1111111100	10
-10	111111101	9
-9	111111100	9
-8	11111101	8
-7	11111100	8
-6	11111011	8
-5	11111001	8
-4	111110	7
-3	11110	5
-2	1110	4
-1	110	3
0	0	1
1	100	3
2	1010	4
3	10110	5
4	1011100	7
5	10111010	8
6	10111011	8
7	10111100	8
8	10111101	8
9	101111100	9
10	101111101	9
11	1011111100	10
12	1011111101	10
13	1011111110	10
14	1011111111	10

Table 2 Proposed VLC for Differential motion vector

4.11.1 Differential Motion Vector Coding -

The coding of the displacement vector with fixed codeword length (FLC) as used in RM5 wastes a lot of bits when the moving area in the scene is large. Instead of transmitting the value of the calculated vector itself, the differential vector is transmitted applying a variable length code.
The differential technique is employed based on a 1-D prediction: the prediction is the motion vector of the previous macro block.
In case of the first macro block in the GOB, the previous vector is zero. The adopted VLC is depicted in table 8.

Amplitude	Code word	Code word length
-14	1111111111	10
-13	1111111110	10
-12	1111111101	10
-11	1111111100	10
-10	111111101	9
-9	111111100	9
-8	11111101	8
-7	11111100	8
-6	11111011	8
-5	11111001	8
-4	111110	7
-3	11110	5
-2	1110	4
-1	110	3
0	0	1
1	100	3
2	1010	4
3	10110	5
4	1011100	7
5	10111010	8
6	10111011	8
7	10111100	8
8	10111101	8
9	101111100	9
10	101111101	9
11	1011111100	10
12	1011111101	10
13	1011111110	10
14	1011111111	10

Table 8 Adopted VLC for Differential motion vector

new page 396'

CCITT SGXV
Working Party XV/4
Specialists Group on Coding for Visual Telephony

Document 396'
1988

TITLE: Description of Ref. Model 6 (RM6)

SOURCE: Specialist Group On Coding for Visual Telephony

version : October 20, 1988

revision 1	: May 20	, 1988	doc #339
revision 2	: July 20	, 1988	
revision 3	: Sept 5	, 1988	doc #375
revision 4	: Oct 19	, 1988	doc #396
revision 5	: Nov 19	, 1988	

CONTENTS

1 INTRODUCTION

This document gives a description of Reference Model 6 (RM6). This model is a configuration which has the ability to operate at px64 kbit/s (p=1,..,30),(see doc # 395)

The objective of this document is to use it as a guideline, new amendments will be added resulting in a new reference model.

The readers are asked to give comments and corrections to remove ambiguous parts.

2 DESCRIPTION OF REFERENCE MODEL 6

In the proceeding text reference model 6 will be described. Starting with the basic format parameter choice referred to as common source input format. The spatial sizes are specified where these are most critical where the temporal frequency could be variable.

3 COMMON SOURCE INPUT FORMAT (C.S.I.F).

The parameters for the C.S.I.F. are:

	Full CSIF	1/4 CSIF
Number of active lines		
Luminance (Y)	288	144
Chrominance (U,V)	144	72
Number of active pixels per line		
Luminance (Y)	360	180
Chrominance (U,V)	180	90

Table 1 : Source format (full CSIF and 1/4 CSIF)

The number of coded pels per line is reduced, because 360 divided by 16 does not yield in an integer value. The obtained format is called significant pel area (SPA).

3.1 Definition Of The Significant Pel Area.

```
            CSIF                                           1/4 CSIF

                    360                                      180
 -------------------------------------     ^     ----------------------   ^
 |      |                    |      |     |     |   |            |   |   |
 |      |                    |      |     |     | 2 |    176     | 2 |  144
 |      |                    |      |     |     |   |            |   |   |
 |  4   |        352         |  4   |    288    |   |            |   |   |
 |      |                    |      |     |     ----------------------   V
 |      |                    |      |     |
 |      |                    |      |     |
 -------------------------------------     V


                180
          ----------------    ^            ---------------    ^
          |  |       |   |    |            | |         | |    |
          | 2|  176  | 2 |   144           |1|   88    |1|    72
          |  |       |   |    |            | |         | |    |
          |  |       |   |    |            ---------------    V
          ----------------    V

                          U = B - Y = C
                                       b

                180
          ----------------    ^            ---------------    ^
          |  |       |   |    |            | |         | |    |
          | 2|  176  | 2 |   144           |1|   88    |1|    72
          |  |       |   |    |            | |         | |    |
          |  |       |   |    |            ---------------    V
          ----------------    V

                          V = R - Y = C
                                       r
```

Figure 1 : Definition significant pel area

The number of pixels of the reduced picture formats become:

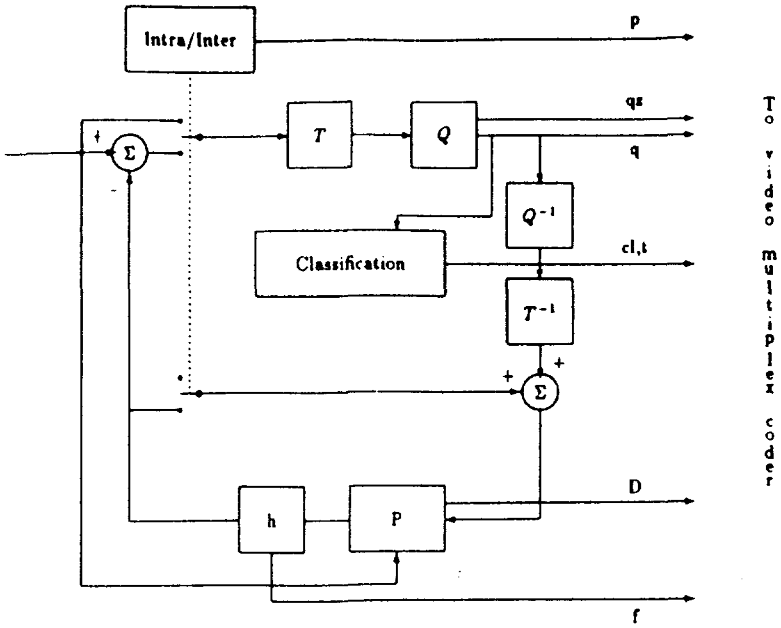
352 x 288	= 101,376	pixels (Y)	
176 x 144 x 2	= 50,688	pixels (U,V)	
total	= 152,064	pixels/frame	(Y,U,V)

CIF

176 x 144	= 25,344	pixels (Y)	
88 x 72 x 2	= 12,672	pixels (U,V)	
total	= 38,016	pixels/frame	(Y,U,V)

1/4 CIF

In table 2 the influence of the frame rate on the number of pixels per second is given. This figure includes the number of pixels for the chrominance as well.



p	=	Flag for intra/inter
P	=	Picture memory
qz	=	Quantization index
q	=	Quantization index for transform coefficients
cl,t	=	Classification index, threshold
D	=	Motion vector
f	=	Switch loopfilter on/off
T	=	Transform
Q	=	Quantization
h	=	Loop filter
T^{-1}	=	Inverse transform
Q^{-1}	=	Reconstruction Quantization

