AVC-166

INTERNATIONAL ORGANIZATION FOR STANDARDIZATION
ORGANISATION INTERNATIONALE DE NORMALISATION
ISO/IEC JTC1/SC2/WG11
CODING OF MOVING PICTURES AND ASSOCIATED AUDIO

ISO/IEC JTC1/SC2/WG11
MPEG 91/230
18 Nov. 1991

Source:	Tsuyoshi Hanamura, Wataru Kameyama, Syun-ichi Sekiguchi and Hideyoshi Tominaga (JAPAN)
Title:	Hierarchical Video Signal Coding (Pre-registration Number: #38)
Purpose:	Proposal
Relevant Group:	MPEG/VIDEO

Mailing Address: Tsuyoshi Hanamura, Wataru Kameyama
Department of Elec. & Comm. Eng.,
School of Science and Engineering,
Waseda University
3-4-1 Ohkubo, Shinjuku-ku
Tokyo 169 JAPAN
TEL: +81 3 3205 4501
FAX: +81 3 3205 4997
E-mail: hana@tom.comm.waseda.ac.jp
momoko@tom.comm.waseda.ac.jp

1 Introduction

We propose a new video coding scheme that enables hierarchical video signal processing in MPEG II. In this proposed coding scheme, each picture is divided into hierarchical layer pictures. And the prediction between neighbor layer pictures in each picture is also used, as well as ordinary predictions that are I to P, P to P and P to B. We call the former prediction method "layer prediction", and the latter method "temporal prediction".

The first layer of this hierarchical video coding scheme consists of 59.94Hz non-interlaced pictures that have same dimensions with CCIR Rec. 601, and are generated by a certain algorithm from CCIR Rec. 601 video signal. The second layer of it consists of 29.97Hz non-interlaced pictures that have almost same dimensions with SIF/CIF, so that MPEG I level video signal is available on this layer. The third layer of it consists of 9.99Hz non-interlaced pictures that have almost same dimensions with QCIF. The forth layer consists of 9.99Hz non-interlaced pictures that sizes are suitable to be displayed as a icon in a certain window system for the workstation environment.

For calculating motion vector, the hierarchical motion vector estimation method, that is developed by us, is applied. This method gives precise motion vectors, less computational time than ordinary method and individual motion vectors for each layer.

In our implementation, hierarchical layer pictures are generated by reducing each dimension to half from 59.94Hz non-interlaced pictures as a starting layer. However, another reduction method by a certain scaling factor, such as 3, to generate layers are able to be implemented when appropriate up-sampling and down-sampling filters are available. Also, the number of layers are variable depending on your implementation, nevertheless 4 is selected in our proposal.

Our proposed coding scheme is mostly based on MPEG I syntax except "layer prediction".

Because different sizes of video image are obtained from one source video signal, this kind of hierarchical video coding is preferable to conventional video coding in the multimedia communication environment.

2 Algorithm

2.1 System Overview

As shown in Figure 1, the encoder is composed of following blocks:

1. pre-process
2. motion vector estimation
3. layer encoder

The pre-process block converts CCIR Rec. 601 video signal into 59.94Hz non-interlaced pictures. The motion vector estimation block provides precise motion vectors that are need for each layer process. The layer encoder acts a main role in this encoder system. It takes motion vectors for this layer, an up-sampled picture from the lower layer encoder and a down-sampled picture from the pre-process, which is not down-sampled in the first layer, as inputs. And it puts pictures for higher layer process and encoded data to coding buffer as output.

As shown in Figure 2, the decoder is composed of following blocks:

1. layer decoder
2. post-process

The layer decoder takes coded data for each layer and an up-sampled picture form the lower layer as inputs, and decodes pictures for each layer. The post-process selects which layer to be displayed and converts obtained non-interlaced pictures into interlaced pictures if necessary.

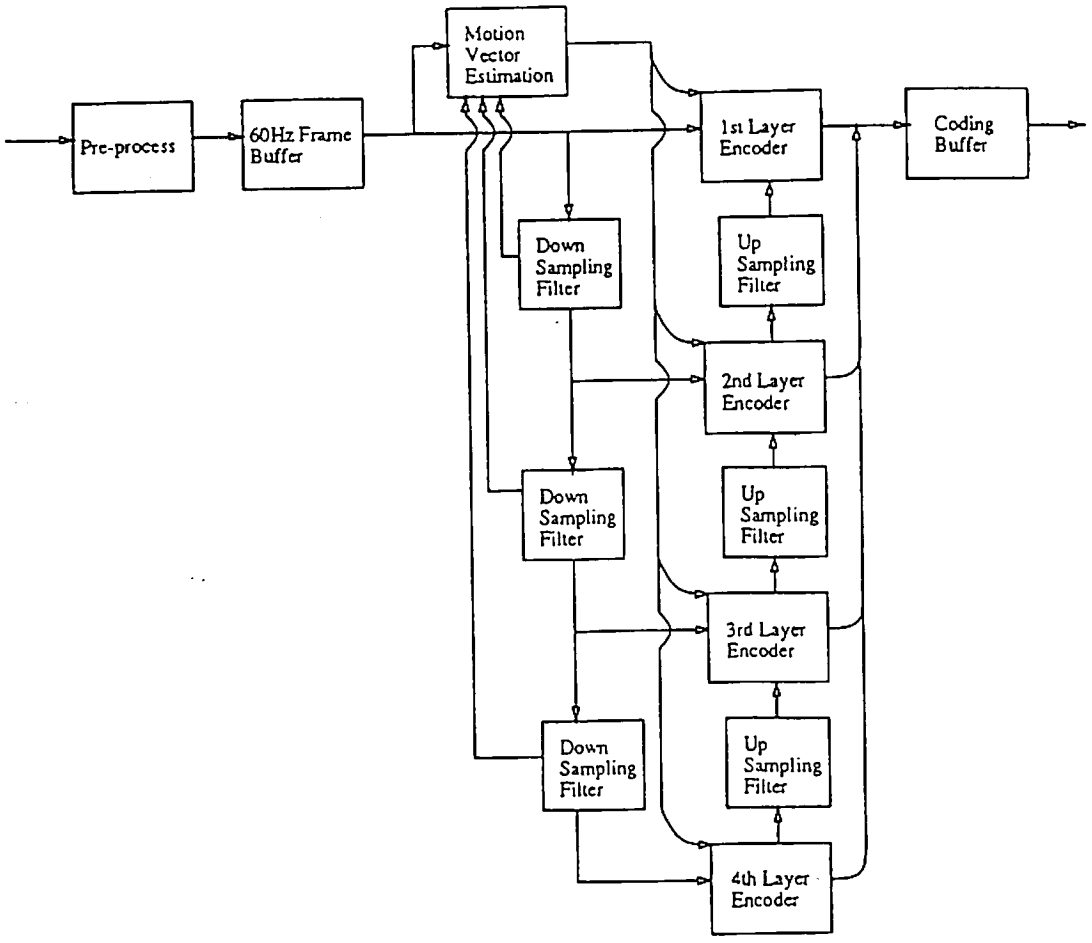


Figure 1: Encoder overview

2.2 Pre-process

2.2.1 Making Non-interlaced TV Signal

Generally, the interlaced scanning of video introduces artifacts (interline flicker, line crawl), unless the moving image is properly pre-filtered with a non-separable low-pass filter. Thinking about the coding efficiency of the block-based transform coding with motion compensation, the interlaced video signal increases the number of coded bits, since high frequency coefficients of DCT domain become large owing to motion detection error and low vertical correlation.

Under these consideration, in this proposal, it is assumed that the non-interlaced video signal is given to the encoder as inputs. To get the non-interlaced signal from the interlaced NTSC signal (CCIR Rec.601), a deinterlacing scheme is employed[1]. This deinterlacing method is based on the time-recursive motion compensated prediction. For the motion estimation (Sec 2.3), we use hierarchical motion estimation with half pel accuracy, which is different from that of [1]. After this process, 59.94Hz non-interlaced input signal to the encoder is obtained. Figure 3 shows the block diagram of this pre-process.

2.2.2 Picture Format

To reduce the number of coded pixels, the given 59.94Hz non-interlaced pictures (4:2:2 format) is converted into those of 4:2:0 format by sub-sampling crominance signal (Cb, Cr) vertically after filtering. Then the size of Y component picture is reduced into 704x480 from 720x480 by cutting 8 pels from each side of a picture. Also, the size of Cb and Cr component pictures are reduced into 352x240.

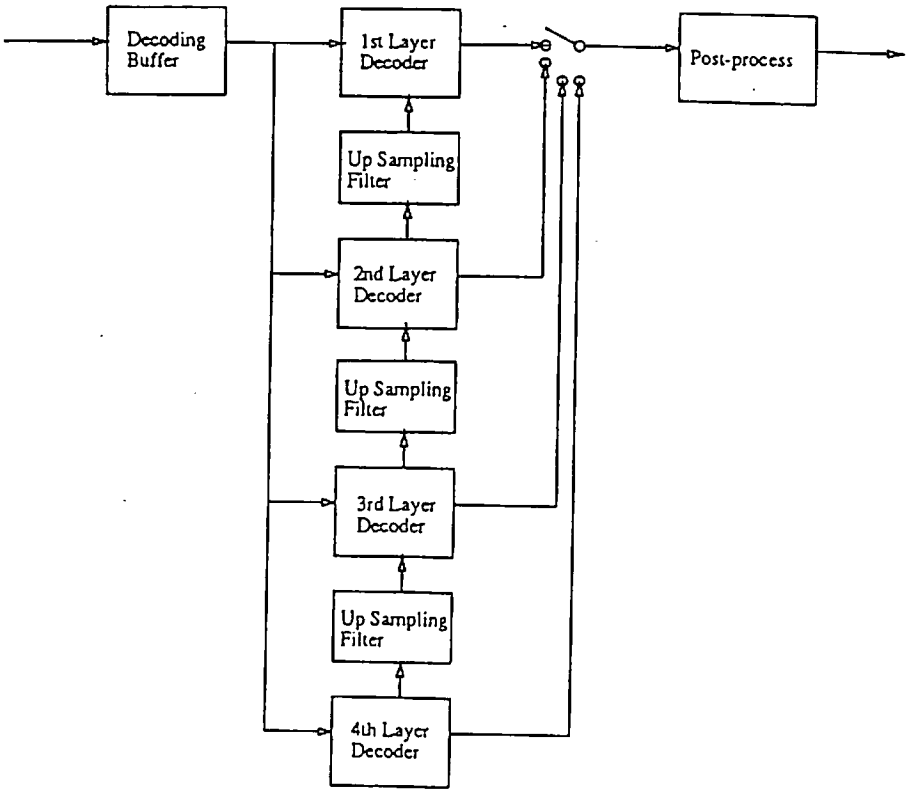


Figure 2: Decoder overview

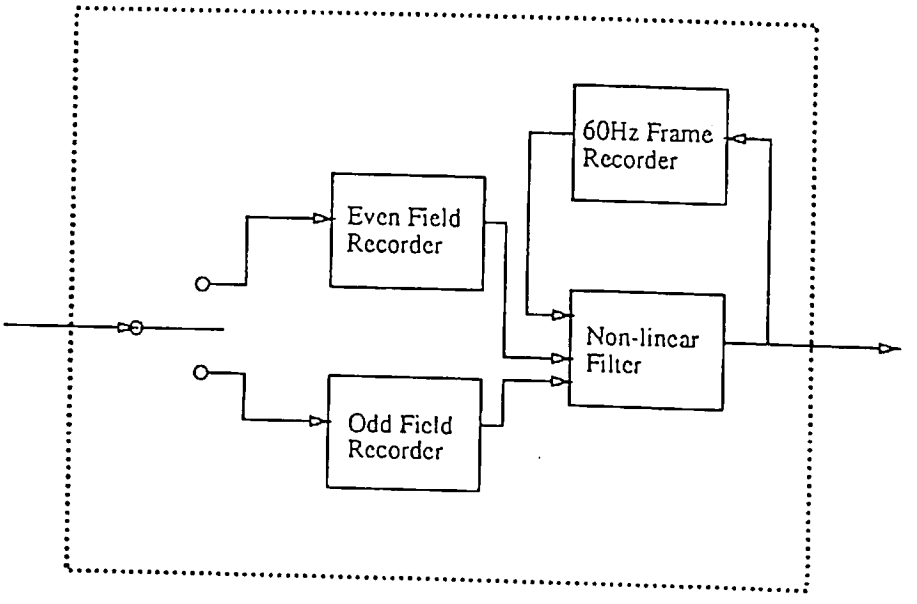


Figure 3: Pre-processor

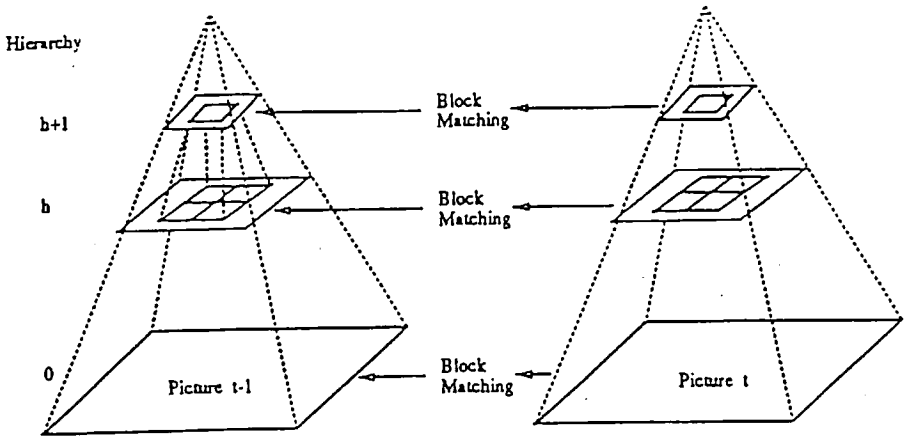


Figure 4: Overview of Hierarchical Motion Estimation

2.3 Hierarchical Motion Estimation

To realize the hierarchical video signal coding, every macroblock of each layer picture must have the motion vector of itself. From other view point, motion detection using conventional method needs large number of additions to enlarge the coverage area of motion detection for P- and B-picture predictions.

