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

Document 375
1988

TITLE: Description of Ref. Model 5 (RM5)

SOURCE: The Netherlands, France, FRG, Italy, Sweden, BTRL

version : March 17, 1988

revision 1 : May 20 , 1988
revision 2 : July 20 , 1988
revision 3 : Sept 5 , 1988

CONTENTS

1 INTRODUCTION

This document gives a description of Reference Model 5 (RM5). This model is a configuration which has the ability to operate at px64 kbit/s (p=1,..,30)

The objective of this document is to use it as a guideline, new ammendments 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 5

In the proceeding text reference model 5 will be described. First 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) Chrominance (U,V)	 288 144	 144 72
Number of active pixels per line Luminance (Y) Chrominance (U,V)	 360 180	 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.

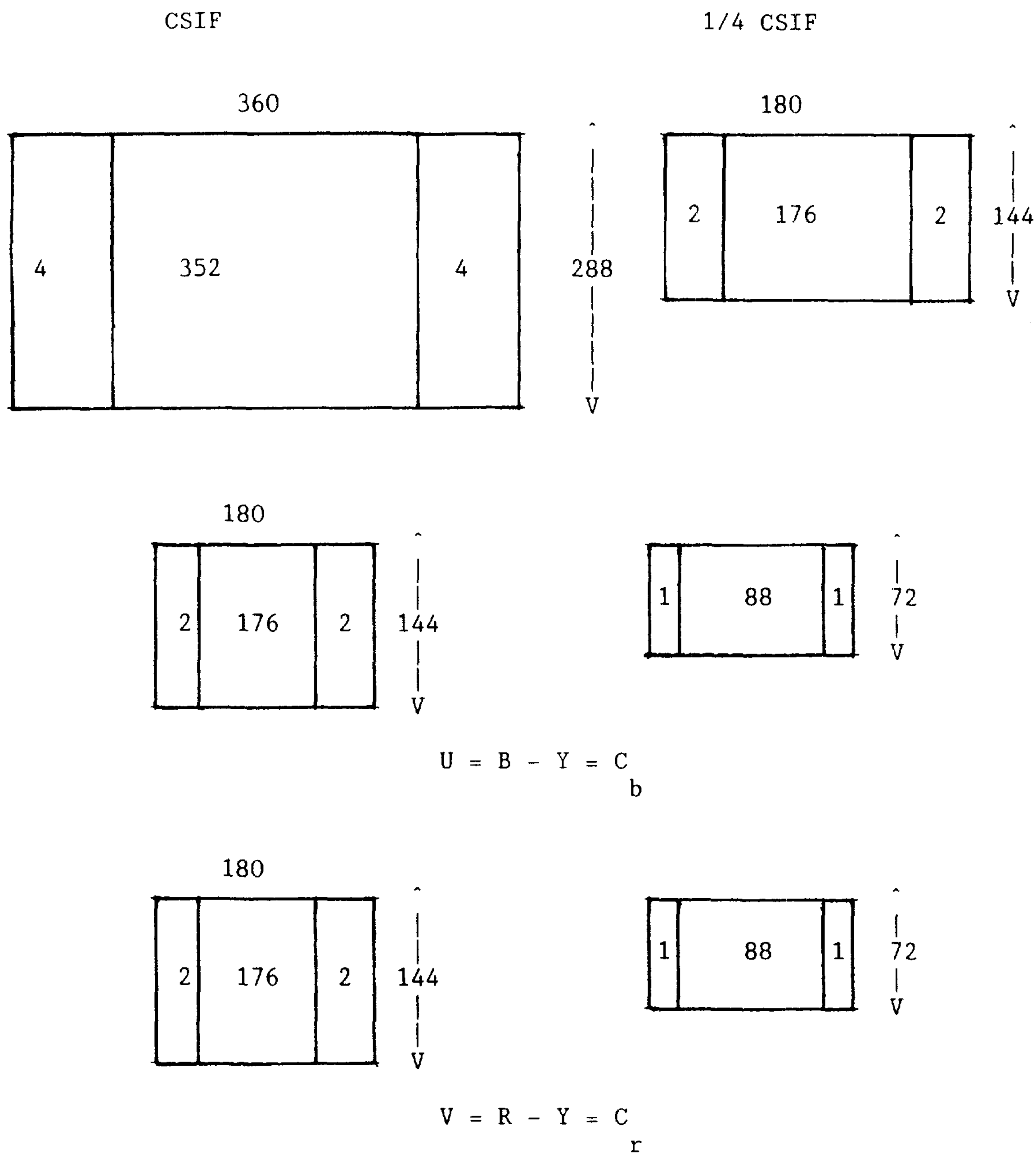


Figure 1 : Definition significant pel area

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

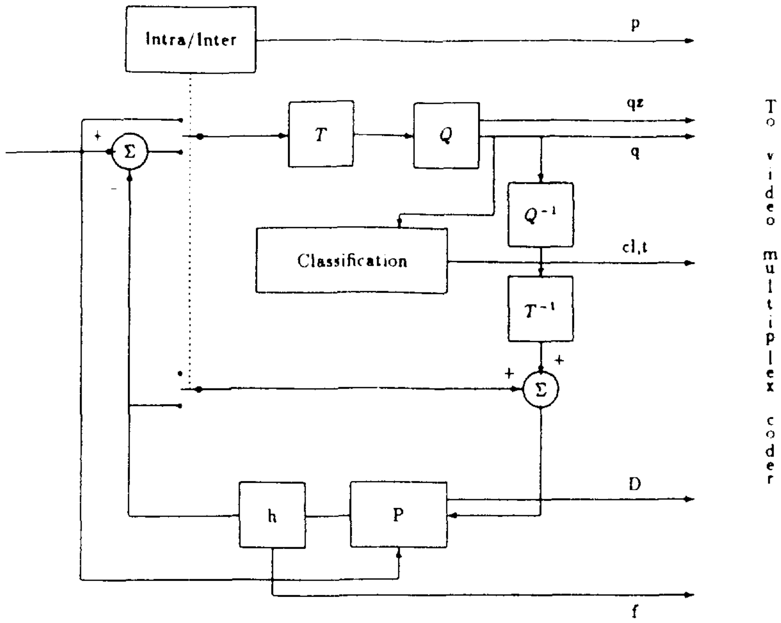
1/4 CIF

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.

Frame Rate 30 Hz	Frame Rate 25 Hz	Sub-sampling factor	number active 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 ommited 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.



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

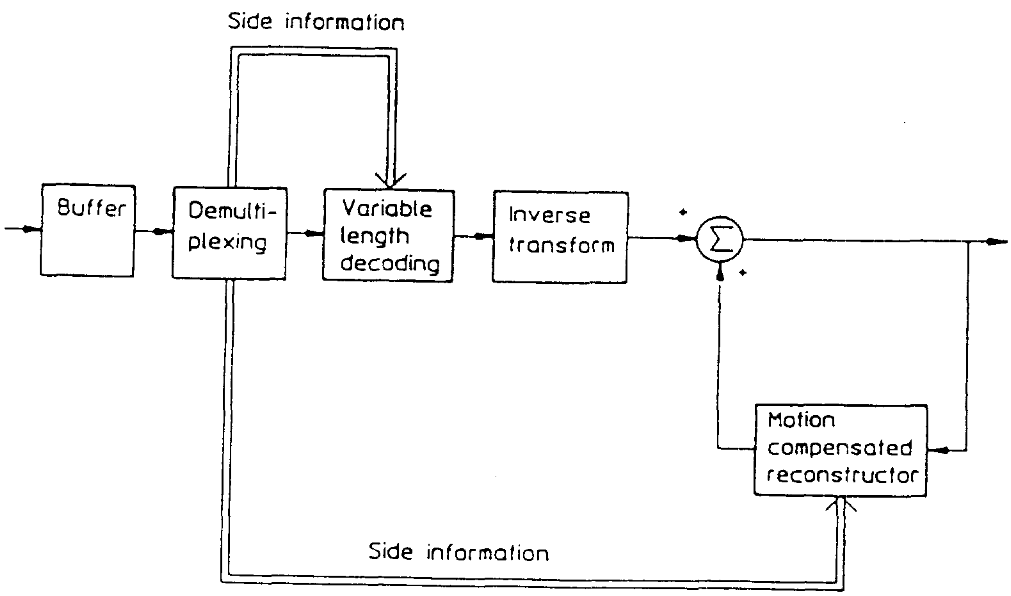


Figure 2 b Hybrid transform/DPCM decoder

4 BASICS OF REFERENCE MODEL 5

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,

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

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 :