Figure 2 Hybrid transform/DPCM encoder

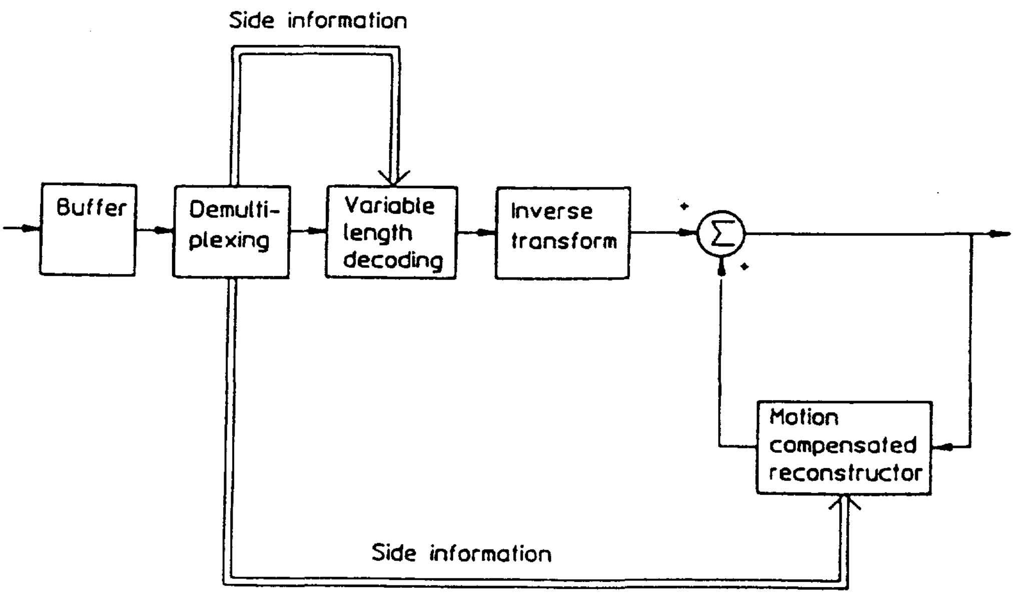


Figure 2 b Hybrid transform/DPCM decoder

Frame Rate 30 Hz	25 Hz	Sub-sampling factor	number significant pixels/second	Mbit/s	number active pixels/second 1/4 CSIF	Mbit/s
15 Hz		1 : 2	2,280,960	18.3	570,240	4.6
	12.5Hz	1 : 2	1,900,800	15.2	475,200	3.8
10Hz		1 : 3	1,520,640	12.2	380,160	3.1
	8.33Hz	1 : 3	1,267,200	10.1	316,673	2.5
7.5Hz		1 : 4	1,140,480	9.1	285,120	2.3

Table 2 Bitrate versus frame rate

The first column of table 2 gives the frame rate and the third column depicts the number of omitted frames in the coding process for full CSIF and 1/4 CSIF the different values are tabulated. Applying the number of active pixels in one frame the total number of pixels per second are given in the next column with the corresponding bitrates.

4 BASICS OF REFERENCE MODEL 6

4.1 Introduction

The used coding configuration is known as a hybrid DPCM/transform coder. Hybrid denotes a technique which involves more than one redundancy reduction technique, in this case interframe methods where the calculations are performed in pixel and transform domain. This coding procedure requires two transforms, i.e. a forward transform and an inverse transform, which are both located in the coding loop. Due to the usage of a block transform the incoming image is partitioned in non-overlapping blocks of N x N pixels . At the moment the blocksize of the transform is set to N=8.
A simple differential pulse coding modulation loop (DPCM) can be identified as the generic structure of the configuration. This DPCM-loop works in the temporal dimension i.e. interframe. For this purpose a frame memory is included in the loop containing the previously reconstructed image or frame. The generic structure of the reference model depicted in figure 2 is based on:

1. Macro blocks
2. Discrete Cosine Transform (DCT)
3. Variable length coding applying a semi-uniform quantizer
4. A zig-zag scanning of quantized coefficients
5. Displacement estimation
6. Buffer control

Figure 2 Hybrid transform/DPCM encoder.

Let us assume a sequence S of images,

$$S = f(t) \quad \text{with} \quad t = \ldots -3,-2,-1,0,1,2, \ldots$$

where f(t) is a 2-D intensity distribution at time t. Denoting the actual frame by f(t) and the previous frame by $f(t-\tau)$, the frame difference fd(t) becomes :

$$fd(t) = f(t) - f(t-\tau)$$

$$\tau = 1 \text{ for skip } 1:1$$
$$\tau = 2 \text{ for skip } 1:2$$

The frames are partitioned in blocks of N x N pixels and are numbered from left to right along a row of blocks. Let b(s,t) denote the intensities of the pixels in a block s at time t and let B(s,t) denote the coefficients of that block after transformation. The block difference bd(s,t) is obtained by subtracting the previous block $b(s,t-\tau)$ from b(s,t) :

$$bd(s,t) = b(s,t) - b(s,t-\tau)$$

Only blocks which have changed significantly are processed. This procedure is known as Conditional Replenishment (CR). With the change detector a distinction is made been between significant and non-significant changed blocks also called block type discrimination (see section 4.6).

The displacement estimation is achieved by a block matching technique with a search of +/- 7 pixels. The blocksize is 16 x 16.

To obtain the displaced block difference the coder applies a displacement vector $\vec{D}$ which might reduce the block difference bd(s,t).

In the case of a translatory motion, the displaced block difference can be expressed as:

$$dbd(s,t) = b(s,t) - b(s_{\vec{D}},t-\tau)$$

where $\vec{D}$ is the obtained displacement vector for the block under consideration.
Let mb(s,t) be a block of size 2N x 2N in the actual frame f(t), and let SW be a M x M search window in a previous frame $f(t-\tau)$, where $M > N$ and $M = 2N+2D_{max}$.
If a brute force method is used and $D_{max} = N-1$, the number of possible integer displacements within this search window becomes $(2N-1)^2$. The prediction error dbd for all these positions is calculated and the displacement vector $\vec{D}$ which produces the minimum error $dbd(s,t-\tau)$ is stored. Zero displacement can be interpretated as the orthogonal projection. After completion of the calculations the minimum error results in the displaced block difference. In a noiseless case, a pure translation by an integer number of pixels will result in an exact match i.e. $b(s_{\vec{D}},t-\tau) = b(s,t)$. The motion trajectory is used to obtain the displaced block difference dbd(s,t).

Only integer displacement is considered, the brute force algorithm is optimal but for implementation purposes a coarse-fine 3 step algorithm is used (see appendix A). For each macro-block the displacement vector $\vec{D}$ is calculated indicating a block in the previous frame which results in the smallest prediction error. The displacement calculations are performed outside the coding loop and therefore this vector has to be transmitted as side information. For the transmission of the non-zero displacement vectors a differential method is adopted using a 1-D prediction of the preceeding calculated

motion vectors (see section 4.11). The differential values are transmitted applying a VLC.
Next the prediction error (dbd(s,t)) is transformed using a 2-D Discrete Cosine Transform with blocksize N = 8.

4.2 Macro Block Approach

A macro block (MB) consists of a 16 x 16 luminance block and the two corresponding 8 x 8 U and V chrominance blocks. The luminance block is divided into four 8 x 8 sub blocks, i.e. a MB consists of six 8 x 8 sub blocks.
The construction is depicted in figure 3.