Under these considerations, in this proposal, a hierarchical motion estimation method is employed[2]. The overview of this method is shown in Figure 4.

In the hierarchical layer pictures, the motion is detected for each layer from a lower resolution layer to a higher resolution layer. In each layer the motion is detected using the block matching method with initial vectors that are twice the result of detection of a lower layer. Consequently, the motion detected by block matching in each layer is represented as follows:

$$\begin{aligned} V_h &= 2V_{h+1} + V'_h \\ &= 2^2 V_{h+2} + 2V'_{h+1} + V'_h \\ &= 2^{h_{max}-h} V_{h_{max}} + 2^{h_{max}-h-1} V'_{h_{max}-1} + \cdots + V'_h \\ &= \sum_{i=h}^{h_{max}-1} 2^{i-h} V'_i \end{aligned} \tag{1}$$

Where V_{h+1} is the motion vector detected in (h+1)-th layer, V'_h is the differential vector from the initial vector, V_h is motion in h-th layer.

In each layer, the range of the vector search is ± 2 pel and the motion is detected with half pel accuracy in this implementation. Figure 4 shows the overview of this hierarchical motion estimation method.

2.4 Prediction —Layer Prediction—

2.4.1 Picture Type

The overview of prediction is shown in Figure 5 and 6.

The prediction scheme is almost same as that of MPEG I except for using an up-sampled picture which is a spatially lower resolution picture. As shown in Figure 6, encoding proceeds in order of the number with circle. Every I-picture and P-picture has 4 layers each. Every B-picture, which is between I-picture and/or P-pictures and two picture interval far from I- or P-picture, has 2 layers for each. Rest of pictures is called T-picture which has only one layer.

In the beginning of encoding GOP, each layer picture of an I-picture (No.1–4) is encoded. At first, the picture having the lowest resolution of an I-picture is encoded in entirely same way as an

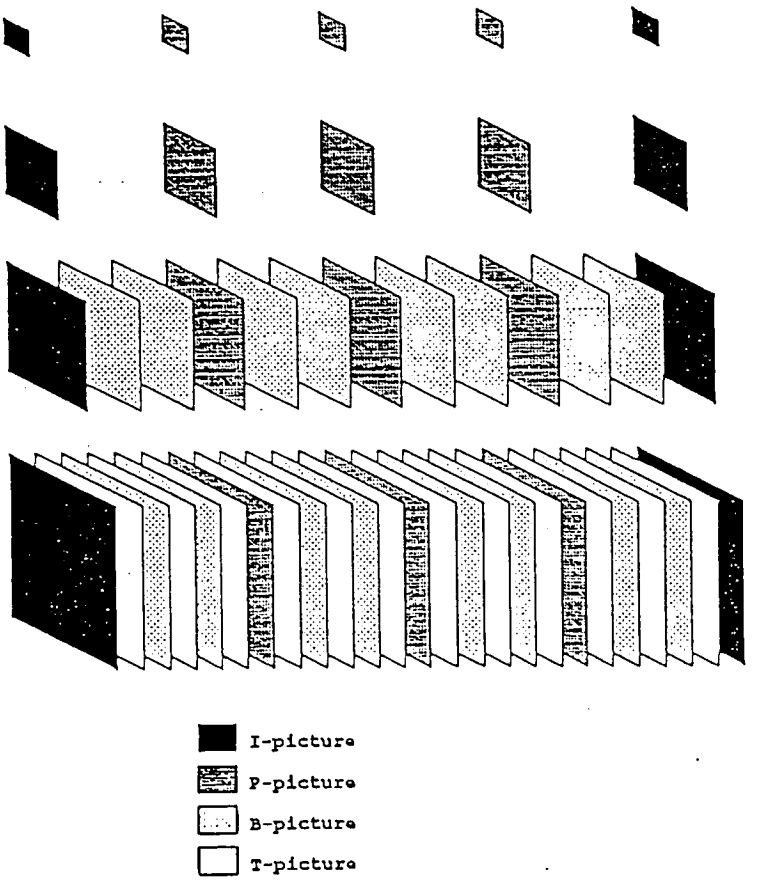


Figure 5: Overview of Hierarchical Pictures

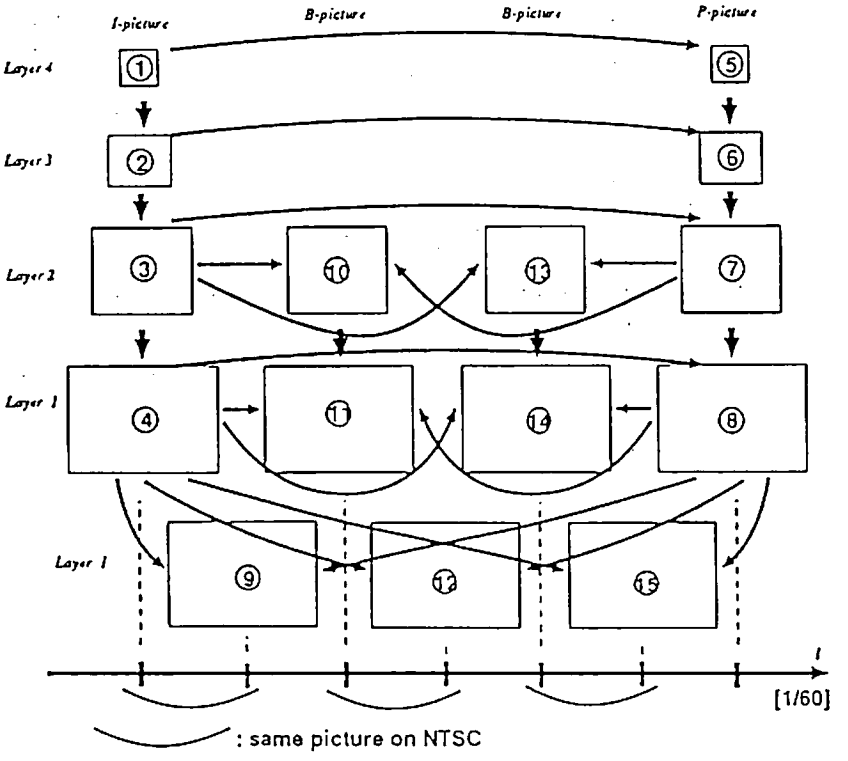


Figure 6: Picture Type and Prediction Overview

I-picture of MPEG I. In the next step, a picture having the higher resolution is encoded, predicted by the up-sampled picture of the lower resolution.

After all layer pictures of an I-picture are encoded, each layer picture of a P-picture (No.5–8) is encoded. The picture having the lowest resolution of a P-picture is also encoded in same way as a P-picture of MPEG I, predicted by the same layer of the I- or P-picture encoded previously. Then a picture having the higher resolution of a P-picture is encoded, predicted by the up-sampled picture of the lower resolution and same layer pictures of already encoded I-picture or P-picture. In this case, the prediction can be done from two directions, one direction or none.

After all layer pictures of P-pictures or I-pictures is encoded, the pictures having only one layer (No.9) is encoded. In our proposal, this kind of picture is called T-picture. In the syntax, however, a T-picture is not distinguished from the highest layer pictures of a B-picture. Every T-picture is encoded in entirely same way as a B-picture of MPEG I.

After a T-picture is encoded, the lowest resolution of a B-picture (No.10) is encoded in same way as B-pictures of MPEG I and T-pictures except for picture sizes. Then the higher resolution layer is encoded, predicted by two same layers of I- or P-pictures and up-sampled picture of the lower layer of a B-picture. In this case, the prediction can be done from three directions at most.

Summarizing these layer picture types, our proposed syntax has six different picture coding types as shown in Table 1.

After all, IT- PT- and BT-picture have entirely same coding types as I-, P- and B-pictures of MPEG I respectively.

Table 1: Picture Types

picture type	coding type
I-picture:	IT-picture (lowest resolution layer)
	IL-picture (other layers of I-pictures)
P-picture:	PT-picture (lowest resolution layer)
	PL-picture (other layers of P-pictures)
B-picture:	BT-picture (lowest resolution layer)
	BL-picture (other layers of B-pictures)
T-picture	BT-picture (same type as lowest resolution layer of B-picture)

2.4.2 Macroblock Type

A macroblock consists of four sub-blocks (8x8) for the Y component and two sub-blocks for Cb and Cr components. It means that the macroblock structure is same as that of MPEG I.

As described above, IT-, PT- and BT-pictures have entirely same macroblock types of I-, P- and B-pictures of MPEG I. Macroblock types for other coding picture types are as follows:

```
  - Intra          : Intra mode
  - Up + NC        : Up-Sampled and Not Coded (skipped)
  - Up + Cd        : Up-Sampled and Coded
-------------------------------------- these are for IL-picture
  - Fx             : No Motion Compensation and Not Coded (skipped)
  - NM + Cd        : No Motion Compensation and Coded
  - MC + NC        : Motion compensated and Not Coded (Frwd Not Coded)
  - MC + Cd        : Motion compensated and Coded (Frwd Coded)
  - Up + Fx        : Up-Sampled, No MC and Not Coded
  - Up + NM + Cd   : Up-Sampled, No MC and Coded
  - Up + MC + NC   : Up-Sampled,    MC and Not Coded
  - Up + MC + Cd   : Up-Sampled,    MC and Coded
-------------------------------------- these are for PL-picture
  - IP + NC        : Interpolative prediction and Not Coded
  - IP + Cd        : Interpolative prediction and     Coded
  - BW + NC        : Backward prediction and Not Coded
```

```
  - BW + Cd       : Backward prediction and     Coded
  - Up + IP + NC : Up-Sampled, Interpolative prediction and not coded
  - Up + IP + Cd : Up-Sampled, Interpolative Prediction and     Coded
  - Up + BW + NC : Up-Sampled, Backward prediction and not Coded
  - Up + BW + Cd : Up-Sampled, Backward prediction and     Coded
---------------------------------------- these are for BL-picture
```

Basically, these macroblock types consists of all macroblock types of MPEG I syntax and those predicted with up-sampled pictures.

When up-sampled prediction is selected with other type or none, we predict the pixels by same placed pixels in the up-sampled picture of the lower resolution layer picture. Pixels in a macroblock can be predicted by pixels from three directions. When the prediction is done from more than 2 directions, each predicted pixel value is calculated by:

```
pel[][] = (pel_dir_1[][] + pel_dir_2[][]) // 2;                    (2 directions)
  or
pel[][] = (pel_dir_1[][] + pel_dir_2[][] + pel_dir_3[][]) // 3; (3 directions)
```

Where `pel[][]` is a value of predicted pixels, `pel_dir_1[][]`, `pel_dir_2[][]` and `pel_dir_3[][]` are values of pixel of retrieved macroblocks of up-sampled forward and backward pictures, and '//' means integer division with rounding to the nearest integer.

2.5 Transform

DCT is performed for each 8x8 sub-block. No modifications to MPEG I are made on transform in this proposed coding scheme.

2.6 Quantizer

Same quantizing matrices are used for the DCT coefficients as in MPEG I scheme. The matrices for an intra-picture and a predictive-picture are not modified.

2.7 Scan

After quantized, the same zig-zag scan to the coefficients is applied to generate one dimensional run. The run is coded in the same way of MPEG I scheme.

2.8 Up-sampling Filter

In order to generate up-sampled pictures at the encoder and the decoder side to differentiate lower and higher layers, the filter described in Figure 7 is applied.

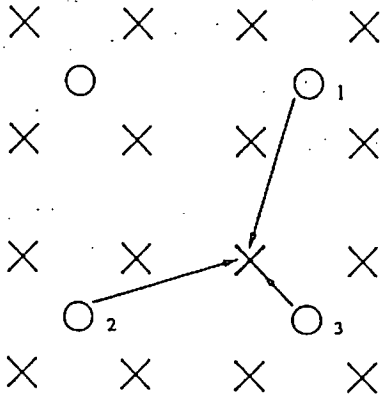


Figure 7: Up-sampling filter

Where "x" is to be up-sampled pixel, that is in the higher layer, "o" is the lower layer pixel. The value of the pixel "X" is obtained by the following equation:

$$X = (o_1 + 4 \times o_3 + o_2)//6 \quad (2)$$

Other pixels of the higher layer are calculated in the same way, respectively.

This filter generates enlarged pictures by twice without any phase distortion between a current layer picture and an up-sampled picture.

2.9 Down-sampling Filter

At encoder side, the filter described in Figure 8 is used to down-sample the higher resolution picture.

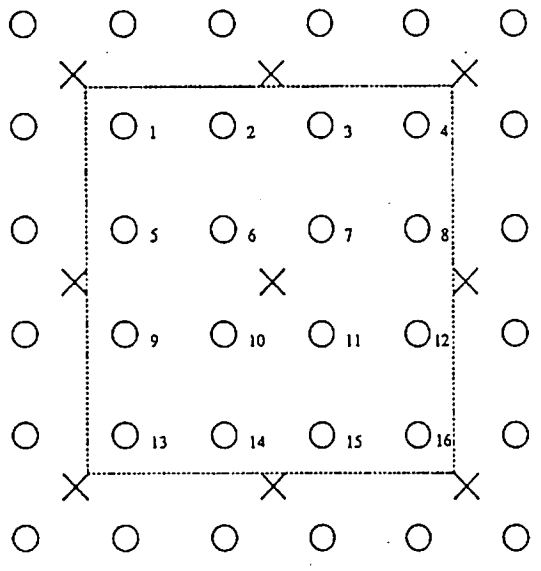


Figure 8: Down-sampling filter

Where "x" is to be down-sampled pixel, that is in the lower layer, "o" is the higher layer pixel. The value of the center pixel "x" is obtained by following equation:

$$X = \{(o_1+o_4+o_{13}+o_{16})-5(o_2+o_3+o_5+o_8+o_9+o_{12}+o_{14}+o_{15})+25(o_6+o_7+o_{10}+o_{11})\}//64 \quad (3)$$

Other pixels of the lower layer are calculated in the same way, respectively.