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

$\tau = 1$ for skip 1 : 1
$\tau = 2$ 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 b(s,t) be a block of size N x N 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 $M = N+2D_{max}$.
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 D). 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. The non-zero displacement vectors are transmitted using an 8 bit fixed length code (FLC).

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 subblocks, i.e. a MB consists of six 8 x 8 subblocks.
The construction is depicted in figure 3.

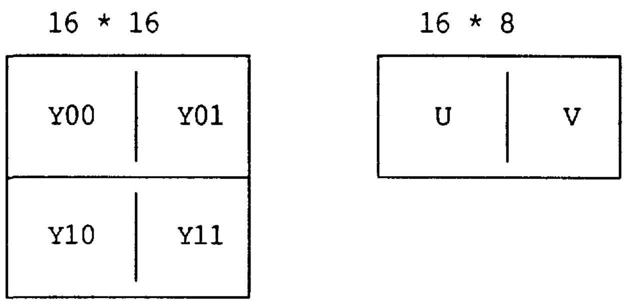


Figure 3 : Construction of a Macro Block (MB)

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 $u = 0,1,2, \ldots 7$
$v = 0,1,2, \ldots 7$

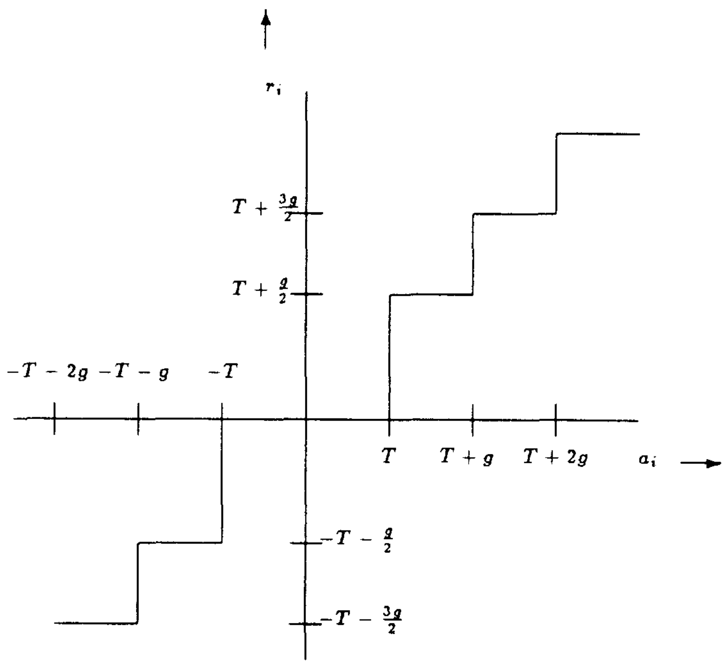
$$bd(s[x,y],t) = \frac{1}{4} \sum_{u=0}^{7} \sum_{v=0}^{7} C(u)\, C(v)\, BD(s[u,v],t) \cos\left[\frac{\pi u(2x+1)}{16}\right] \cos\left[\frac{\pi v(2y+1)}{16}\right]$$

with $x = 0,1,2, \ldots 7$
$y = 0,1,2, \ldots 7$

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.



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

4.4 Quantization Strategy

The result after the transformation is quantized with an uniform quantizer. The uniform quantizer is defined by a step g and controlled by the buffer state. For RM5 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

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 preceeding 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 +128g and -128g.