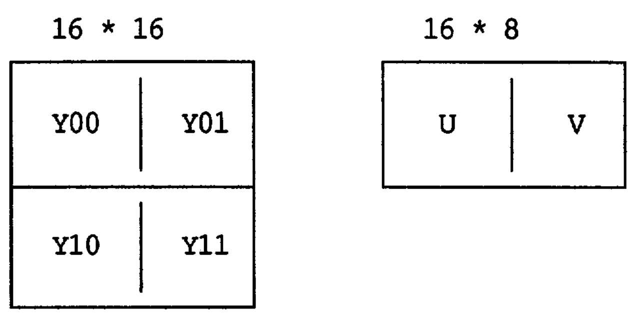


Figure 3 : Construction of a Macro Block (MB)

NOTE : A 16 x 16 Luminance block and the two corresponding 8 x 8 U and V chrominance blocks have the same physical size.

In table 3 the number of macro blocks per frame and the number of group of blocks per frame are shown:

Format	number of GOB in a frame	number of MB in a GOB	total number of MB in a frame
CSIF	18	22	396
1/4 CSIF	9	11	99

Table 3 Relationship between number of Macro blocks and picture format

4.3 Discrete Cosine Transform

The block-differences bd(s[x,y],t) are transformed with the Discrete Cosine Transform (DCT).
The 2-D DCT is defined as :

$$BD(s[u,v],t) = \frac{1}{4} C(u)\, C(v) \sum_{x=0}^{7} \sum_{y=0}^{7} bd(s[x,y],t) \cos\left[\frac{\pi u(2x+1)}{16}\right] \cos\left[\frac{\pi v(2y+1)}{16}\right]$$

with u = 0,1,2, ... 7
v = 0,1,2, ... 7

$$bd(s[x,y],t) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)\, C(v)\, BD(s[u,v],t) \cos\left[\frac{\pi u(2x+1)}{16}\right] \cos\left[\frac{\pi v(2y+1)}{16}\right]$$

with x = 0,1,2, ... 7
y = 0,1,2, ... 7

where x,y = spatial coordinates in the pixel domain
u,v = coordinates in the transform domain

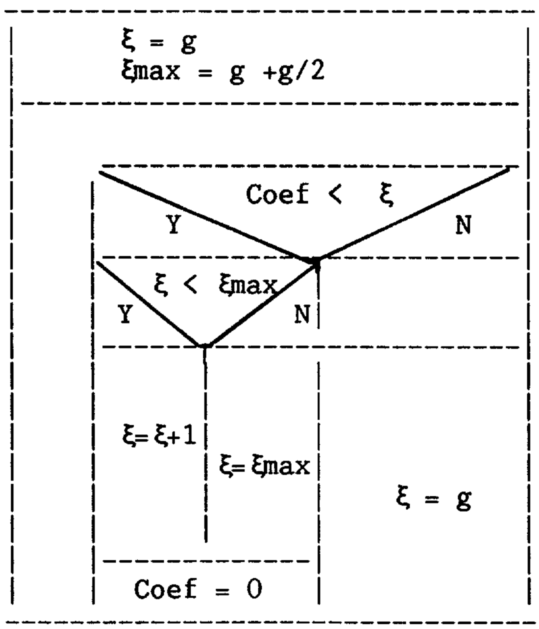
$C(u),C(v) = 1/\sqrt{2}$ for u,v = 0
1 otherwise

The luminance blocks and the chrominance blocks are transformed with a blocksize of 8 x 8 pixels. To assure that the simulation results at the different laboratories are similar it is advisable to exchange the software of the DCT.
The proposed specification fot the IDCT Chips can be found in appendix C.

4.4 The Quantization

4.4.1 Variable Threshold -

A variable threshold is applied independantly of the quantization strategy to increase the number of zero coefficients. In the case of the variable threshold the threshold is variable and its value depends on the length of string of zeroes. It is assumed that the transformed components have been zigzag scanned to form a one dimensional set of coefficients, before the quantization process. The accuracy of the coefficients is 12 bits. Referring to this scale the threshold ξ is modified within the block according to the variable thresholding algorithm as described below.



Example for g=32:

Coefficients	50	0	0	0	33	34	0	40	33	34	10	32
Threshold ξ *	32	32	33	34	35	36	37	38	32	32	32	33
New Coeff.	50	0	0	0	0	0	0	40	33	34	0	0
Quantized value	48	0	0	0	0	0	0	48	48	48	0	0

* The threshold is valid for teh actual coefficient
New Coeff. denotes new coefficients after thresholding and before quantization.

4.4.2 The Quantization Strategy - The result after the transformation and the variable thresholding technique is quantized with an uniform quantizer. The uniform quantizer is defined by a step g and controlled by the buffer state. For RM6 the quantizer threshold has a value T = g.

$$q_{dec}(n) = T + (n-1)\,g \quad , n = 1,2,...$$

$$q_{dec}(0) = 0$$

Taking into account the negative values the expression becomes:

$$q_{dec}(n) = \frac{n}{|n|}\{ T + (|n|-1)\, g \} \quad , |n| = 1, 2, 3,...$$

$$q_{rep}(n) = \frac{q_{dec}(n) + q_{dec}(n + n/|n|)}{2} \quad \text{for } |n| = 1, 2, ...$$

$$q_{rep}(0) = 0$$

with q_{dec} the decision level

q_{rep} the representation level

g the quantizer stepsize

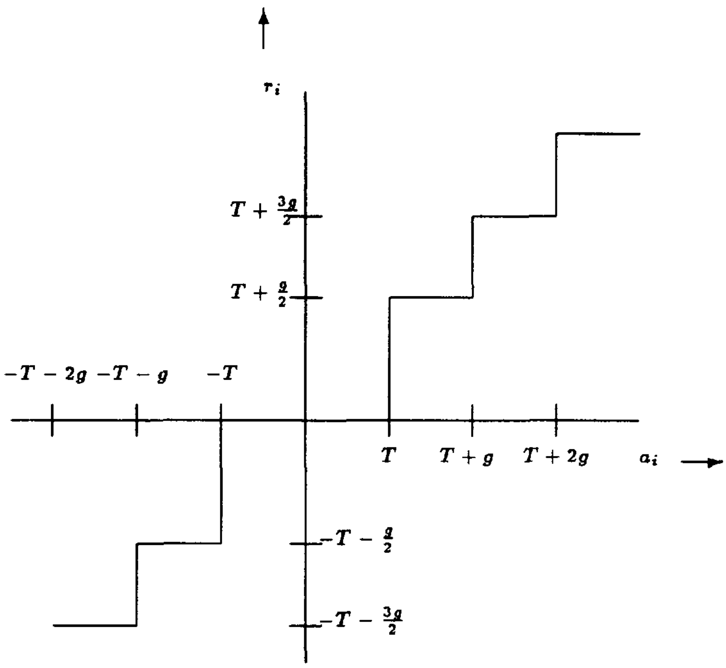
T threshold

Example : A transform coefficient c with :

$$1.0\,g \leq c < 2.0g$$

is quantized to the value of 1.5 g. The characteristic of the quantizer is depicted in figure 4.

Figure 4 . Characteristic of the quantizer



r_i
$T + \frac{3g}{2}$
$T + \frac{g}{2}$
$-T - 2g$
$-T - g$
$-T$
T
$T + g$
$T + 2g$
a_i
$-T - \frac{g}{2}$
$-T - \frac{3g}{2}$

The dynamic range of the coefficients in the case of a blocksize of 8 x 8 is [-2048,2047]. The same quantizer is used both for luminance and chrominance coding.

The quantization stepsize g is transmitted for each group of block.