Phase distortion between a current layer picture and an up-sampled picture causes serious error for the hierarchical motion estimation. This filter generates reduced pictures by half without any phase distortion.

2.10 Rate Control

2.10.1 Picture Quantizer Value

To keep bit rate fixed, picture quantizer values (integer) are prepared for every layer picture of each picture type. These values are initialized at the start point of a sequence. They are updated using total bits of each type picture encoded most recently before encoding every layer picture. And half values of them are used as a sequence quantizer scale which is not to be changed through a layer picture.

Picture quantizer values are calculated as follows:

```
lr = l1 = l2 = l3 = 0;
for( i = 1; i <= 4; i++) {
    lr += lsr[i];
    l1 += ls1[i];
    if(i <= 1) l2 += ls2[i];
}
```

```
l3 = ls3[1];
r  = PictureRate / N;
rate  = BitRate / ( (lr + l1 * p + l2 * b + l3 * t) * r);
pstep[ptype][layer] = (int)(pstep[type][layer]/rate/2+.5)*2;
```

Where "lsr[i]", "ls1[i]", "ls2[i]" and "ls3[i]"(i=[1,4]) denote coded bits of most recently encoded layer pictures of I-picture, P-picture, B-picture with lower layers and B-picture without them respectively, "PictureRate" is picture rate per second and is 60, "N" denotes the number of pictures in a GOP (=24), "BitRate" is the defined bit rate.

And where "p"(=3), "b"(=8) and "t"(=12) indicate the number of P-picture, B-picture with lower layers and B-picture without them, included in a GOP.

"Pstep[ptype][layer]" is an integer value which indicates a picture quantizer value. In this array, "ptype" means a picture type (I,P,B,T-picture). "Pstep[type][layer]" must be clipped to the range [8, 62] or [4, 62] (for 4Mbps or 9Mbps, respectively) and conveyed for the next pstep calculation process of the same layer of the same type.

2.10.2 Macroblock Quantizer Value (mquant)

A mquant is calculated using squant and the standard deviation of the luminance of every input macroblock as follows:

```
q = (int)(sqrt((double)std_dev) * sq / 10) * 2;
if(q < (t=sq/3*2+2) )  q = t;
if(q > (t=sq*2) )      q = t;
if(mq-2 <= q && q <= mq+2 ) q = mq;
mq = q;
```

Where, "sq" is an integer value which equals twice of the sequence quantizer value, "q" is an integer value for calculation process, "std_dev" is an integer value which means the standard deviation of the luminance of the input macroblock, and "mq" is a macroblock quantizer value derived consequently. The "mq" value must be clipped to the range [8, 62] or [4, 62] (for 4Mbps or 9Mbps, respectively) and is to be conveyed for the next mquant calculation.

2.11 Completion of Partial Macroblock

If the horizontal_size_N (See 4.1) of each layer is not a multiple of 16, the encoding process shall extend the number of pixels per a line to complete the right-most macroblocks. Similarly, if the vertical_size_N (See 4.1) of each layer is not a multiple of 16, the encoding process shall extend the number of lines to complete the bottom-most macroblock. Any extended pixels and lines added by the encoder shall be removed by the decoder.

2.12 Decoding Process

The bit stream are divided suitable to appropriate layer decoders at first. Each layer decoder decodes its own pictures and pass the decoded pictures to a next higher layer decoder. In the layer decoder, a picture is reconstructed according to the macroblock type. After decoding all layers, each layer is transmitted to the post-process block.

2.13 Post-process

As the post-processing, following two features are required:

1. layer selection capability

2. possible non-interlace/interlace converter suitable to your display

As output from layer decoders, different sizes of pictures are available. It should be selected that which layer to be displayed. When the first layer is selected, a certain non-interlace/interlace converter algorithms is to be needed to display 59.94Hz non-interlaced video signal suitable to your

display. These kinds of features should not be standardized in order to be left as an implementation matter of the decoder.

In our implementation, even lines of a 59.94Hz even picture are taken as an even field, and odd lines of a 59.94Hz odd picture are taken as an odd field to be displayed on an interlaced TV set. The other fields are obtained respectively.

3 Functionality

3.1 Compatibility

According to Annex 1 and Figure 3 in MPEG91/100 Rev., our proposal scheme has one of the embedded compatibility. However, some parameters should be modified to be compatible to MPEG I scheme. The dimensions of lower pictures should be suitable to an input to MPEG I encoder. And number of layers may be limited such as 2, because lowest layer dimensions are already defined.

After modified above features, we think that our proposal is enough to be compatible with MPEG I coding scheme.

3.2 Random Access

Random access are available as to be in MPEG I scheme. After rate controlled, I and P pictures have enough data size to be displayed in required speed.

3.3 Encoding/Decoding Delay

Delay times for each block of the codec are shown in Table 2. In this table, it is assumed that process times such as transform, quantizer and etc. are nearly ideal, enough to execute in real time.

Table 2: Encoding/Decoding Delay

Block	Delay (msec)	Comments
Pre-process	17	1 field delay
Motion Vector Estimation	negligible	
Down-sampling Filter	negligible	
DCT	negligible	
Quantizer	negligible	
Predictor	negligible	
VLC	negligible	
Encoder buffer	84	5 picture delay
Transmission (encoder+decoder)	108 (4M)	$((300K + 200K - 4M/60)/4M)$
	105 (9M)	$((700K + 400K - 9M/60)/9M)$
Decoder buffer	34	2 picture delay
VLD	negligible	
Dequantizer	negligible	
IDCT	negligible	
Post-process	17	1 field delay
Total	260 (4M) + α	
	257 (9M) + α	

3.4 Layer Selection

As described in section 2.13, the decoder selects which layer should be displayed. This function is performed just to choose an appropriate layer and dispose other layers at the post-process block. It depends on the post-process capability how to display obtained layer pictures.

3.5 Layered Coding

Our proposed coding scheme is a kind of layered coding so that high priority should be applied to lower layers, that are reduced layer pictures.

Particularly in the ATM environment, coding data of the lower layer should be transmitted in high priority cells. When the cell loss is occurred, one of the possible recovering method is to reconstruct a lost block of the higher layer picture as an up-sampled block of the lower layer picture. This is too simple method so that error propagation may be serious until next I-picture comes.

4 Bitstream Syntax

In our proposal, the bitstream consists of video sequence layers, group of pictures layers, picture layers, slice layers, macroblock layers and block layers. The syntax of each layer, except the sequence header and GOP layer, is almost same as CD 11172. The overview of syntax is shown in Figure 9. And the syntax diagram is shown in Figure 10. All mnemonics in Figure 10 are described in Table 3.

In this document, some different points are shown in following subsections.

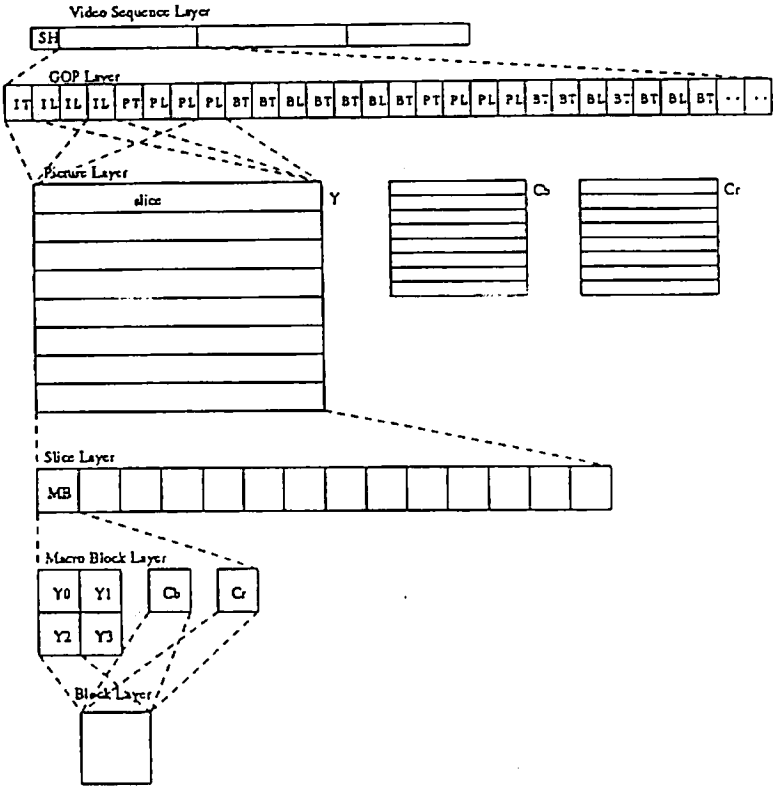


Figure 9: Overview of Syntax

4.1 Sequence Header

We modify the syntax of the sequence header portion as shown in Figure 10 and 11.

In order to describe the number of layers in each type of pictures, we use first 8 bits of sequence_extension_data in sequence header. These 8 bits consists of number_of_layers_code_I, number_of_layers_code_P, number_of_layers_code_B and number_of_layers_code_T, that identify the number of layers of I-, P-, B- and T-pictures respectively. The code for the number of layers has 2 bits each. The numbers of layers, number_of_layer_I, number_of_layer_P, number_of_layer_B and number_of_layer_T, are derived from the FLC table as shown in Table 4.

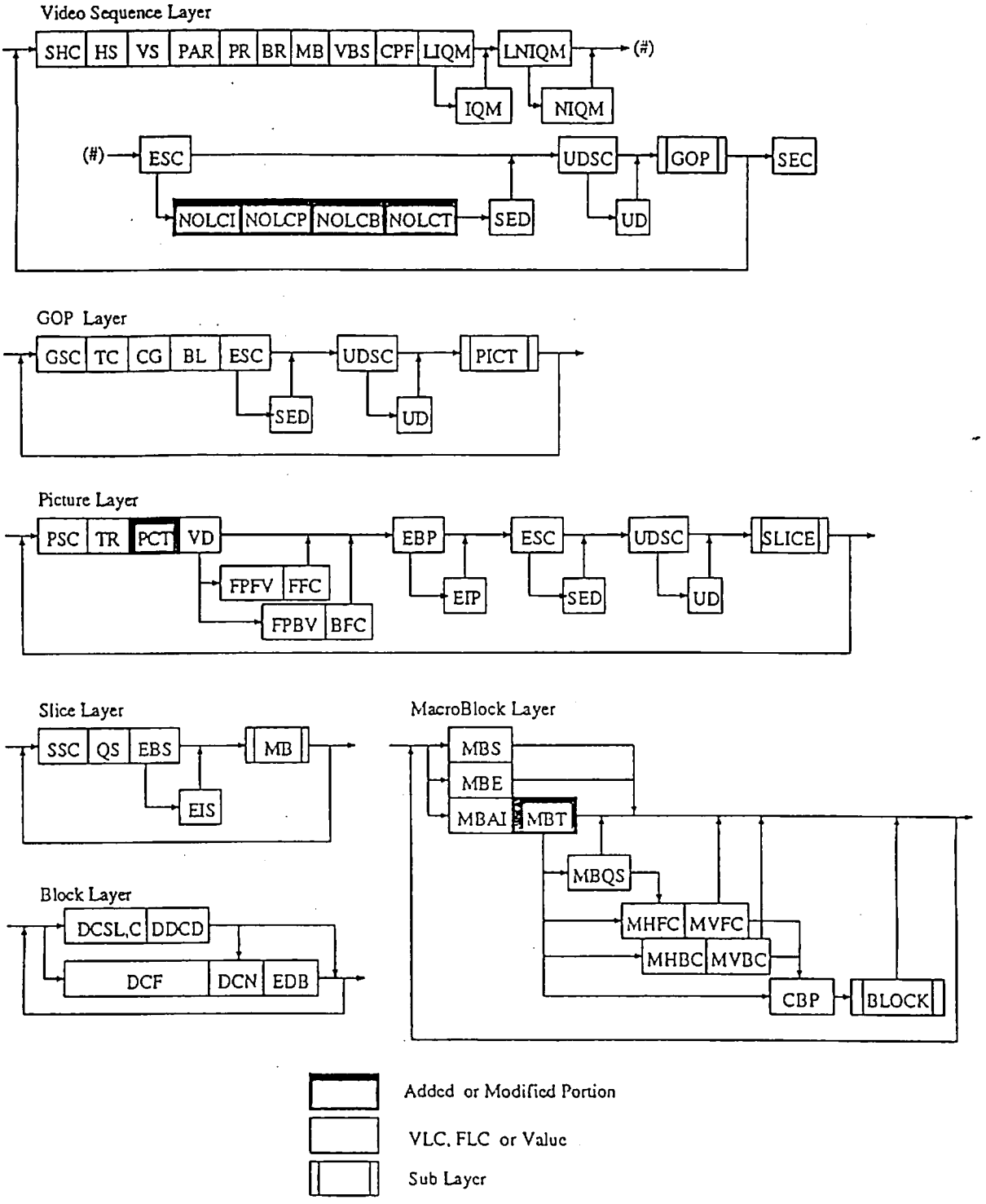


Figure 10: Syntax Diagram

Table 3: Mnemonics for describing Syntax Diagram

• Video Sequence Layer:					
SHC	sequence_header_code	HS	horizontal_size	VS	vertical_size
PAR	picture_rate	BR	bit_rate	MB	marker_bit
VBS	vbv_buffer_size	CPF	constrained_parameter_flag	LIQM	load_intra_quantizer_matrix
IQM	intra_quantizer_matrix	LNIQM	load_non_intra_qntzr._matrix	NIQM	non_intra_qntzr._matrix
ESC	extension_start_code	SED	sequence_extension_data	NOLCI	number_of_lines_code_I
NOLCP	number_of_lines_code_P	NOLCB	number_of_lines_code_B	NOLCT	number_of_lines_code_T
UDSC	user_data_start_code	UD	user_data	GOP	group of picture
SEC	sequence_end_code				
• Group of Picture Layer:					
GSC	group_start_code	TC	time_code	CG	closed_gop
BL	broken_link	PICT	picture		
• Picture Layer:					
PSC	picture_start_code	TR	temporal_reference	PCT	picture_coding_type
VD	vbv_delay	FPFV	full_pel_forward_vector	FFC	forward_f_code
FPBV	full_pel_backward_vector	BFC	backward_f_code	EBP	extra_bit_picture
EIP	extra_information_picture	SLICE	slice		
• Slice Layer:					
SSC	slice_start_code	QS	quantizer_scale	EBS	extra_bit_slice
EIS	extra_information_slice	MB	macroblock		
• MacroBlock Layer:					
MBS	macroblock_stuffing	MBE	macroblock_escape	MBAI	macroblock_address_increment
MBT	macroblock_type	MBQS	quantizer_scale for MB	CBP	coded_block_pattern
MHFC	motion_horizontal_forwd._code	MVFC	motion_vertical_forwd._code	MHBC	motion_horizontal_ backwd._code
MVBC	motion_vertical_backwd._code	BLOCK	sub-block		
• Block Layer:					
DCSL,C	dct_dc_size_lumi.,crom.	DDCD	dct_dc_differential	DCF	dct_coeff_first
DCN	dct_coeff_next	EDB	end_of_block		

Table 4: FLC table for number of layer code

number_of_layer_code	number of layers
00	1
01	2
10	3
11	4

In case that all types of pictures have only one layer (identical to MPEG I), sequence_start_code, number_of_layers_code_I, number_of_layers_code_P, number_of_layers_code_B and number_of_layers_code_T is not inserted in the bitstream to match it to MPEG I syntax.