Note 2: $0 \leq$ 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

3	0	0	0	0	0	0	0
2	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
1	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

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 preceeded 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 RM5 four different block types can be distinguished :

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

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

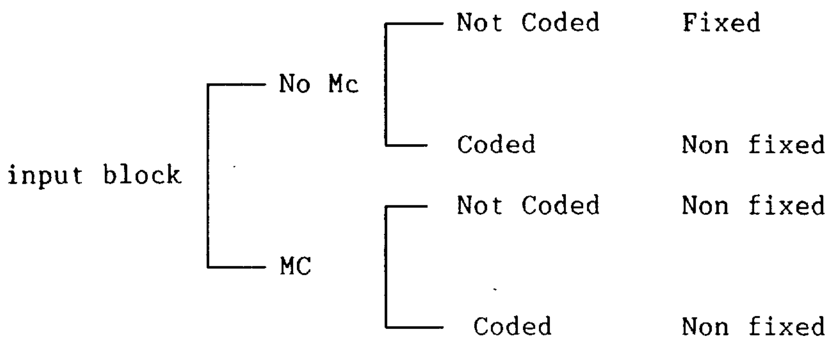


Figure 6 Decision Tree RM5

Macro Block type		code with rel. addr.	code without rel. addr.
1.	no MC not coded	-	1
2.	MC coded	1	01
3.	MC not coded	01	001
4.	no MC coded	00	000

Table 4 : Adopted VLC for macro block types

If after quantization, all the quantized components of a subblock are zero, the subblock is declared to be not coded (blocktype 1,3).
If all six subblocks 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 for these blocks :

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 Relative Addressing For Blocks Within A Group Of Blocks Structure

The relative addresses is adopted as depicted in figure 10. Run lengths are generated for the number of blocks to the next active blocks.

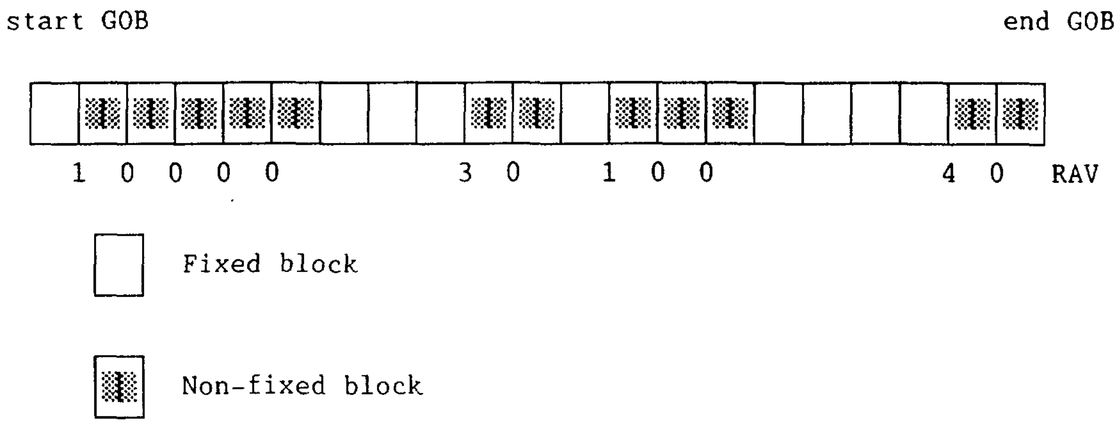


Figure 10 Relative addressing

Where RAV is the relative addressing value (i.e. the number of fixed blocks preceeding 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.

4.8 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.

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

4.8.1 <u>Filtering Inside The Block Boundaries.</u> -

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 5 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.

4.9 Buffer

4.9.1 Buffer Control

For RM5 the stepsize varies from 4 to 64 with step 2. The dependence on the buffer fullness is depicted in table 6 (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 ⟶	52
< 6200*p ⟶	58
> 6200*p ⟶	64

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

Table 6 : quantizer stepsize as a function of the buffer fullness

4.9.2 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.

4.10 Motion Estimation

The prediction error can be minimized with motion compensation.

At the moment the 3-step method is adopted in RM5 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 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.