4.5 Coding Of Coefficients

4.5.1 Scanning Technique -

In order to increase the efficiency of capturing the non zero components a zig-zag scanning class has been adopted:

ZIG - ZAG SCANNING :

1	2	6	7	15	16	28	29
3	5	8	14	17	27	30	43
4	9	13	18	26	31	42	44
10	12	19	25	32	41	45	54
11	20	24	33	40	46	53	55
21	23	34	39	47	52	56	61
22	35	38	48	51	57	60	62
36	37	49	50	58	59	63	64

The transmission of the coefficients must stop when the last non zero coefficient has been reached.

4.5.2 Coding Of The Scanned Coefficients With A Two Dimensional VLC.

To increase coding efficiency a two dimensional variable length code has been adopted. This means that "events" are coded. "Event" is defined as :

event : a combination of a magnitude (non-zero quantization index) and a RUN (Number of zero indexes preceding the current non-zero index)

Coefficients unequal to zero defining the end of the run-length are considered as composite rather than separate statistical event.

The run-length and the magnitude of composite events define the entries of the 2-D VLC table which contains the code words for the composite events. Events are coded with Huffman's algorithm. However, events with low probabilities are coded using fixed length codes. These codes consists of the following three parts.

1. Escape (6 bits) for indicating the use of fixed length codes.

2. Run (6 bits)

3. Level (8 bits; See Note 1).

Note 1: Note that clipping must be introduced for the quantized coefficients F : $-128\,g \leq F < 128\,g$. The maximum range for the non-zero coefficients is now +127g and -128g.

Note 2: $0 \leq \text{run} < 64$ (for blocksize 8)
After the last non-zero coefficient an End-Of-Block (EOB) marker is sent indicating that all other coefficients are zero. The length of the EOB word is two bits.

An example of the two dimensional VLC is given in figure 5 and the table is annexed.

EVENT = (RUN,LEVEL)

Example: (0,3) (1,2) (7,1) EOB

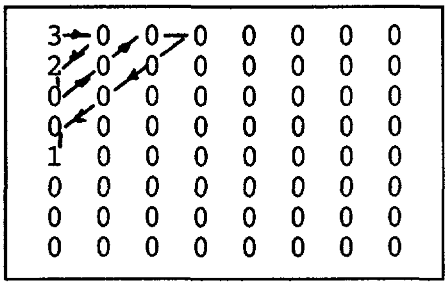


Figure 5 Example 2-D VLC

That means:

* (0,3) The DC component which has the value +3

* (1,2) is next non-zero component according to the zig-zag scanning the number of zeroes is 1.

* The next component is 1 preceded by 7 zeroes, result (7,1)

* EOB is an End of Block marker which indicates that there are no more non zero components.

4.6 Coding Strategy And Block Type Discrimination

In RM6 five different block types can be distinguished :

- no MC not coded
- no MC coded
- MC not coded (only motion vectors are transmitted)
- MC coded
- Intra

The order in which the block type is determined is depicted in figure 6.

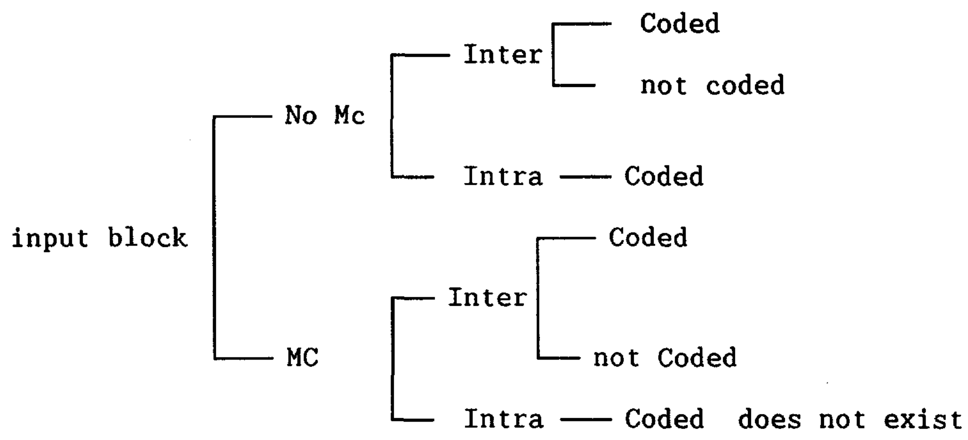


Figure 6 Decision Tree RM6

Macro Block type		code with rel. addr.
1.	no MC not coded	-
2.	MC coded	1
3.	MC not coded	01
4.	no MC coded	001
5.	Intra	000

Table 4 : Adopted VLC for macro block types

If after quantization, all the quantized components of a sub block are zero, the sub block is declared to be not coded (blocktype 1,3).
If all six sub blocks in a MB are not coded, the MB is declared to be not coded. In all other cases the MB is declared to be coded (blocktype 2,4). When one or more blocks in a coded MB are not coded, only an EOB is sent if the subblock is included in the pattern as in section 4.7 (pattern 7 is assumed) :

Y00	Y01	Y10	Y11	U	V

e.g. Y00, Y10, Y11, U and V are not coded

Y01 is coded

```
Y00         Y01                Y10  Y11   U    V
EOB, EVENT-----EVENT EOB, EOB, EOB, EOB, EOB
```

NOTE : The Data Per Block (DPB) is only transmitted if the macro block is coded (blocktype 2,4).

All modified MB's (blocktype 2,3,4) are addressed with relative addressing, similar to the relative addressing used in RM4 (table 5) . The other block-types are coded according to the VLC in table 4.

N.B. : The last string of fixed blocks in a GOB is not encoded.

number of fixed MB's	codeword length
0	1
1	3
2	3
3	4
4	4
5	5
6	5
7	6
8	6
9	7
10	7
11	8
12	8
13	9
14	9
15	11
16	11
17	11
18	11
19	11
20	11
21	11
22	0

Table 5 : Adopted VLC for relative addressing of non-fixed MB's

NOTE : More simulations have to be done to ensure that relative addressing causes visible gain.

4.7 Block Addressing For Macro Block Attribute

By the introduction of the Macro Block scheme in RM5, the side information could be reduced further with the introduction of pattern information. This pattern information consists of a set of seven pattern indicating codec/non-coded blocks within the macro block. The patterns are depicted in figure 7.

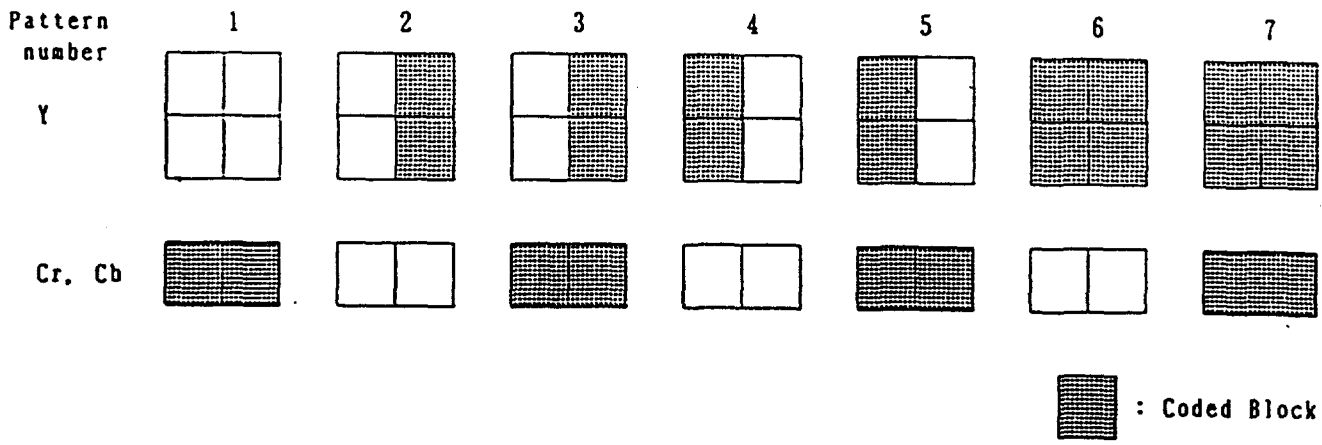


Figure 7 Pattern information Macro block

Pattern number	Code length	Code
6	2	00
2	2	10
4	2	11
7	3	011
1	4	0100
3	5	01010
5	5	01011

Table 6 Code word length Pattern information

NOTE:

If a macro block type is "intra", its pattern information is <u>not</u> transmitted.