In our proposal, the size of each layer pictures should be derived at the receiver side. The sizes are calculated from horizontal_size and vertical_size. The horizontal sizes of N-th layer picture (when highest resolution is 1st layer), horizontal_size_N, is horizontal_size >> (N−1) . And the vertical size of N-th layer picture, vertical_size_N, is vertical_size >> (N−1).

4.2 Group of Pictures Layer

A GOP has all layer pictures. But the syntax of the GOP layer in our proposal is almost same as CD 11172.

4.3 Picture Layer

The syntax of the picture layer is identical to that of MPEG I except for picture_coding_type. Though, in our proposal, the number of bits (3 bits) for it is same as MPEG I, several coding methods are added to the VLC table for picture_coding_type as shown in Table 5.

The syntax diagram of the picture layer is shown in Figure 10.

```
sequence_header() {
    sequence_header_code                                    32      bslbf
    horizontal_size                                         12      uimsbf
    vertical_size                                           12      uimsbf
    pel_aspect_ratio                                        4       uimsbf
    picture_rate                                            4       uimsbf
    bit_rate                                                18      uimsbf
    marker_bit                                              1       "1"
    vbv_buffer_size                                         10      uimsbf
    constrained_parameter_flag                              1
    load_intra_quantizer_matrix                             1
    if ( load_intra_quantizer_matrix )
      intra_quantizer_matrix[64]                            8*64    uimsbf
    load_non_intra_quantizer_matrix                         1
    if ( load_non_intra_quantizer_matrix )
      non_intra_quantizer_matrix[64]                        8*64    uimsbf
    next_start_code()
    if (nextbits() == extension_start_code ) {
      extension_start_code                                  32      bslbf
      number_of_layers_code_I                               2       bslbf
      number_of_layers_code_P                               2       bslbf
      number_of_layers_code_B                               2       bslbf
      number_of_layers_code_T                               2       bslbf
      while ( nextbits () != '0000 0000 0000 0000 0000 0001' ) {
        sequence_extension_data                             8
      }
      next_start_code()
    }
    if (nextbits() == user_data_start_code ) {
      user_data_start_code                                  32      bslbf
      while ( nextbits() != '0000 0000 0000 0000 0000 0001' ) {
        user_data                                           8
      }
      next_start_code()
    }
}
```

Figure 11: Sequence Header

Table 5: FLC table for picture coding type

picture_coding_type	coding method
000	forbidden
001	intra-coded(IT*)
010	predictive-coded(PT*)
011	bidirectionally-predictive(BT*)
100	dc intra-coded (D)
101	hierarchically and intra (IL*)
110	hierarchically and predictive (PL*)
111	hierarchically and bidirectionally (BL*)

* These picture coding types are described in 2.4.1.

4.4 Slice Layer

The syntax of the slice layer is almost same as MPEG I. The syntax diagram of the slice layer is shown in Figure 10.

4.5 Macroblock Layer

Though the VLC tables for macroblock_type of IT-, PT- and BT-pictures are identical to those of I-, P- and B-pictures in MPEG CD, three VLC tables for macroblock_type of IL-, PL- and BL-pictures are added. In our proposal, all VLC tables for macroblock_type are shown in Table 6–11.

The syntax diagram of the macroblock layer is shown in Figure 10.

4.6 Block Layer

The syntax of the block layer is almost same as MPEG I. The syntax diagram of the block layer is shown in Figure 10.

5 Implementation Study

5.1 Picture Buffer

5.1.1 Motion Vector Estimator

In this module, two neighbor layers are needed for each picture. The picture buffer size should be 5,068,800 bits.

5.1.2 Encoder

For the sufficient input pictures kept in the input buffer, the number of B-pictures is concerned. In our implementation, this is five so that the input buffer size should be 20,275,200 bits.

In the layer encoder, three sets of the hierarchical picture buffer are enough to perform coding. The picture buffer size in the layer encoder should be 16,156,800 bits.

5.1.3 Coding Buffer at Encoder

At 4 Mbps, the coding buffer needs 300 kbits with ±10% width. At 9 Mbps, it needs 700 kbits with ±10% width.

5.1.4 Decoder

In the layer decoders, three sets of the hierarchical layer picture buffer are enough. The picture buffer size is the layer decoders should be 16,156,800 bits.

5.1.5 Display Buffer

The post-process has a choice which layers, not only one layer but several layers if needed, should be displayed. If one choice is only allowed for it, 4055040 bits should be needed as the display buffer. However, several choices are allowed for it, 5385600 bits should be needed as the display buffer.

Table 12 summarizes above values.

5.2 Implementation for Each Module

5.2.1 Encoder

The encoder consists of modules as described in Table 13. This table shows some implementation features. In calculating each items, we estimate each module of the simulation program written in C-language by using UNIX command "gprof" and by hand.

Since the coding syntax of these modules is almost same as MPEG I in the encoder, Forward and Inverse DCT, Quantizer, Dequantizer and VLC modules are to be applied as they are in MPEG I.

Table 6: VLC table for MB type in IT-pictures

VLC code	MB quant
1	0
01	1

Table 7: VLC table for MB type in IL-pictures

VLC code	MB quant	MB up-sampled	MB pattern	MB intra
none(skipped)	0	1	0	0
1	0	1	1	0
001	0	0	0	1
01	1	1	1	0
0001	1	0	0	1

Table 8: VLC table for MB type in PT-pictures

VLC code	MB quant	MB forward	MB pattern	MB intra
1	0	1	1	0
01	0	0	1	0
001	0	1	0	0
00011	0	0	0	1
00010	1	1	1	0
00001	1	0	1	0
000001	1	0	0	1

Table 9: VLC table for MB type in PL-pictures

VLC code	MB quant	MB up-sampled	MB forward	MB pattern	MB intra
none(skipped)	0	0	0	0	0
10	0	1	1	1	0
010	0	0	1	1	0
0010	0	0	1	0	0
00011	0	1	1	0	0
00010	0	1	0	1	0
000011	0	1	0	0	0
000010	0	0	0	0	1
11	1	1	1	1	0
010	1	0	1	1	0
0011	1	1	0	1	0
000001	1	0	0	0	1

Table 10: VLC table for MB type in BT-pictures

VLC code	MB quant	MB forward	MB backward	MB pattern	MB intra
10	0	1	1	0	0
11	0	1	1	1	0
010	0	0	1	0	0
011	0	0	1	1	0
0010	0	1	0	0	0
0011	0	1	0	1	0
00011	0	0	0	0	1
00010	1	1	1	1	0
000011	1	1	0	1	0
000010	1	0	1	1	0
000001	1	0	0	0	1

Table 11: VLC table for MB type in BL-pictures

VLC code	MB quant	MB up-smpld.	MB frwd	MB bkwd	MB pattern	MB intra
10	0	0	1	1	0	0
11	0	0	1	1	1	0
010	0	1	1	1	0	0
011	0	1	1	1	1	0
0010	0	0	1	0	0	0
0011	0	0	1	0	1	0
00010	0	0	0	1	0	0
00011	0	0	0	1	1	0
000010	0	1	0	1	0	0
000011	0	1	0	1	1	0
0000010	0	1	1	0	0	0
0000011	0	1	1	0	1	0
00000010	0	1	0	0	0	0
00000011	0	1	0	0	1	0
000000010	0	0	0	0	0	1
000000011	1	1	1	1	1	0
0000000010	1	0	1	1	1	0
0000000011	1	1	0	1	1	0
00000000010	1	1	1	0	1	0
00000000011	1	0	1	0	1	0
000000000010	1	0	0	1	1	0
000000000011	1	1	0	0	1	0
000000000001	1	0	0	0	0	1

Table 12: Summary of picture buffers

Picture buffer name	Buffer size (bits)	Width	Comments
Motion estimator	5,068,800		Buf(L1) + Buf(L2)
Input buffer at enc.	20,275,200		Buf(L1)× 5
Layer encoders	16,156,800		(Buf(L1)+Buf(L2)+Buf(L3)+Buf(L4))× 3
Coding buffer at enc.	300,000 (at 4Mbps)	±10%	
	700,000 (at 9Mbps)	±10%	
Layer decoder	16,156,800		(Buf(L1)+Buf(L2)+Buf(L3)+Buf(L4))× 3
Display buffer	4,055,040 (min)		Buf(L1)
	5,385,600 (max)		Buf(L1)+Buf(L2)+Buf(L3)+Buf(L4)

Buf(L1): first layer picture buffer size (704 × 480 × 8 × (1 + 1/4 + 1/4))
Buf(L2): second layer picture buffer size (352 × 240 × 8 × (1 + 1/4 + 1/4))
Buf(L3): third layer picture buffer size (176 × 120 × 8 × (1 + 1/4 + 1/4))
Buf(L4): forth layer picture buffer size (88 × 60 × 8 × (1 + 1/4 + 1/4))

Table 13: implementation features for each module of Encoder

Module	Sz. mem. (byte)	Mem. Bnd. (k times/s)	NO. add. (k times/s)	NO. mul. (k times/s)	Table Sz. (N.O.entry)	NO. tbl LU (k times/s)
Down-sampling filter	16	107,038	100,742	18,889	-	-
Up-sampling filter	16	100,742	75,568	50,371	-	-
Motion estimator	4776	6,881,871	5,750,359	442,417	(fixed) 130	20,275
Forward DCT	832	694,666	590,466	556,275	(fixed) 64	555,733
Inverse DCT	832	309,066	192,815	192,815	(fixed) 64	192,815
Quantizer & Scan	940	207,576	208,580	104,742	(fixed) 608 (loaded) 128	1,804 34,733
Dequantizer	832	34,733	69,466	104,199	(loaded) 128	34,733
Mode controller & Predictor	7680	1,215,368	864,805	305,800	(fixed) 114 (loaded) 58	270
Q controller	240	46,311	69,829	272	-	-
Total	16164	8,215,237	8,465,862	2,614,692	1294	

Sz. mem. : size of memory
Mem. Bnd. : number of access instead of memory bandwidth(± 10%)
NO. add. : number of additions (± 10%)
NO. mul. : number of multiplications(± 10%)
Table Sz. : table sizes
NO. tbl LU. : number of table lookups(± 10%)

5.2.2 Decoder

The decoder consists of modules as described in Table 14. We estimate each module of the simulation program in same way, as estimating modules of the encoder,

Table 14: Implementation features for each module of Encoder

Module	Sz. mem. (byte)	Mem. Bnd. (k times/s)	NO. add. (K times/s)	NO. mul. (K times/s)	Table Sz. (N.O.entry)	NO. tbl LU (K times/s)
Up-sampling filter	16	100,742	75,568	50,371	–	–
Inverse DCT	832	309,066	227,548	193,357	(fixed) 64	192,815
Dequantizer	832	23,010	32,878	16,439	(loaded) 128	23,010
Predictor	1280	723,629	527,093	142,188	(loaded) 58	90,220
Total	2960	1,156,448	863,089	402,357	250	215,915

Since the syntax of this proposal is almost same as MPEG I in the decoder, Inverse DCT, Dequantizer and VLC decoder modules are to be applied as they are in MPEG I.

5.3 Global

In this coding method, number of layers is one of the most important parameters to suit applications. To suit the video transmission through ATM-based network, two layers at least are to be employed. The lower resolution layer is packetized and transmitted in high priority cells and the higher resolution layer is transmitted in low priority cells. The number of layers can also be set to one if needed. In this case, the decoding procedures and the bit stream syntax are completely identical to those defined in MPEG I.

It is not to be mentioned that the parameters, the number of pictures in a GOP, the picture size , the I- and P-picture interval (M) and etc., which are also defined in MPEG I, can be altered.

5.4 Automatic Encoder

The pictures we submit to the Kurihama tests are generated almost automatically. But only one non-automatically adjusted coding parameter is a set of initial values of the picture quantizer value. It is adjusted after a few simulations such that the total coded bits can be of the nearest amount and under defined bits as much as possible.