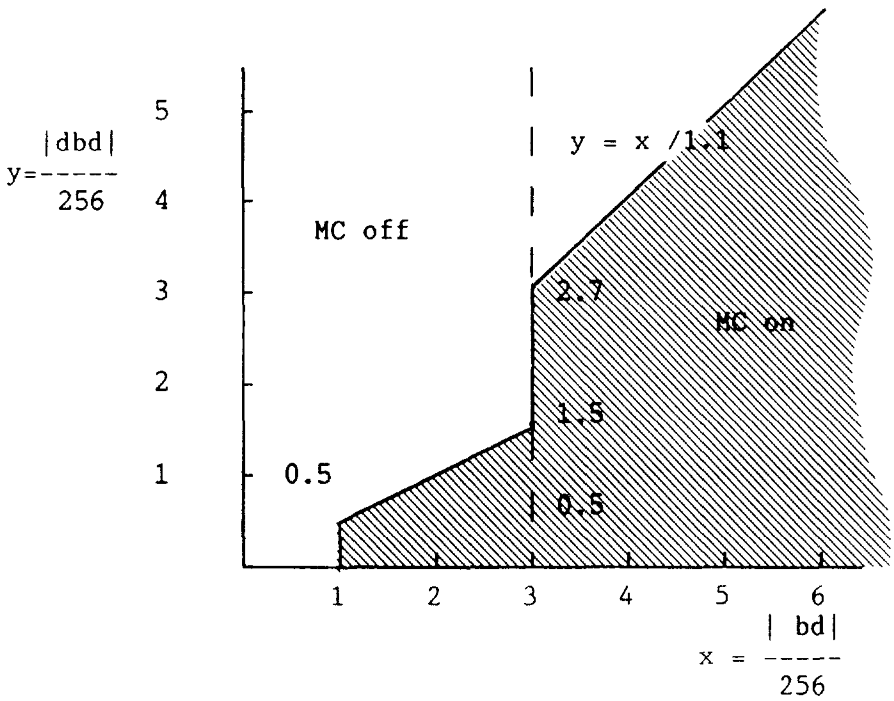


Figure 7. Characteristic MC/ No MC

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).