Example:
Y00 block is coded, Y01,Y10,Y11,U and V blocks not coded the pattern will be pattern 4.

4.8 Relative Addressing For Blocks Within A Group Of Blocks Structure

The relative addresses is adopted as depicted in figure 8. Run lengths are generated for the number of blocks to the next active blocks.

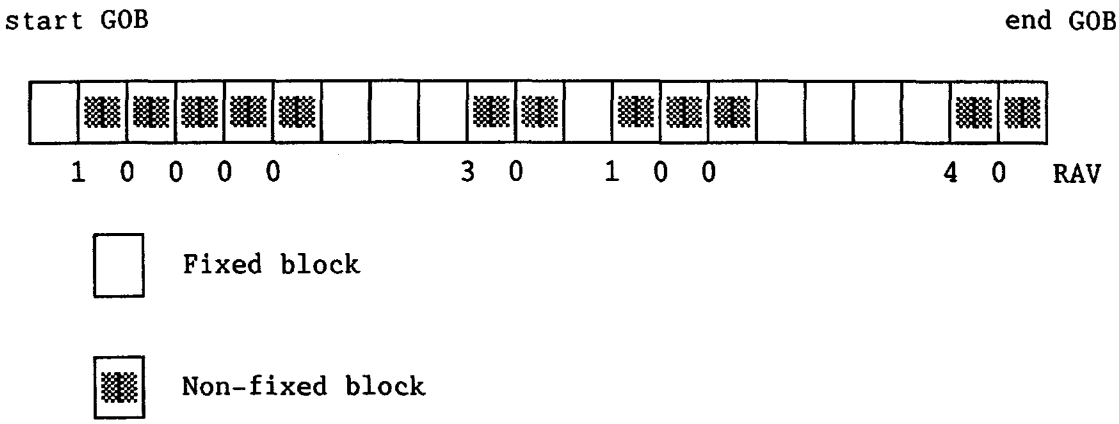


Figure 8 Relative addressing

Where RAV is the relative addressing value (i.e. the number of fixed blocks preceding a non-fixed block).
The last string of fixed blocks in a group of blocks is not encoded.
The GOB start code (see video multiplex) indicate the beginning of the next GOB. The table used for the runs is given in table 5.

4.9 Filter In The Loop

The introduction of a low pass filter after motion compensation (MC) could have the following advantages:

i. A reduction of high frequency artifacts introduced by MC.

ii. A reduction of quantization noise in the feedback loop.

The filter could be controlled with:

1. Displacement vector
2. Prediction error

A filter with impulse response as depicted in (1) is applied on a block of 8 x 8 pixels. The filter is applied both on luminance and the chrominance.

$$h(k,l) = \begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix} \quad 1/16 \qquad (1)$$

4.9.1 Filtering Inside The Block Boundaries. -

At the block boundaries the filter coefficients need to be adjusted in the case of adopting filtering inside the block.

i. $\begin{vmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{vmatrix}$ for pixels inside the block edges

ii. $\begin{vmatrix} 3 & 1 \\ 6 & 2 \\ 3 & 1 \end{vmatrix}$ for pixels on the block edges

iii. $\begin{vmatrix} 9 & 3 \\ 3 & 1 \end{vmatrix}$ for pixels on the block corner positions

In reference model 6 the filter is controlled with the motion vector i.e. if the motion vector is non zero the filter is on. Luminance blocks as well as chrominance blocks are filtered.

4.10 Buffer

4.10.1 Buffer Control

For RM6 the stepsize varies from 4 to 64 with step 2. The dependence on the buffer fullness is depicted in table 7 (the bitrate is p*64 kbit/s). Each GOB the stepsize is adapted according to table 6. This means that 5 bit/GOB are spent because of the 31 different stepsize values.

buffer content [kbit]	stepsize quantizer
< 400*p	——> 4
< 600*p	——> 6
< 800*p	——> 8
.	. .
.	. .
.	. .
< 6000*p	——>60
< 6200*p	——>62
≥ 6200*p	——>64

or : step = 2 * INT (buf_cont / [200*p]) + 2

Table 7 : Quantizer stepsize as a function of the buffer fullness

Where INT denotes the truncation of the fraction, i.e. 1.5 -> 1 , 1.3 -> 1 and 1.6 -> 1.

4.10.2 Buffer Overflow

A buffer size of p*6.4 kbit is intended. After each MB the buffercontent can be calculated (mean 15 bit/MB for p = 1; 10 Hz). When the buffer fullness exceeds p*6.4 kbits, the coefficients and the motion vector are set to zero in the next macro block (however resulting in a small buffer overflow).

N.B. : When the frame rate is not equal 10 Hz or the bitrate is not equal to 64 kbit/s, the mean number of bits per MB can be computed then according to (3).

```
bitrate      : p * 64 kbit/s  ( p = 1..30)

frame rate   : 30 / k Hz      ( k = 1..4)

--> mean number of bits per MB :

    for CSIF     :  5 * k * p  bits/MB                    (3)
    for 1/4 CSIF : 20 * k * p  bits/MB

Example : CSIF, 64 kbit/s and 10 Hz  --> p = 1 and k = 3
                                     --> mean 15 bits/MB
                                     --> 5940 (data-)bits/frame
```

In this way a number of overhead bits are taken into account.

4.11 Motion Estimation

The prediction error can be minimized with motion compensation.

At the moment the 3-step method is adopted in RM6 with blocksize 16 x 16 i.e. macro block based. The method can be found in appendix A.

The 3-step method is applied on luminance only. The motion vector for the chrominance is derived from the luminance by dividing the luminance vector by two and truncate the result to integer value.

example :

for luminance	-->	for chrominance
(3, 2)		(1, 1)
(-5,-6)		(-2,-3)

4.11.1 Differential Motion Vector Coding -

The coding of the displacement vector with fixed codeword length (FLC) as used in RM5 wastes a lot of bits when the moving area in the scene is large. Instead of transmitting the value of the calculated vector itself, the differential vector is transmitted applying a variable length code.
The differential technique is employed based on a 1-D prediction: the prediction is the motion vector of the previous macro block.
In case of the first macro block in the GOB, the previous vector is zero. The adopted VLC is depicted in table 8.