The set of initial values of the picture quantizer value is adjusted by hand in generating the submitted pictures. However, pictures are automatically generated with decreasing encoded bits and SNR slightly if using the set of default values.

6 Statistics

6.1 Bit Amount and SNR

The amount of coded bits and SNR for each picture are shown in Figure 12–25. And the amount of coded bits for each GOP are shown in Table 15–21. The amount of coded bits is the sum of bits of all layers in every picture. In order to evaluate the Coder/Decoder of itself, the SNR is calculated between the first layer picture after the pre-process in the encoder and that before the post-process in the decoder.

6.2 Coded Bit Stream Files

Coded bit stream files for each sequence are shown in Table 22.

6.3 Other Items

Some other items averaged over each sequence are shown in Table 23–29.

7 Conclusion

We propose "Hierarchical Video Signal Coding" for MPEG II coding scheme.

Our proposal is to be characterized on following points:

1. Non-interlaced pictures for each layer picture, even for the first layer.
2. Hierarchical motion estimation to provide precise motion vectors in less process time, to each layer of pictures.
3. Hierarchical prediction and layer pictures to provide various size of video signal for display, and to be compatible with MPEG I.
4. Up-sampling and down-sampling filters without phase distortion between neighbor layers.

Other functions are almost same as MPEG I.

As the experimental result, the quality of decoded pictures highly depends on the quality of non-interlaced pictures that are provided from the pre-process.

For the future discussion items, we think that video coding should not depend on the interlaced signal only, such as NTSC and PAL. In coming multimedia communication, such as workstation environment, the fusion of ordinary TV and multimedia terminal will be needed. In this sense, we think that MPEG II should not be specialized for interlaced pictures only.

References

[1] Feng-Ming WANG, Dimitris Anastassiou and Arun N. Netravali: "Time-recursive deinterlacing for IDTV and pyramid coding", Signal processing Image Communication Vol.2 No.3 pp365-374 (1990)

[2] H.Tominaga, N.Komatsu, T.Miyashita and T.Hanamura: "A Motion Detection Method on Video Image by using Hierarchical Pixels", Trans.(D) IEICE of Japan (Mar.1989)

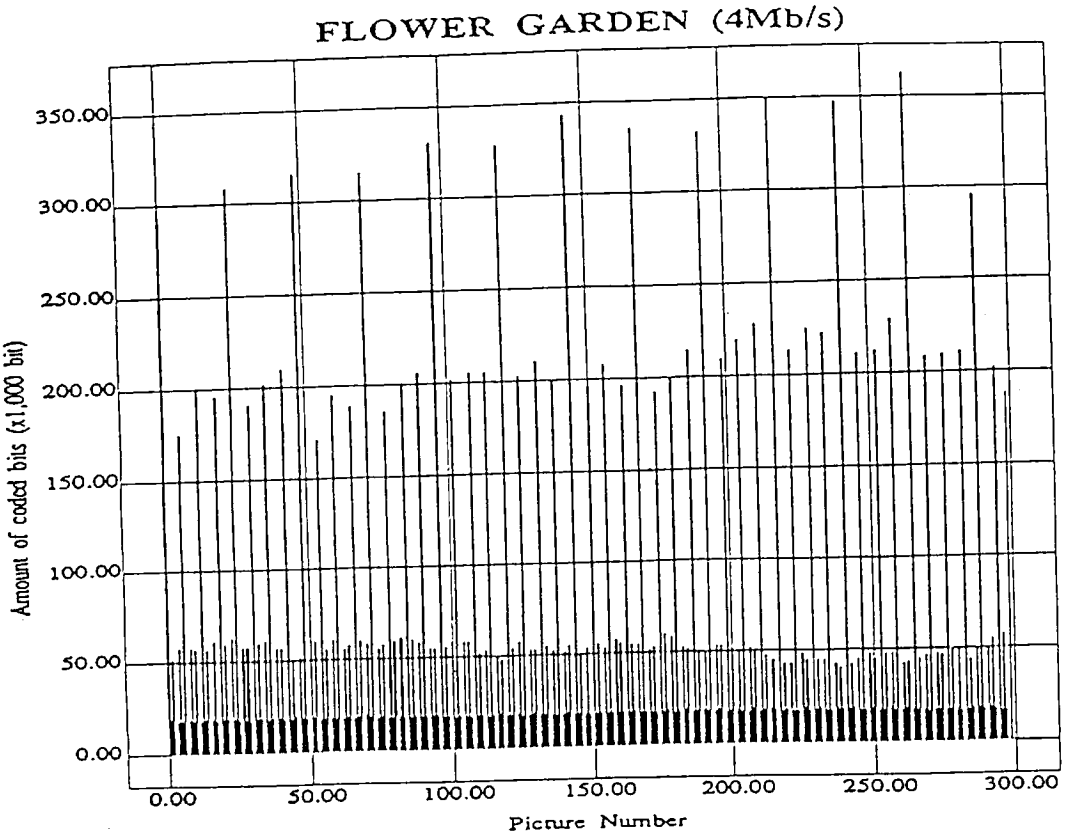


Figure 12: Amount of Coded Bits (1)

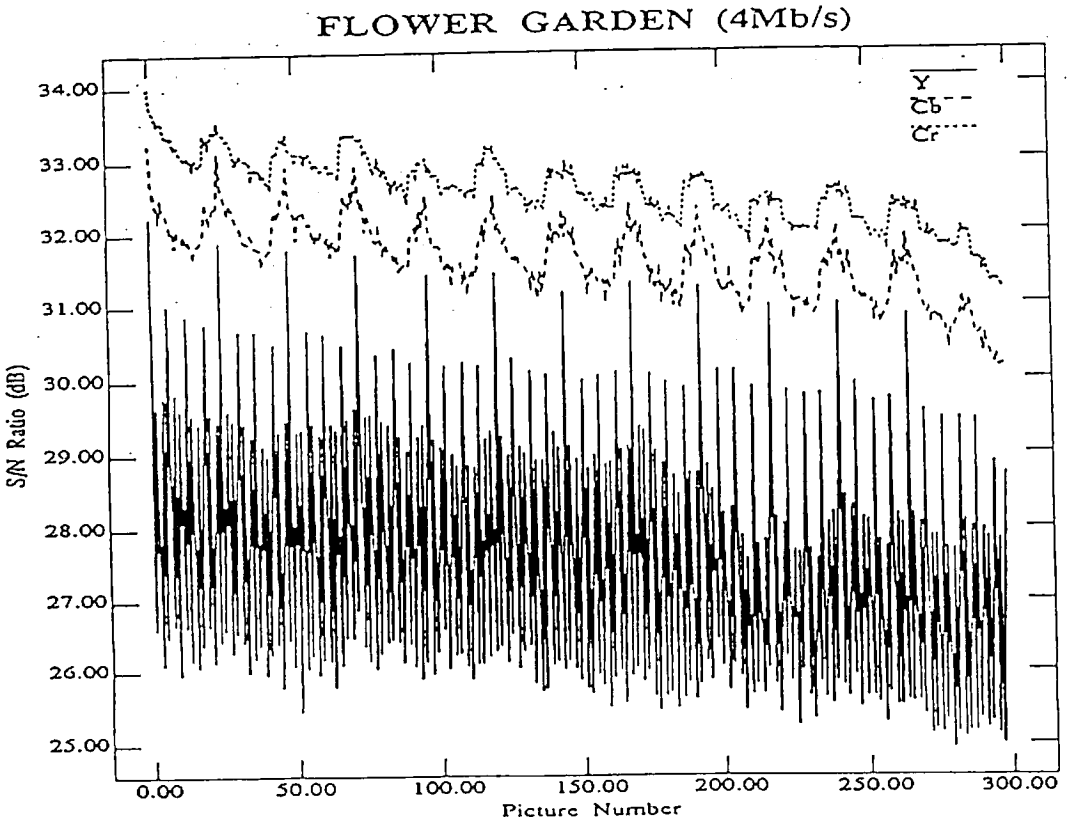


Figure 13: S/N Ratio (1)

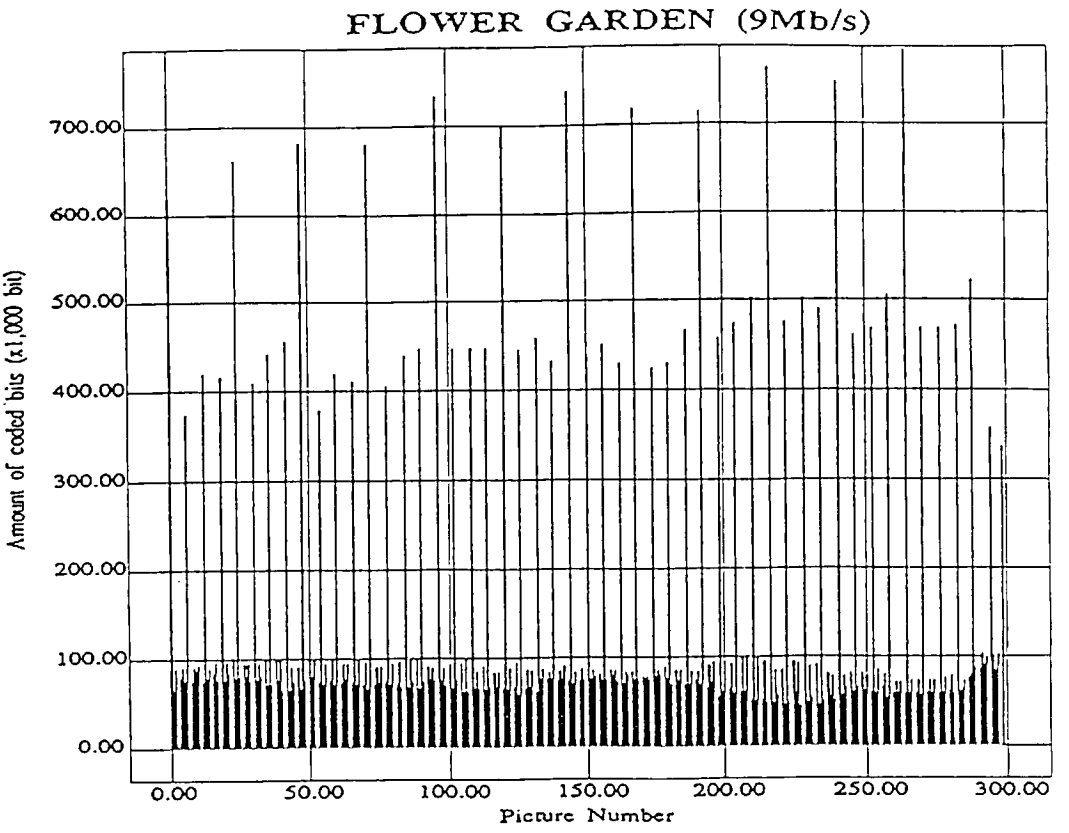


Figure 14: Amount of Coded Bits (2)

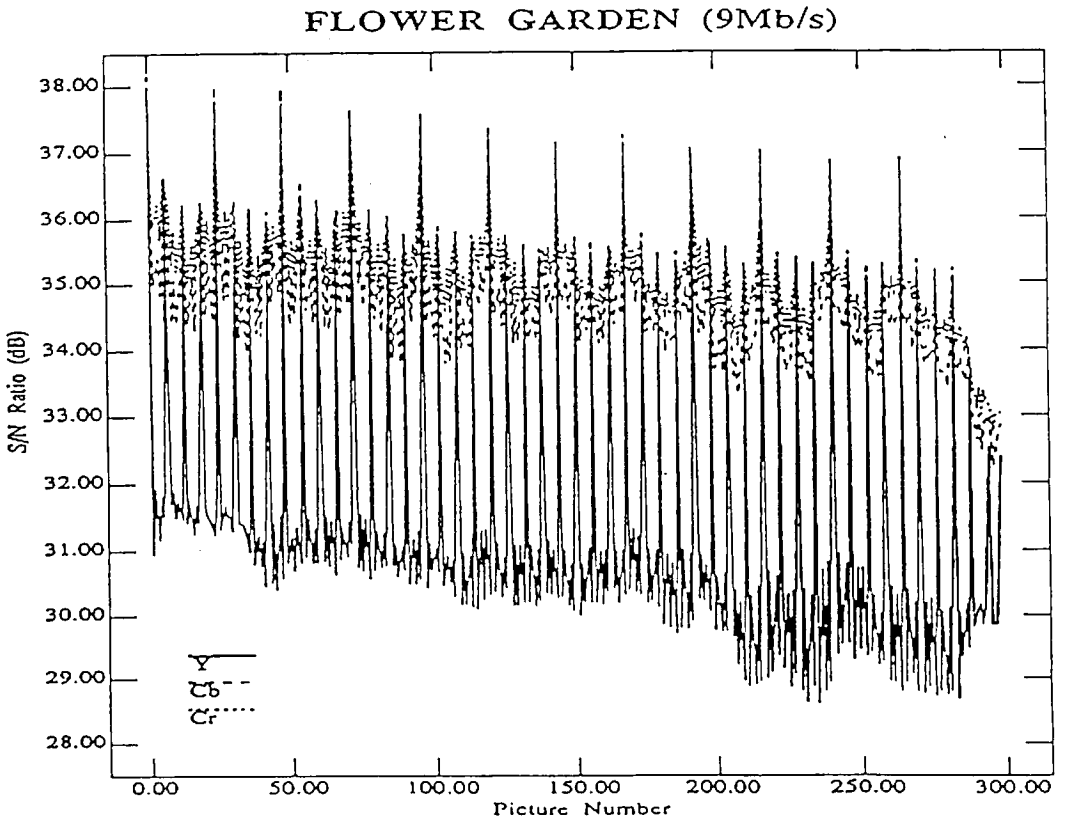


Figure 15: S/N Ratio (2)