4.12 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 7 : 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 Macroblocks per frame

```
Per Macroblock : - 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 4.8.2), 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.

5 PRESENTATION OF RESULTS

INSTITUTE

STATISTICS RM5 DATE :
SEQUENCE :
MODIFICATION : 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			
7. Block type of Y	FIXED CODED MC FIXED MC CODED			
8. Block type of UV	FIXED CODED			
9. Number of bits	Macro attributes			
	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 :

$$P_{err}(t) = \sum_{i=1}^{\overset{312}{\cancel{360}}} \sum_{j=1}^{288} [\, \hat{f}(i,j) - f(i,j)\,]^2$$

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

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

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.

The number of bits for 'End of block' must be equal to (excluding rounding errors) :

12 (6 blocks) x (#macro coded MC + #macro coded)

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

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

INDEX

APPENDIX A

MODIFICATIONS TO RM4

1. Macro blocks.

   A macro block is defined as a 16·16 Y-block (four 8·8 blocks), an 8·8 U block and a 8·8 V block, corresponding to the same physical picture area.

2. Addressing of macro blocks.

   A non-fixed macro block is addressed with a similar relative addressing as used in RM4.

3. Macro block types.

   Non-fixed macro blocks can be of three types, which are signalled with the following VLC.

MC, coded	1
MC, not coded	01
no MC, coded	00

   With "coded" is meant that at least one of the six 8·8 blocks have non-zero transform coefficients. For "coded" macro blocks six EOB words are transmitted (Same 2-D VLC as in RM4).

4. Motion compensation.

   One motion vector is transmitted for each "MC" macro block.

5. Motion compensation of chrominance.

   The motion vector is applied also on the chrominance. To cope with the lower resolution, the Y-MV is devided by two with truncation to integer value.

Example. Y-MV = (+3,+2) => U,V-MV = (+1,+1)
Y-MV = (-5,-6) => U,V-MV = (-2,-3)

(SNR measurements on U and V should be presented in the future).

6. Filter in the loop.

The filter is applied within 8x8 Y block as well as within the 8x8 U and V blocks.

7. Signalling of filter in the loop.

Filter is applied in "MC" macro blocks.
(No extra side information)

8. Intra.

Intra mode is only used for the first image. The mean values of Y,U and V are first transmitted with 8 bits each (24 bits per macro block). Mean values are calculated with rounding (1.49 = 1 , 1.5 = 2) The residual between original picture and mean value reconstruction is transmitted with inter coding (both DC and AC).
Scene cuts are not handled separately.

9. Quantizer.

The quantizer can have the step sizes 4,6,8,...,64.
(31 different values)

10. Choise of quantizer.

The same quantizer is used in the whole GOB (18 GOBs per picture). Quantizer is selected from buffer content at the start of the GOB according to the following formula.

step = 2 * INT (buf_cont / 200) + 2

11. Quantizer at scene cut.

For the first GOB in a scene cut picture the step size is forced to be minimum 16.

12. Buffer overflow.

If the buffer overflows during coding of a macro block, the next macro block is forced to be fixed. (This may give some small buffer overflows, which are not critical for simulations)

13. Number of GOBs.

There are 18 GOBs in a picture, each having 22 macro blocks.

14. Bit rate.

To simulate p·64 kbit/s the following number of bits are used.

30 Hz	p·5	bits / macro block
15 Hz	p·10	bits / macro block
10 Hz	p·15	bits / macro block
7.5 Hz	p·20	bits / macro block

This number of bits are subtracted from the buffer content after each macro block. (For p=1 the simulation bit rate becomes 59.4 kbit/s.)

15. Buffer size.

Buffer size is derived from the bit rate (p·64 kbit/s).

$$\text{buffer size} = p \cdot 6400 \text{ bits}$$

APPENDIX B

3-STEP-ALGORITHM

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

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

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

3-STEP-ALGORITHM

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 C

FIGURES

APPENDIX D

TABLES

APPENDIX E

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	VLC
6	2	1	5	10010	
7	0	-2	5	10001	
8	0	2	5	10000	
9	3	-1	6	001111	VLC
10	3	1	6	001110	
11	4	-1	6	001101	
2	4	1	6	001100	
3	0	-3	6	001011	
4	0	3	6	001010	
5	5	-1	7	0001111	VLC (4bits) + FLC (3bits)
6	5	1	7	.	
7	1	-2	7	.	
8	1	2	7	.	
9	6	-1	7	.	
20	6	1	7	.	
1	7	-1	7	.	
2	7	1	7	0001000	
3	8	-1	8	00001111	VLC (5bits) + FLC (3bits)
4	8	1	8	.	
5	0	-4	8	.	
6	0	4	8	.	
7	9	-1	8	.	
8	9	1	8	.	
9	2	-2	8	.	
30	2	2	8	00001000	
31	10	-1	9	001001111	VLC (6bits) + FLC (3bits)
2	10	1	9	.	
3	0	-5	9	.	
4	0	5	9	.	
5	1	-3	9	.	
6	1	3	9	.	
7	3	-2	9	.	
8	3	2	9	001001000	
9	11	-1	9	000001111	VLC (6bits) + FLC (3bits)
40	11	1	9	.	
1	12	-1	9	.	
2	12	1	9	.	
3	0	-6	9	.	
4	0	6	9	.	
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 F

PROBLEMS TO BE SOLVED

* Intraframe mode

  Using the intra frame mode extra side information bits are needed. The algorithm could perform a little better BUT extra decisions are necessary. We have to consider a method for the refresh mechanism. If we do not use the intra frame mode it is recommendable to use forced update.

* Scene cut

  We should agree up on a method how to handle the scene cut see section 4.11.

  It was recommended to use a stepsize of 16 but from simulations a buffer overflow occurred after the first row of macroblocks. We should agree on another build up (progressive build up ?)

* relative adressing

  Studies are needed to use the relative addressing, possible a small gain is to be expected but the hardware complexity will be increased.