Amplitude	Code word	Code word length
-14	11111110000	11
-13	11111101111	11
-12	11111101110	11
-11	1111110110	10
-10	1111110101	10
-9	1111110100	10
-8	111111001	9
-7	111111000	9
-6	11111011	8
-5	11111001	8
-4	111110	7
-3	11110	5
-2	1110	4
-1	110	3
0	0	1
1	100	3
2	1010	4
3	10110	5
4	1011100	7
5	10111010	8
6	10111011	8
7	101111000	9
8	101111001	9
9	1011110100	10
10	1011110101	10
11	1011110110	10
12	10111101110	11
13	10111101111	11
14	10111110000	11

Table 8 Adopted VLC for Differential motion vector

4.12 MC/No MC Decision

For the moment we use the characteristic as defined in RM4 (adapted to blocksize 16 x 16). The evaluation function for displacement estimation is a sum of absolute differences concerning to all of the pels in a block. The characteristic whether to suppress the displacement vector is depicted in figure 7. The characteristic is determined experimentally.

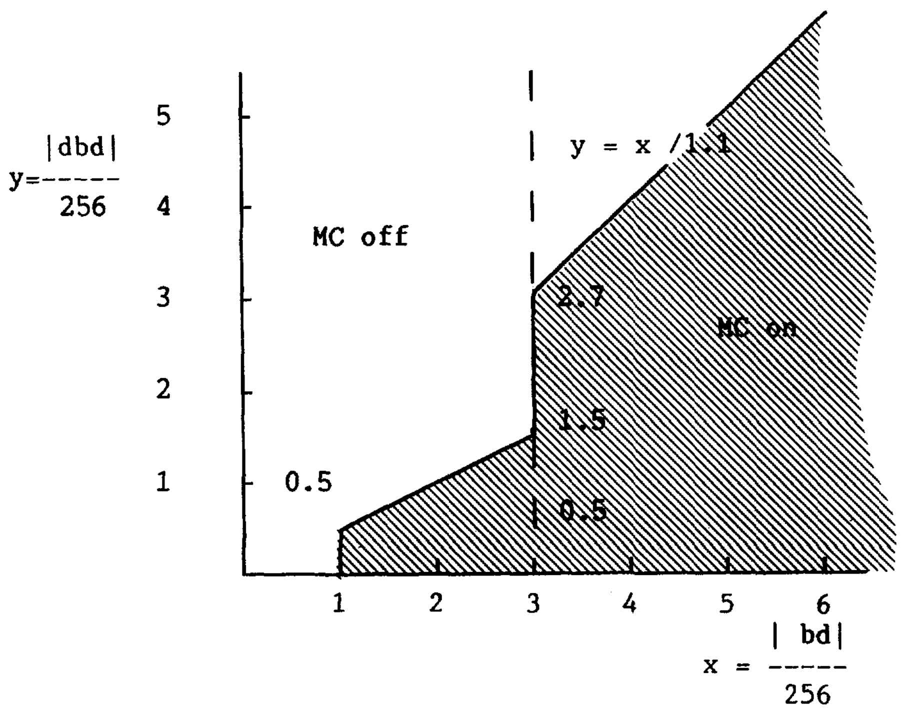


Figure 9. Characteristic MC/ No MC

Where dbd denotes the displaced block difference and bd the block difference, see also page 9.

NOTE: MC off includes the solid line. This characteristic resolves partly the sticking noise in the uncovered background (#107).

4.13 Intra Mode Decision

The most important reason for the re-introduction of the inter/intra switch are:

1. much better performance at scene cuts, too fast movement and in areas of decovered background (e.g. when Trevor raises his hands)

2. better error resilience and very simple implementation of (necessary) forced update

The implementation of the decision can be described with:

```
 -------------------------------------
|         k = 1, 16                   |
|     --------------------------------|
|    |    l = 1, 16                   |
|    |    ----------------------------|
|    |   |  OR = O(k,l)               |
|    |   |----------------------------|
|    |   | Dif   = OR - S(k,l)        |
|    |   | VAR   = VAR+ Dif*Dif       |
|    |   | VAROR=VAROR+OR*OR          |
|    |   | MWOR =MWOR + OR            |
|    |   |                            |
|-------------------------------------|
| VAR = VAR/256                       |
| VAROR=VAROR/256-MWOR/256*MWOR/256   |
 -------------------------------------
```

Where: O(k,l) denotes the pixels in the original macro block
S(k,l) denotes the pixels of the motion compensated estimated macro block.

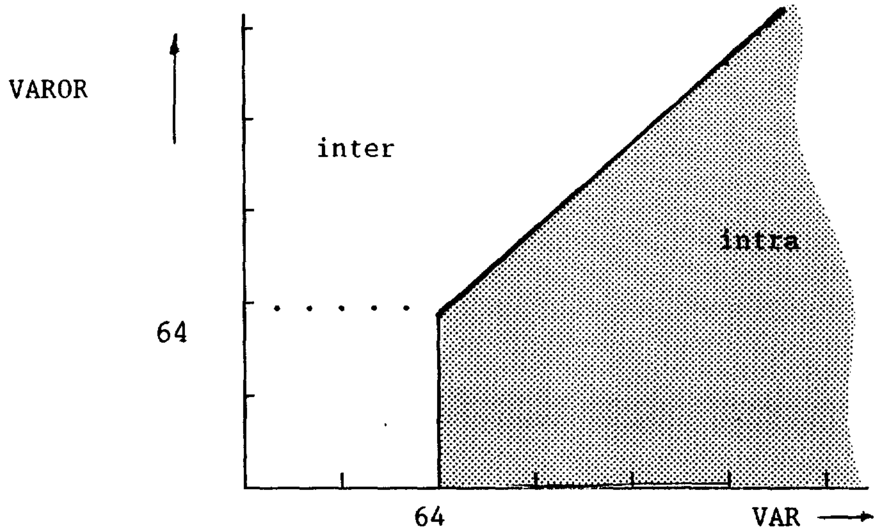


Figure 10. Characteristic intra / inter

NOTE1:
The parameters are calculated as integer values.
The lower threshold of 64 for the decision
is emperically optimized.
The decision is depicted in figure 10.

NOTE2:
Inter mode includes the solid line

After the blocks have been declared intra the coefficients are transmitted as depicted below:

FOR LUMINANCE (Y)

M=rounded mean value of the 16x16 MB	Differential DC Value of LSB1 by comparison with 8 x M	AC	EOB
8 bits FLC	VLC		2bits

. . . .

Differential DC Value of LSB4 by comparison with 8 x M	AC	EOB

FOR CHROMINANCE (U,V)

Differential DC Value of U SB by comparison with 1024	AC	EOB	Differential DC Value of V SB by comparison with 1024	AC	EOB

where MB = Macro block
L = Luminance
U,V= Chrominance
LSB= Luminance Sub Block

The comparison with 1024 is derived from the average value of the chrominance components (128 * 8).

Example:
Suppose mean value of 16 x 16 Macroblock is 10, than the DC value's of Y00,Y10,Y01 and Y11 have the value 80 see formula on page 12. Therefore the differential DC value is compaired wit 8 x M.

4.14 First Picture And Scene Change

To overcome the problem of a buffer overflow at coding the first picture, the second picture is skipped. This means that for the first picture the amount of available bits is twice compared to the average amount of bits per picture.

original sequence	1	2	3	4	5	6	7	8	(10 Hz)
coded sequence	1	1	3	4	5	6	7	8	(10 Hz)

Table 9 : Temporal reference

Conditions for coding the first frame

- Coding starts with zero buffer occupancy.
- Stepsize for the first GOB is 16.
- For each MB at least the averages of Y, U and V are sent. Depending on the buffer fullness and quantizer stepsize, also some AC-components are coded. If there are one or more AC-components in a MB non-zero, the macro block type (BT) becomes coded. In other cases the block-type is not coded, the MB contains only the mean values.