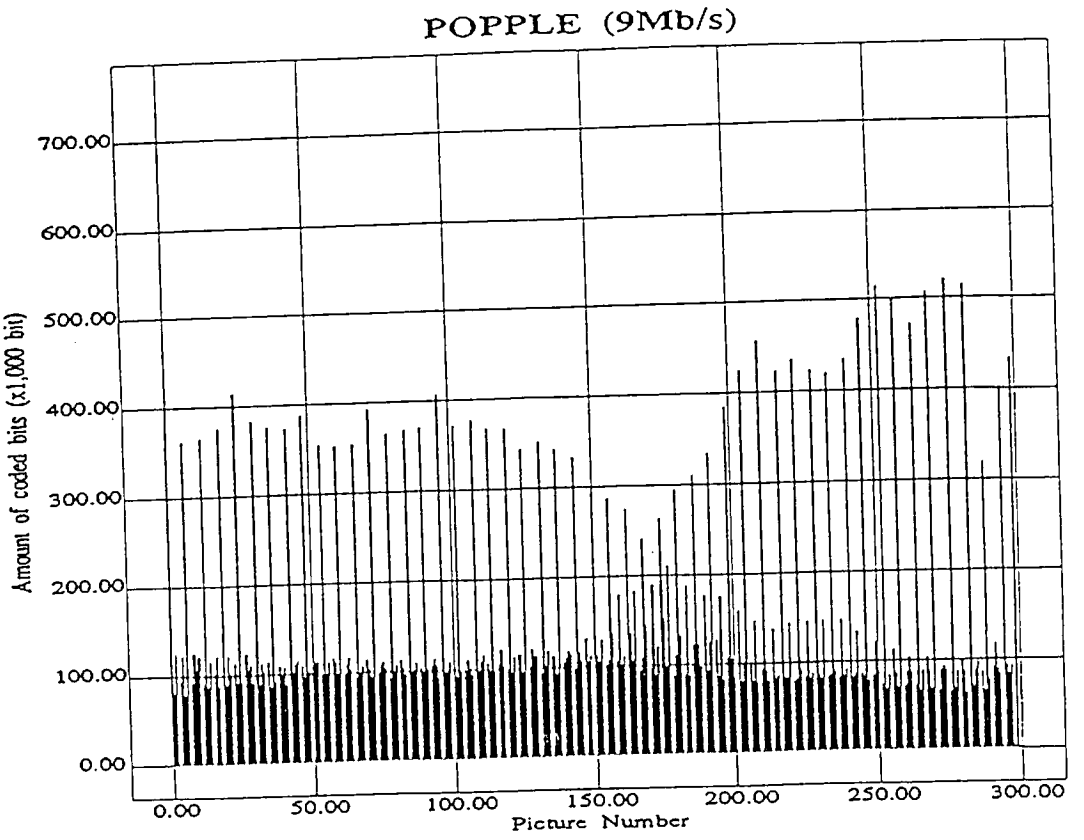


Figure 16: Amount of Coded Bits (3)

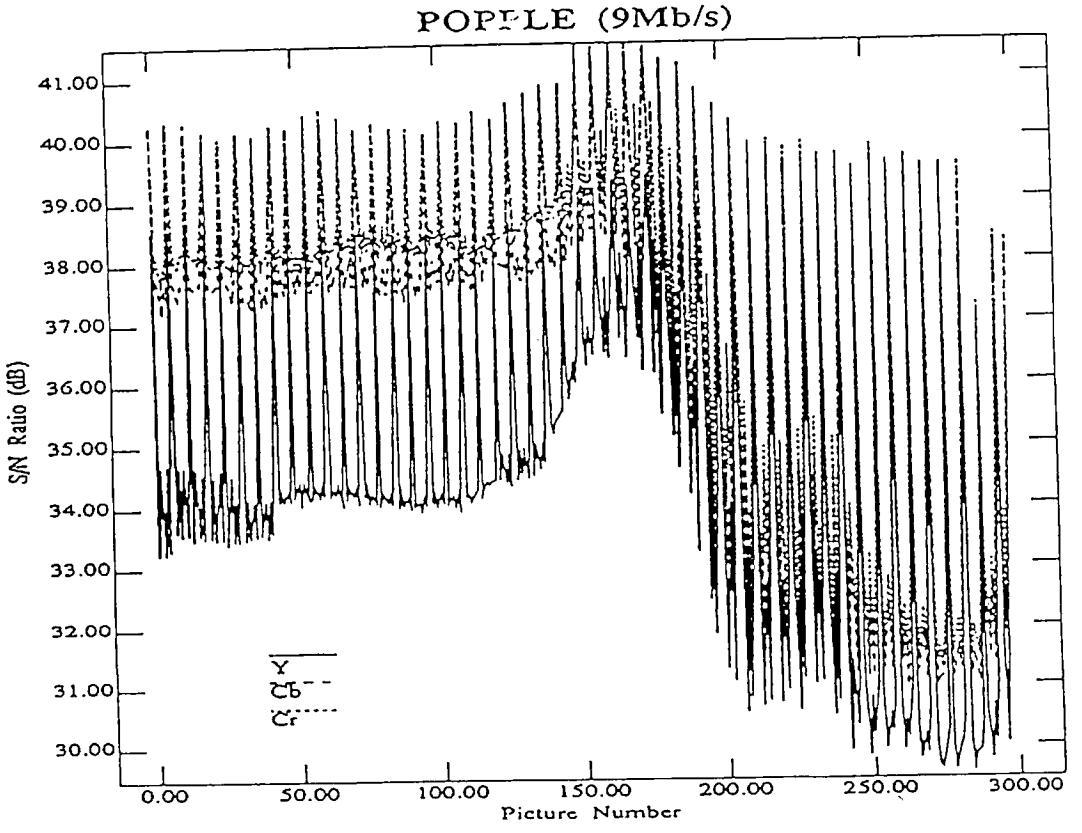


Figure 17: S/N Ratio (3)

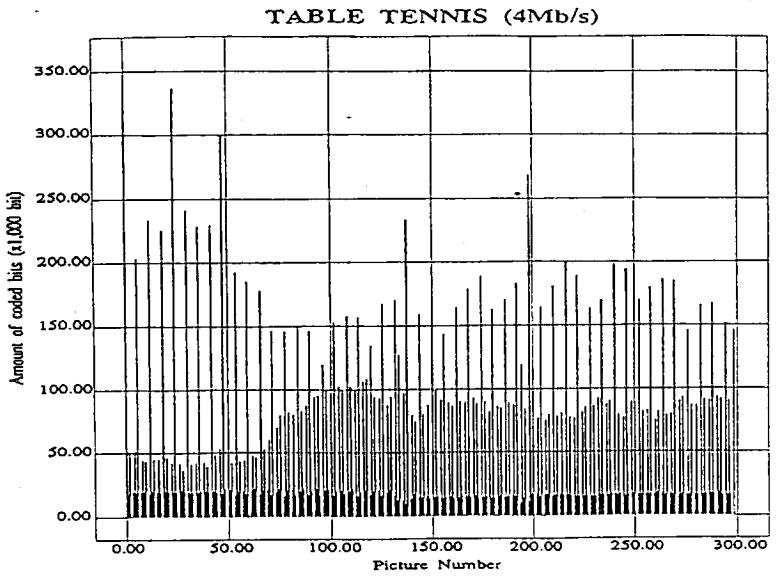


Figure 18: Amount of Coded Bits (4)

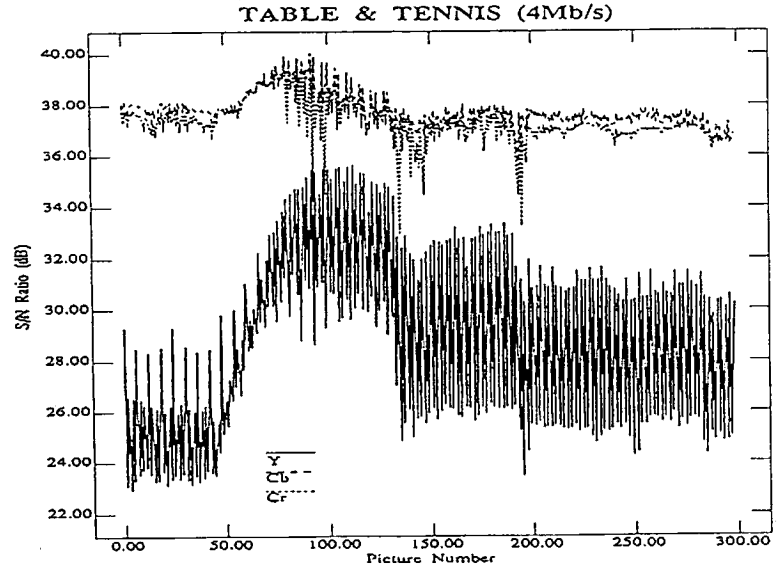


Figure 19: S/N Ratio (4)

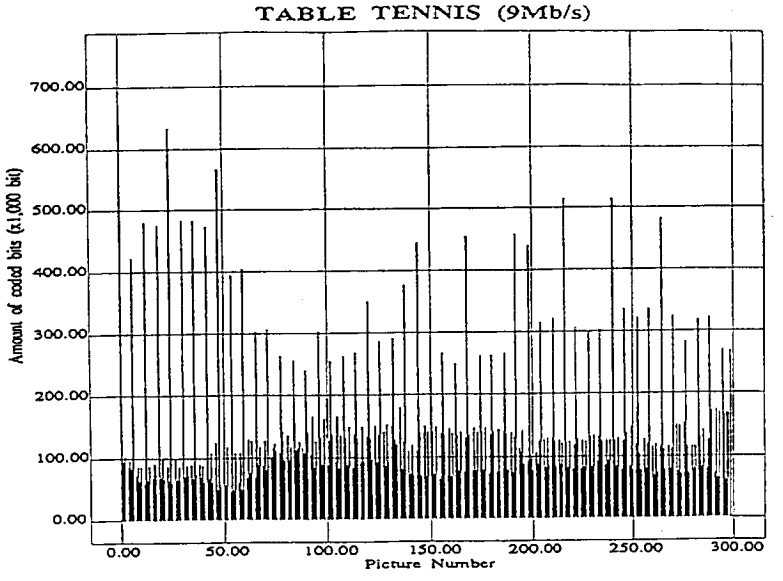


Figure 20: Amount of Coded Bits (5)

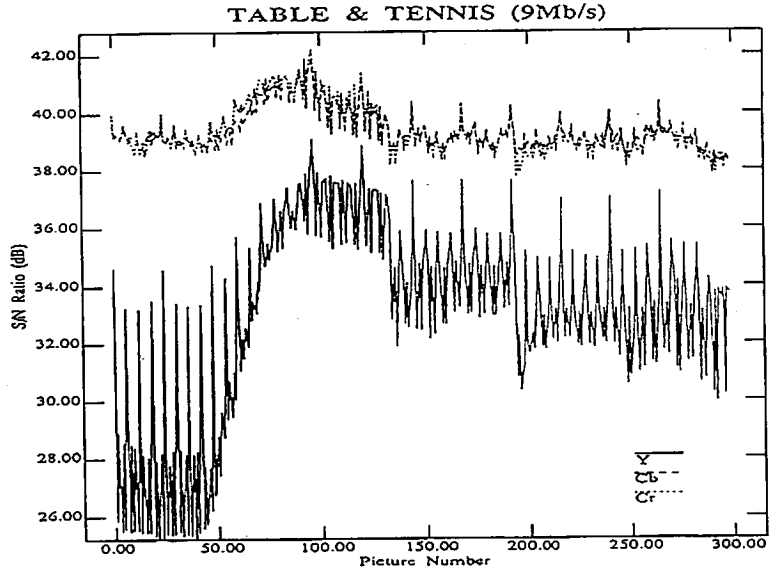


Figure 21: S/N Ratio (5)

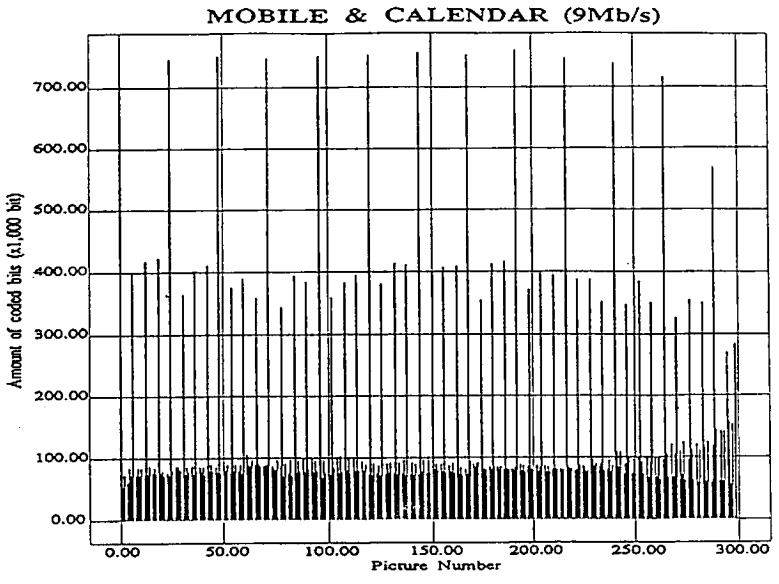


Figure 24: Amount of Coded Bits (7)

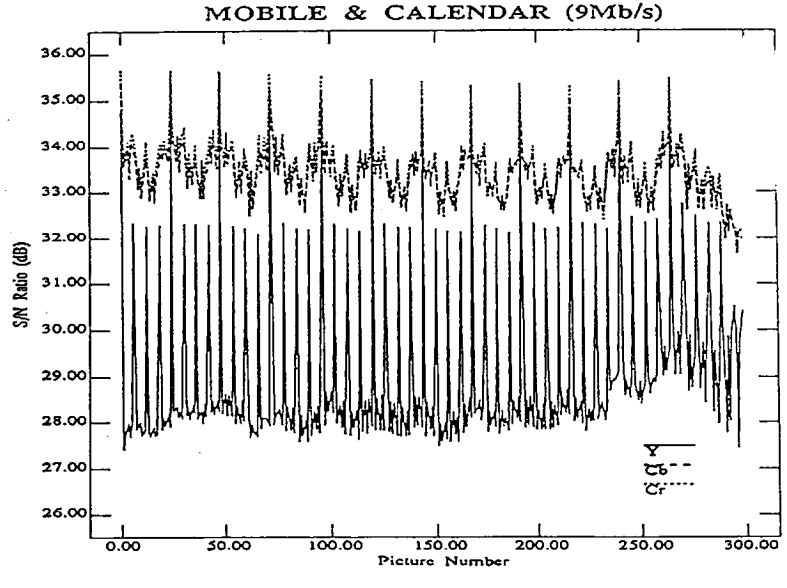


Figure 25: S/N Ratio (7)

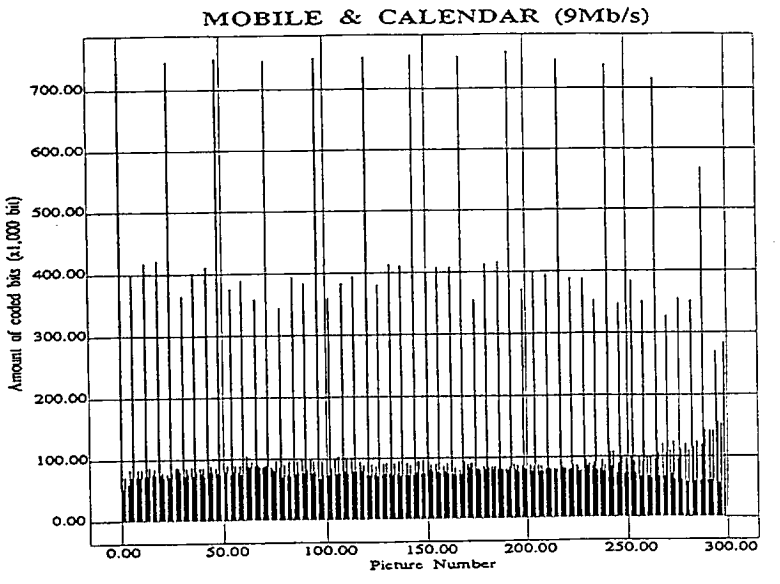


Figure 22: Amount of Coded Bits (6)

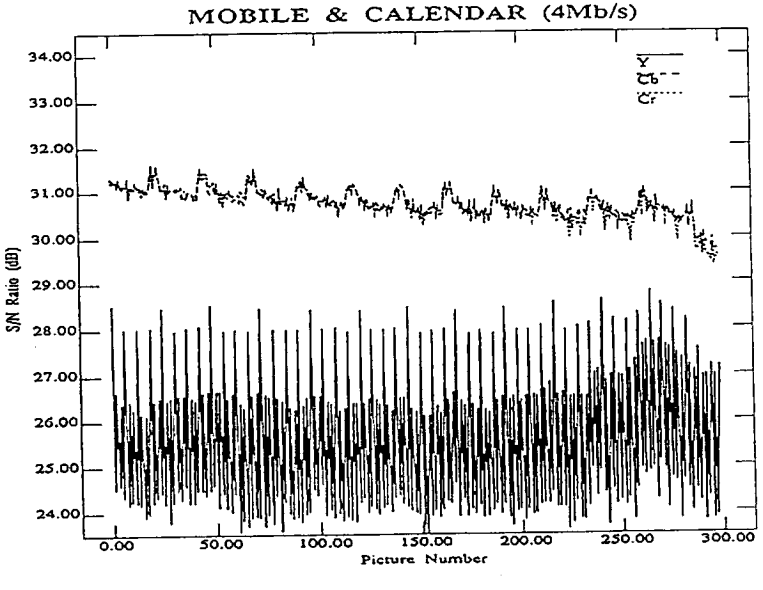


Figure 23: S/N Ratio (6)

Table 15: Bit amount for each GOP (1) "flower garden" (4Mb/s)

GOP No.= 1	1375206	(bits)	of 19	(pictures)
GOP No.= 2	1610176	(bits)	of 24	(pictures)
GOP No.= 3	1551464	(bits)	of 24	(pictures)
GOP No.= 4	1606428	(bits)	of 24	(pictures)
GOP No.= 5	1611799	(bits)	of 24	(pictures)
GOP No.= 6	1586462	(bits)	of 24	(pictures)
GOP No.= 7	1599496	(bits)	of 24	(pictures)
GOP No.= 8	1602579	(bits)	of 24	(pictures)
GOP No.= 9	1623564	(bits)	of 24	(pictures)
GOP No.=10	1592253	(bits)	of 24	(pictures)
GOP No.=11	1577332	(bits)	of 24	(pictures)
GOP No.=12	1573640	(bits)	of 24	(pictures)
GOP No.=13	1084951	(bits)	of 16	(pictures)
total bits =	19995350	(bits)	of 299	(pictures)

Table 16: Bit amount for each GOP (2) "flower garden" (9Mb/s)

GOP No.= 1	3079006	(bits)	of 19	(pictures)
GOP No.= 2	3656682	(bits)	of 24	(pictures)
GOP No.= 3	3530381	(bits)	of 24	(pictures)
GOP No.= 4	3585226	(bits)	of 24	(pictures)
GOP No.= 5	3646637	(bits)	of 24	(pictures)
GOP No.= 6	3555407	(bits)	of 24	(pictures)
GOP No.= 7	3660302	(bits)	of 24	(pictures)
GOP No.= 8	3615557	(bits)	of 24	(pictures)
GOP No.= 9	3655416	(bits)	of 24	(pictures)
GOP No.=10	3569185	(bits)	of 24	(pictures)
GOP No.=11	3546155	(bits)	of 24	(pictures)
GOP No.=12	3499776	(bits)	of 24	(pictures)
GOP No.=13	2330682	(bits)	of 16	(pictures)
total bits =	44930412	(bits)	of 299	(pictures)

Table 17: Bit amount for each GOP (3) "popple" (9Mb/s)

GOP No.= 1	3040977	(bits)	of 19	(pictures)
GOP No.= 2	3582241	(bits)	of 24	(pictures)
GOP No.= 3	3570858	(bits)	of 24	(pictures)
GOP No.= 4	3552877	(bits)	of 24	(pictures)
GOP No.= 5	3572056	(bits)	of 24	(pictures)
GOP No.= 6	3553297	(bits)	of 24	(pictures)
GOP No.= 7	3493801	(bits)	of 24	(pictures)
GOP No.= 8	3655829	(bits)	of 24	(pictures)
GOP No.= 9	3829703	(bits)	of 24	(pictures)
GOP No.=10	3627294	(bits)	of 24	(pictures)
GOP No.=11	3757787	(bits)	of 24	(pictures)
GOP No.=12	3656846	(bits)	of 24	(pictures)
GOP No.=13	2289804	(bits)	of 16	(pictures)
total bits =	45183370	(bits)	of 299	(pictures)

Table 18: Bit amount for each GOP (4) "table tennis" (4Mb/s)

GOP No.= 1	1455562	(bits)	of 19	(pictures)
GOP No.= 2	1606306	(bits)	of 24	(pictures)
GOP No.= 3	1465515	(bits)	of 24	(pictures)
GOP No.= 4	1423011	(bits)	of 24	(pictures)
GOP No.= 5	1615227	(bits)	of 24	(pictures)
GOP No.= 6	1723520	(bits)	of 24	(pictures)
GOP No.= 7	1510076	(bits)	of 24	(pictures)
GOP No.= 8	1590533	(bits)	of 24	(pictures)
GOP No.= 9	1663292	(bits)	of 24	(pictures)
GOP No.=10	1581227	(bits)	of 24	(pictures)
GOP No.=11	1626976	(bits)	of 24	(pictures)
GOP No.=12	1561117	(bits)	of 24	(pictures)
GOP No.=13	1060061	(bits)	of 16	(pictures)
total bits =	19882423	(bits)	of 299	(pictures)

Table 19: Bit amount for each GOP (5) "table tennis" (9Mb/s)

GOP No.= 1	3229786	(bits)	of 19	(pictures)
GOP No.= 2	3592948	(bits)	of 24	(pictures)
GOP No.= 3	3318890	(bits)	of 24	(pictures)
GOP No.= 4	3282381	(bits)	of 24	(pictures)
GOP No.= 5	3520141	(bits)	of 24	(pictures)
GOP No.= 6	3764807	(bits)	of 24	(pictures)
GOP No.= 7	3414133	(bits)	of 24	(pictures)
GOP No.= 8	3532430	(bits)	of 24	(pictures)
GOP No.= 9	3680957	(bits)	of 24	(pictures)
GOP No.=10	3611410	(bits)	of 24	(pictures)
GOP No.=11	3675383	(bits)	of 24	(pictures)
GOP No.=12	3492644	(bits)	of 24	(pictures)
GOP No.=13	2470715	(bits)	of 16	(pictures)
total bits =	44586625	(bits)	of 299	(pictures)

Table 20: Bit amount for each GOP (6) "mobile & calendar" (4Mb/s)

GOP No.= 1	1457696	(bits)	of 19	(pictures)
GOP No.= 2	1573327	(bits)	of 24	(pictures)
GOP No.= 3	1576805	(bits)	of 24	(pictures)
GOP No.= 4	1597176	(bits)	of 24	(pictures)
GOP No.= 5	1595968	(bits)	of 24	(pictures)
GOP No.= 6	1596844	(bits)	of 24	(pictures)
GOP No.= 7	1606819	(bits)	of 24	(pictures)
GOP No.= 8	1597417	(bits)	of 24	(pictures)
GOP No.= 9	1582456	(bits)	of 24	(pictures)
GOP No.=10	1572649	(bits)	of 24	(pictures)
GOP No.=11	1577650	(bits)	of 24	(pictures)
GOP No.=12	1591995	(bits)	of 24	(pictures)
GOP No.=13	1106852	(bits)	of 16	(pictures)
total bits =	20033654	(bits)	of 299	(pictures)

Table 21: Bit amount for each GOP (7) "mobile & calendar" (9Mb/s)

GOP No.= 1	3092109	(bits)	of 19	(pictures)
GOP No.= 2	3514471	(bits)	of 24	(pictures)
GOP No.= 3	3566160	(bits)	of 24	(pictures)
GOP No.= 4	3574878	(bits)	of 24	(pictures)
GOP No.= 5	3610780	(bits)	of 24	(pictures)
GOP No.= 6	3586503	(bits)	of 24	(pictures)
GOP No.= 7	3611063	(bits)	of 24	(pictures)
GOP No.= 8	3594144	(bits)	of 24	(pictures)
GOP No.= 9	3570408	(bits)	of 24	(pictures)
GOP No.=10	3549424	(bits)	of 24	(pictures)
GOP No.=11	3566460	(bits)	of 24	(pictures)
GOP No.=12	3571679	(bits)	of 24	(pictures)
GOP No.=13	2516195	(bits)	of 16	(pictures)
total bits =	44924274	(bits)	of 299	(pictures)

Table 22: Coded Bit Stream files

-rw-r--r--	1	hana	2499423	Oct 12 02:14 flow.bs4
-rw-r--r--	1	hana	5616306	Oct 12 15:07 flow.bs9
-rw-r--r--	1	hana	2504211	Oct 12 20:38 mobl.bs4
-rw-r--r--	1	hana	5615539	Oct 13 01:05 mobl.bs9
-rw-r--r--	1	hana	5647926	Oct 13 04:39 popl.bs9
-rw-r--r--	1	hana	2485307	Oct 11 02:05 tabl.bs4
-rw-r--r--	1	hana	5573333	Oct 11 22:56 tabl.bs9

Table 23: Statistics of Several Items (1) "flower garden" (4Mb/s)

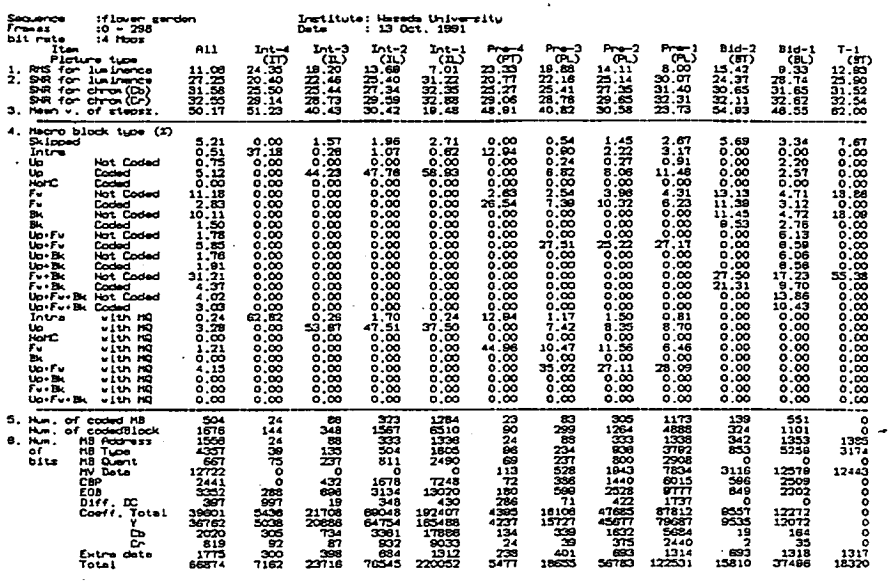


Table 24: Statistics of Several Items (2) "flower garden" (9Mb/s)