The transmission order becomes :

```
------ ---------------- ---------------- ----------------
| BT | | mean value Y | | mean value U | | mean value V |
------ ---------------- ---------------- ----------------
1 bit      8 bits              8 bits              8 bits
                                -------------
when BT = 1 (fixed)   -->     | next MB |
                                -------------
                                ------------------------------------ ------------
     BT = 0 (coded)   -->     | event(s) and EOB's for subblocks | | next MB |
                                ------------------------------------ ------------
```

Transmission order

This yields to :

- at least 9900 bits for the first frame, i.e. :

396 Macro blocks per frame

```
Per Macro block : - 1 bits indicating not coded/coded (block-type)
                 - 8 bits for Y
                 - 8 bits for U
                 - 8 bits for V
                ----------------- +
                  25 bits/MB    =>  396 * 25 bits/frame
```

When a buffer overflow occurs (mean 2 * (5*p*k) bits/MB, twice as much as for the next pictures, see section 4.6), the macro block-type is forced to fixed (which means that 25 bits/MB are sent) until there is no overflow anymore.
Simulation results show that after a few GOB's, the stepsize will be maximal resulting in a lot of fixed MB's.

WE NEED TO CLARIFY THIS PROCEDURE EVERYONE HAS IMPLEMENTED ITS OWN INTERPRETATION, WITH THE DISASTERS COVERED IN THE STATISTICS. (CIFS)!!!!!!

5 PRESENTATION OF RESULTS

INSTITUTE

STATISTICS RM6
SEQUENCE :
MODIFICATION :

DATE :

FRAME RATE:

ITEM			15th Picture	Mean whole seq.
1. RMS for luminance				
2. SNR for luminance				
for chrominance (u)				
for chrominance (v)				
3. Mean value of step size				
4. Mean value of the number of non-zero coefficients				
5. Mean value of the number of zeroes before the last NZ-coefficient				
6. Block type of MACRO	FIXED			
	CODED MC			
	FIXED MC			
	CODED			
	INTRA			
7. Block type of Y	FIXED			
	CODED MC			
	FIXED MC			
	CODED			
	INTRA			
8. Block type of UV	FIXED			
	CODED MC			
	FIXED MC			
	CODED			
	INTRA			
9. Number of bits	Macro attributes mvlmb			
	End of block			
	Motion vectors			
	Coefficients	Y		
		U		
		V		
		Total		
	Total			

Comments to table : Presentation of results

The statistics exclude the first frame. Because the second picture to code is skipped, the number of tracks for the statistics is equal to the number of coded tracks - 2.

ad 1 :

$$P_{err}(t) = \sum_{i=1}^{352} \sum_{j=1}^{288} [\hat{f}(i,j) - f(i,j)]^2$$

$$P_{norm} = \frac{1}{\text{number pixels in frame}} P_{err}(t)$$

$$RMS = \sqrt{P_{norm}}$$

with:

$P_{err}(t)$ = frame based energy error

$f(i,j)$ = original pixel value

$\hat{f}(i,j)$ = reconstructed pixel value

P_{norm} = normalized error

RMS = root mean square error

ad 2 : The SNR is calculated for luminance as well as chrominance.

$$SNR = 20 \log \frac{255}{RMS}$$

ad 3 : The quantizer is adapted after a group of blocks GOB according to the buffer status. Therefore the mean value of the stepsize becomes :

$$E(g) = \frac{\sum^{pic} \text{step size of each GOB}}{\text{no. of GOBs (18)}}$$

ad 4 and 5 :
For the calculation of the mean value of the number of non-zero coefficients (MVNZC) and zero-coefficients (MVZC) the DC-component is included.

ad 7 : The number of fixed Y-blocks must be at least 4 x the number of fixed MB's, because

- a fixed MB consists of 4 fixed Y blocks
- a coded MB consist of 0 or more (but not all) fixed Y blocks

ad 8 : A similar story as in 7 holds for the number of fixed UV-blocks (at least 2 x the number of fixed MB's)

ad 9 : The number of bits for the MB attributes also includes the bits for relative addressing.

number of bits for 'Motion vectors' must be equal to (excluding rounding errors) :

8 x (#macro coded MC + #macro fixed MC)

MVLMB denotes bits for Mean Value of Luminance Macro Block (see intra mode, section 4.13)

INDEX

APPENDIX A

3-STEP-ALGORITHM

Assuming a maximum displacement of 7 pixels the step algorithm iterates in three steps to the resulting minimum absolute error.

Step 1 : The actual block B is matched with 9 blocks in the previous window SW at the following positions :

(-4-4)	(0-4)	(4-4)
(-4 0)	(0 0)	(4 0)
(-4 4)	(0 4)	(4 4)

The position of the non-shifted prediction is (0 0). The order of the search has been defined as:

2	3	4
5	1	6
7	8	9

If the calculated error in position 1 has the same value as on position 3 than position 1 is chosen. The error is calculated by the boolean less than i.e truncation.

3-STEP-ALGORITHM

Step 2 : For the second step a new search pattern is used. The best match of step 1 is the center of this pattern :

```
(-2-2)   (0-2)   (2-2)

(-2 0)   (0 0)   (2 0)

(-2 2)   (0 2)   (2 2)
```

Step 3 : The position of the best match in step 2 is the central position of the third and final search pattern :

```
(-1-1)(0-1)(1-1)
(-1 0)(0 0)(1 0)
(-1 1)(0 1)(1 1)
```

The best match of step 3 is the resulting minimum match error.

An example of the search process for the 3-step algorithm :

```
1       1       1

            2   2   2

1       1   2   1   2
                  3 3 3
            2   2 3 2 3
                  3 3 3
1       1       1
```

APPENDIX B

ADOPTED VARIABLE LENGTH CODES

Word Length of VLC for Two-dimensional Coding

LEVEL (absolute value)

RUN	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	. .128
0	3	5	6	8	9	9	11	13	13	13	13	14	14	14	14	20	20	20	20	20	
1	4	7	9	11	13	14	14	20	20	20	20	20	20	20	20	20	20	20	20	20	
2	5	8	11	13	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
3	6	9	13	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
4	6	11	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
5	7	11	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
6	7	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
7	7	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
8	8	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
9	8	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
10	9	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
11	9	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
12	9	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
13	9	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
14	11	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
15	11	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
16	11	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
17	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
18	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
19	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
20	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
21	13	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
22	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
23	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
24	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
25	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
26	14	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
27	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
28	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	20	
.																					
.																					
63																					

** Word lengths of all other EVENTs (combination of RUN and LEVEL) are 20: ESCAPE CODE(6bits)+RUN(6BITS)+LEVEL(8bits).

** Word length of EOB is 2.

Code Set for Two-dimensional Coding of Coefficient
Quantization Index (RM4)