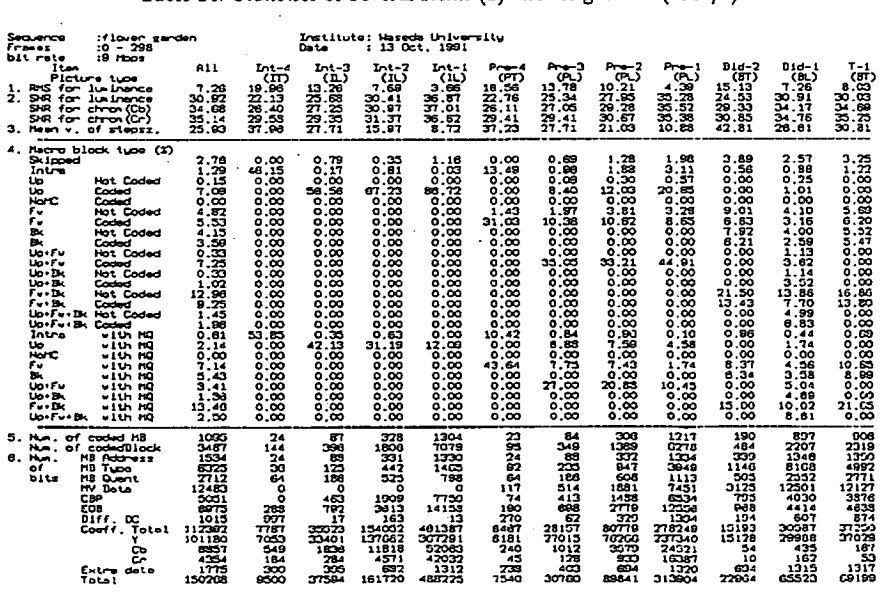


Table 25: Statistics of Several Items (3) "popple" (9Mb/s)

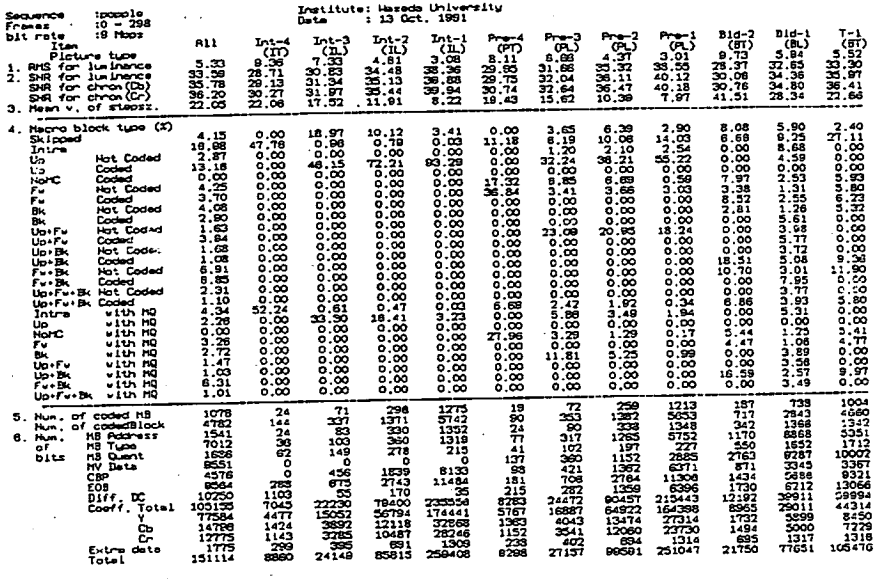


Table 26: Statistics of Several Items (4) "table tennis" (4Mb/s)

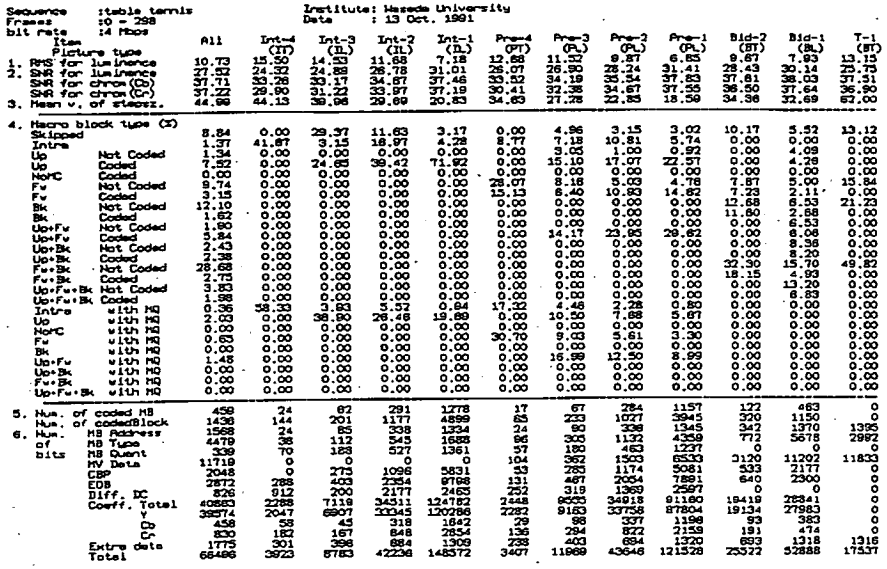


Table 27: Statistics of Several Items (5) "table tennis" (9Mb/s)

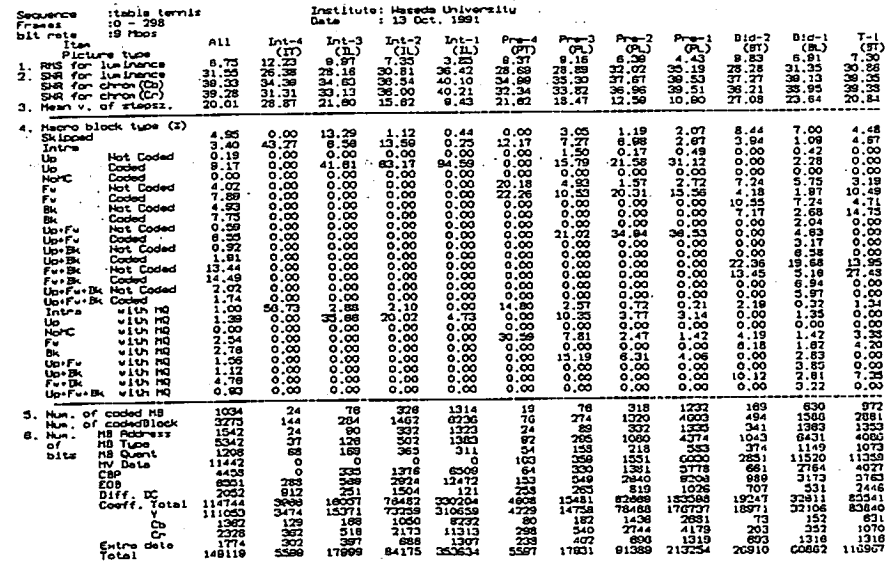


Table 28: Statistics of Several Items (6) "mobile & calendar" (4Mb/s)

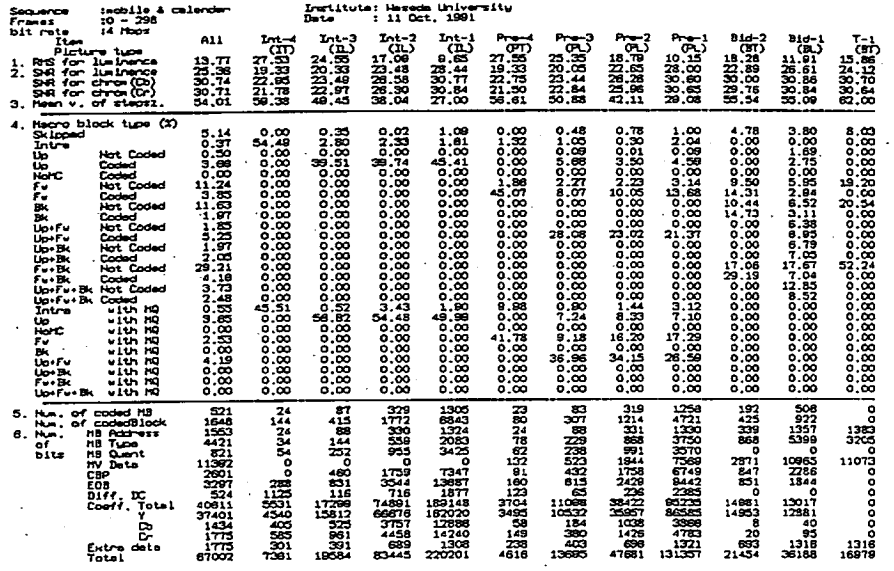
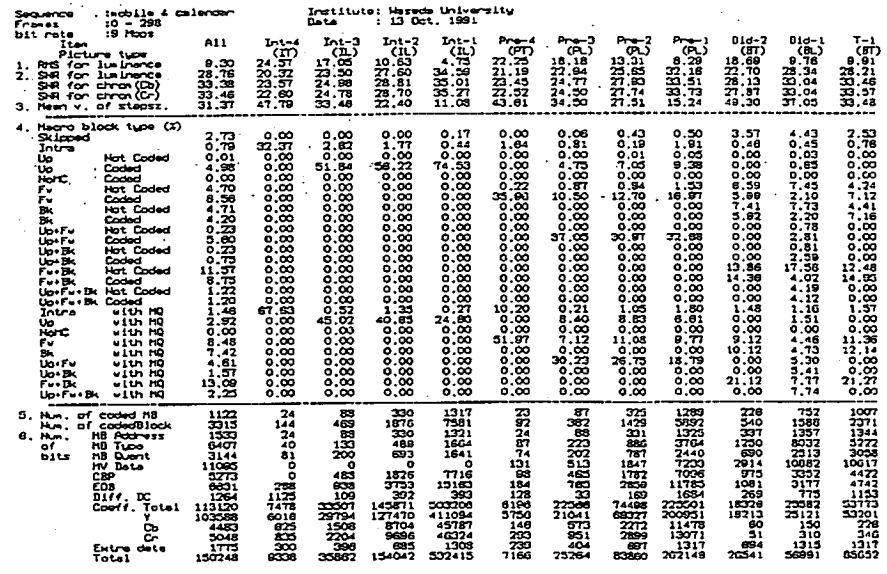


Table 29: Statistics of Several Items (7) "mobile & calendar" (9Mb/s)



Annex

The detail descriptions of the layer encoder and the layer decoder (See 2.1) are shown in Figure 26 and 27 respectively.

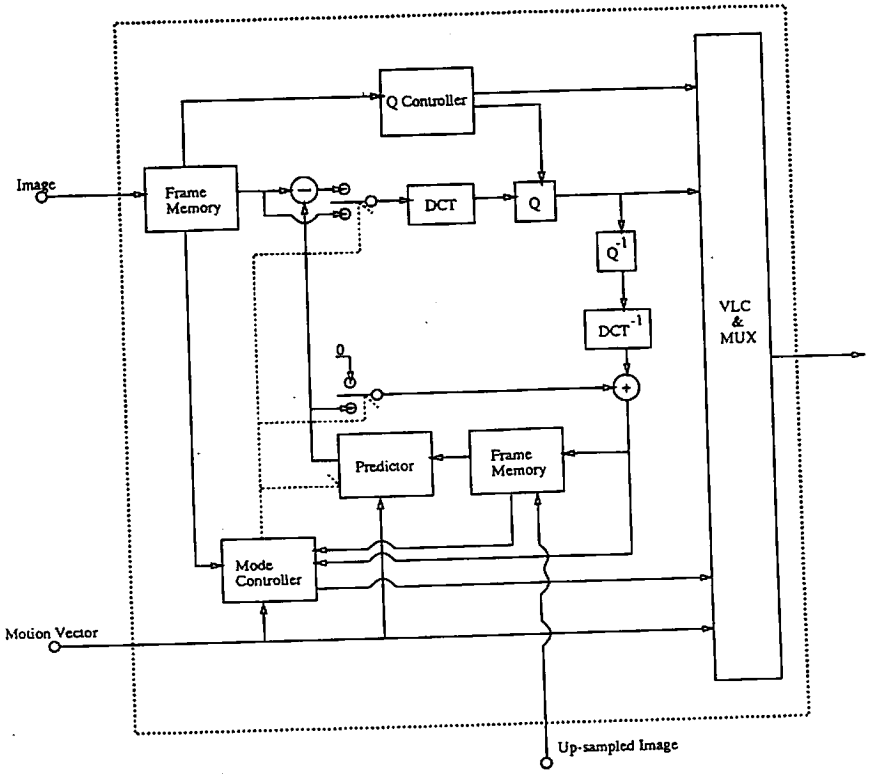


Figure 26: The layer encoder

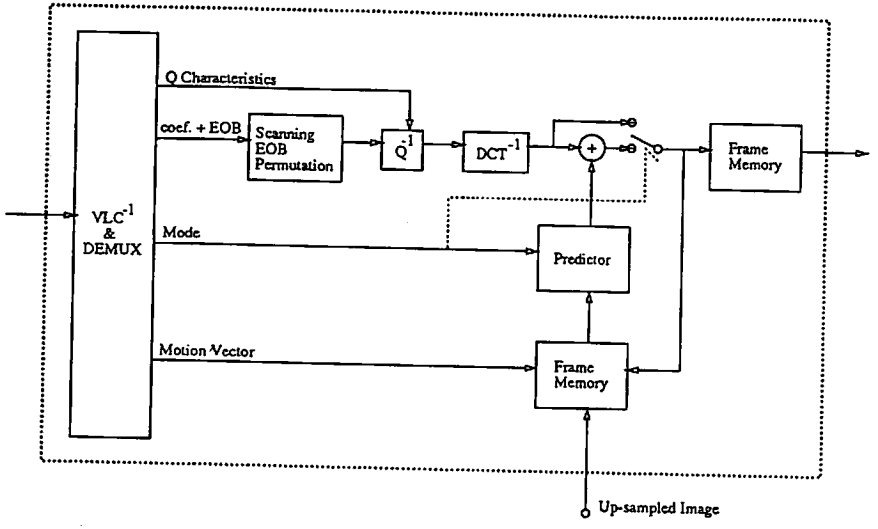


Figure 27: The layer decoder