No	RUN	LEVEL	CODE LENGTH	CODE	CODE STRUCTURE
1	0	-1	3	111	VLC
2	0	1	3	110	
3	1	-1	4	1011	VLC
4	1	1	4	1010	
5	2	-1	5	10011	
6	2	1	5	10010	VLC
7	0	-2	5	10001	
8	0	2	5	10000	
9	3	-1	6	001111	
10	3	1	6	001110	
11	4	-1	6	001101	VLC
2	4	1	6	001100	
3	0	-3	6	001011	
4	0	3	6	001010	
5	5	-1	7	0001111	
6	5	1	7	.	VLC (4bits)
7	1	-2	7	.	
8	1	2	7	.	+
9	6	-1	7	.	
20	6	1	7	.	FLC (3bits)
1	7	-1	7	.	
2	7	1	7	0001000	
3	8	-1	8	00001111	
4	8	1	8	.	VLC (5bits)
5	0	-4	8	.	
6	0	4	8	.	+
7	9	-1	8	.	
8	9	1	8	.	FLC (3bits)
9	2	-2	8	.	
30	2	2	8	00001000	
31	10	-1	9	001001111	VLC (6bits)
2	10	1	9	.	
3	0	-5	9	.	+
4	0	5	9	.	
5	1	-3	9	.	
6	1	3	9	.	FLC (3bits)
7	3	-2	9	.	
8	3	2	9	001001000	
9	11	-1	9	000001111	
40	11	1	9	.	VLC (6bits)
1	12	-1	9	.	
2	12	1	9	.	+
3	0	-6	9	.	
4	0	6	9	.	FLC (3bits)
5	13	-1	9	.	
6	13	1	9	0000010000	

No	RUN	LEVEL	CODE LENGTH	CODE	CODE STRUCTURE
95	22	-1	14	00000000111111	
6	22	1	14	.	
7	23	-1	14	.	
8	23	1	14	.	
9	24	-1	14	.	
100	24	1	14	.	
1	25	-1	14	.	
2	25	1	14	.	
3	26	-1	14	.	
4	26	1	14	.	
5	0	-12	14	.	
6	0	12	14	.	
7	0	-13	14	.	VLC (9bits)
8	0	13	14	.	
9	0	-14	14	.	+
110	0	14	14	.	
1	0	-15	14	.	FLC (5bits)
2	0	15	14	.	
3	1	-6	14	.	
4	1	6	14	.	
5	1	-7	14	.	
6	1	7	14	.	
7	2	-5	14	.	
8	2	5	14	.	
9	3	-4	14	.	
120	3	4	14	.	
1	5	-3	14	.	
2	5	3	14	.	
3	9	-2	14	.	
4	9	2	14	.	
5	10	-2	14	.	
6	10	2	14	00000000100000	
127	EOB WORD		2	01	VLC
128	ESCAPE CODE		6	001000	VLC

** Other EVENTs (combination of RUN and LEVEL) are coded to:
ESCAPE CODE(6bits)+RUN(6bits)+LEVEL(8bits)

No	RUN	LEVEL	CODE LENGTH	CODE	CODE STRUCTURE
47	4	-2	11	00000011111	
8	4	2	11	.	
9	14	-1	11	.	
50	14	1	11	.	
1	15	-1	11	.	
2	15	1	11	.	VLC (7bits)
3	1	-4	11	.	
4	1	4	11	.	+
5	2	-3	11	.	
6	2	3	11	.	FLC (4bits)
7	0	-7	11	.	
8	0	7	11	.	
9	5	-2	11	.	
60	5	2	11	.	
1	16	-1	11	.	
2	16	1	11	00000010000	
3	17	-1	13	0000000111111	
4	17	1	13	.	
5	6	-2	13	.	
6	6	2	13	.	
7	0	-8	13	.	
8	0	8	13	.	
9	3	-3	13	.	
70	3	3	13	.	
1	1	-5	13	.	
2	1	5	13	.	
3	18	-1	13	.	VLC (8bits)
4	18	1	13	.	
5	19	-1	13	.	+
6	19	1	13	.	
7	0	-9	13	.	FLC (5bits)
8	0	9	13	.	
9	20	-1	13	.	
80	20	1	13	.	
1	21	-1	13	.	
2	21	1	13	.	
3	7	-2	13	.	
4	7	2	13	.	
5	2	-4	13	.	
6	2	4	13	.	
7	0	-10	13	.	
8	0	10	13	.	
9	4	-3	13	.	
90	4	3	13	.	
1	8	-2	13	.	
2	8	2	13	.	
3	0	-11	13	.	
4	0	11	13	0000000100000	

APPENDIX C

PROPOSED SPECIFICATION FOR INVERSE DCT CHIPS

PROPOSED SPECIFICATION FOR INVERSE DCT CHIPS

1. Generate random integer pixel data values in the range -L to +H according to the attached random number generator (C). Arrange into 8x8 blocks.* Data sets of 10,000 blocks each should be generated for (L=256,H=255), (L=H=5) and (L=H=300).

2. For each 8x8 block, perform a separable, orthonormal, matrix multiply, Forward Discrete Cosine Transform (FDCT) using at least 64-bit floating point accuracy.

3. For each block, round the 64 resulting transformed coefficients to the nearest integer values. Then clip them to the range -2048 to +2047. This is the 12-bit input data to the inverse transform.

4. For each 8x8 block of 12-bit data produced by step 3, perform a separable, orthonormal, matrix multiply, Inverse Discrete Cosine Transform (IDCT) using at least 64-bit floating point accuracy. Round the resulting pixels to the nearest integer, and clip to the range -256 to +255. These blocks of 8x8 pixels are the 'reference' IDCT output data.

5. For each 8x8 block of 12-bit data produced by step 3, use the proposed IDCT chip or an exact-bit simulation thereof to perform an Inverse Discrete Cosine Transform. Clip the output to the range -256 to +255. These blocks of 8x8 pixels are the 'test' IDCT output data.

6. For each of the 64 IDCT output pixels and for each of the 10,000 block data sets generated above, measure the peak, mean and mean square error between the 'reference' and 'test' data.

7. For any pixel, the peak error should not exceed 1 in magnitude.
   For any pixel, the mean square error should not exceed 0.06.
   Overall, the mean square error should not exceed 0.02.
   For any pixel, the mean error should not exceed 0.015 in magnitude.
   Overall, the mean error should not exceed 0.0015 in magnitude.

8. All-zeros in must produce all zeros-out.

9. Rerun the measurements using exactly the same data values of step 1, but change the sign on each pixel.

* by allocating each set of consecutive 8 numbers in a row.

END

```
                        /*L and H must be long, ie, 32 bits*/
long rand(L,H)
long   L,H;
{
static long    randx = 1;   /*long is 32 bits*/
static double     s = (double)0x7fffffff;

long    i,j;
double x;                       /*double is 64 bits*/

        randx = (randx * 1103515245) + 12345;
        i = randx & 0x7fffffff;                    /*keep 31 bits*/
        x = ( (double)i ) / s;                      /* range 0 to 1.0 */
        x *= (L+H);                             /* range 0 to L+H */
        j = x + 0.5;                            /*rounded integer*/
        return( j - L );
}
```

APPENDIX D

FIGURES

APPENDIX E

TABLES

APPENDIX F

ACTION POINTS

1. Filtering in loop (Doc #356 and #376). The following topics have to be checked:

   - The number of filters in the loop

   - The position of th filter(s); before, after or both sides of the frame memory

   - The adaptivity of the filter(s) and how to control it

   - The effectiveness of the filter(s) in the loop v.s. outside the loop

2. DC intra prediction: coding of the prediction error (related to modification intra/inter mode.

3. Motion estimation technique: use of surrounding motion vectors for the search technique; size and shape of search area

4. Variable threshold: adaptation rules (related to modification of variable threshold in quantizer)

5. Directional Transform(DT): Information will be provided as soon as possible by Norway.

6. Impact of the modification at 384kbit/s

7. Number of scanning classes

8. Impact of the GOB configuration as described in Doc. 385 and 373

9. VLC optimization